Flood Precautionary System

A Report on Miniproject Submitted for the requirement of

University of Mumbai

The practical work done during Semester-V in

Miniproject I
(Electronics and Telecommunication Engineering)

by

Judhajit Roy (14ET1063)

Mayuresh Sawant (14ET1015)

Sarthak Vengurlekar (14ET2006)

Under the Guidance of

Kausar Fakir

Department of Electronics and Telecommunication Engineering
Ramrao Adik Institute of Technology,
Sector 7, Nerul , Navi Mumbai
(Affiliated to University of Mumbai)
October 2016

Ramrao Adik Education Society's

Ramrao Adik Institute of Technology

(Affiliated to the University of Mumbai)

Dr. D. Y. Patil Vidyanagar,Sector 7, Nerul, Navi Mumbai 400706.

Certificate

This is to certify that, the Miniproject I titled

"Flood Precautionary System"

is a bonafide work done by

Judhajit Roy (14ET1063)
Mayuresh Sawant (14ET1015)
Sarthak Vengurlekar (14ET2006)

and is submitted in the partial fulfillment of the requirement for the degree of

Bachelor of Engineering

(Electronics and Telecommunication Engineering)

to the

University of Mumbai.

Project Guide Kausar Fakir	Project Coordinator Amruta Chintawar
Head of Department Dr. M. D. Patil	Principal Dr. Ramesh Vasappanavara

Certificate of Approval by Examiners

This is to certify that the submission entitled for the project

"Flood Precautionary System"
is a bonafide work done by **Judhajit Roy (14ET1063), Mayuresh Sawant (14ET1015) and Sarthak Vengurlekar (14ET2006)** under the guidance of **Kausar Fakir** . This project work has been approved for semester V in **Miniproject-I**, University of Mumbai.

Examiners:

Internal Examiner:

External Examiner:

Acknowledgments

I thank my project guide **Kausar Fakir**, the project Coordinator **Amruta Chintawar** and our **H.O.D., Dr.M.D.Patil** for their encouragement and tremendous guidance. I also thank my colleagues for their support and help. I have been fortunate to have received many useful suggestions from my colleagues which have greatly improved the clarity of my report. At the end special thanks to our **Principal Dr. Ramesh Vasappanavara** without whom this project wouldn't have been possible. I would like to appreciate suggestions and criticisms about the report from the readers.

Abstract

Every monsoon, Mumbai, the financial capital of India, effectively gets paralyzed due to heavy rains. The city experiences extreme water logging, flooding and power cuts at public places including roads, railways stations, subways and airports. Most educational institutes, courts and offices shut down, bringing the whole city to a standstill.

Although the India Meteorological Department provide blanket warnings, they have limited resources. To help understand flooding at a street level there is need for a system that monitors the root cause. The Flood Precautionary System is a system that can be set up in different streams, canals and sewers across the city which are flood prone. Whenever the water level in them rises up to a certain dangerous level, it automatically detects it and sends a warning and alerts people to stay away from that area.

Ultrasonic Sensor senses the water level and feeds the data to the Raspberry Pi. Raspberry Pi further analyses the data and calculates the distance from top of the water surface. This data is then transmitted through various channels and shared publicly. Also a WhatsApp AutoReply Chat Bot continuously is fed by the distance value which can be seen just by typing distance.

List of Figures

Contents

Chapter 1

Introduction

Flooding, both predicted and sudden, has become an increasingly common phenomenon in recent years. Many suffering localities have been unprepared or underprepared, with little reason to believe they were at risk until it was too late. A reliable and cost-effective advance warning system that continuously monitors rivers, drains and overflows for rising water levels can save lives and protect property, alerting authorities and communities to impending danger.

The Flood Precautionary System uses approach of 'Internet of Things', to create a system that warns you about the impending danger. In India, social media applications like WhatsApp, Twitter And Facebook are very popular. There are over 23 million Twitter users, 70 million WhatsApp users and over 195 million Facebook users. So a flood warning in such widely used social media platforms won't go unnoticed. Thus we decided upon social media as our warning platform.

1.1 Internet of things(IOT)

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

A thing, in the Internet of Things, can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low – or any other natural or man-made object that can be assigned an IP address and provided with the ability to transfer data over a network.

A widespread application of Internet of Things is'Environmental monitoring applications of the IoT typically use sensors to assist in environmental protection by monitoring air or water quality, atmospheric or soil conditions, and can even include areas like monitoring the movements of wildlife and their habitats. Development of resource constrained devices connected to the Internet also means that other applications like earthquake or tsunami early-warning systems can also be used by emergency services to provide more effective aid. IoT devices in this application typically span a large geographic area and can also be mobile. It has been argued that the standardization IoT brings to wireless sensing will revolutionize this area. In this project we use Internet of Things (IOT) to create a system that alerts those in danger about the incoming flood.

1.2 Raspberry Pi and its Features

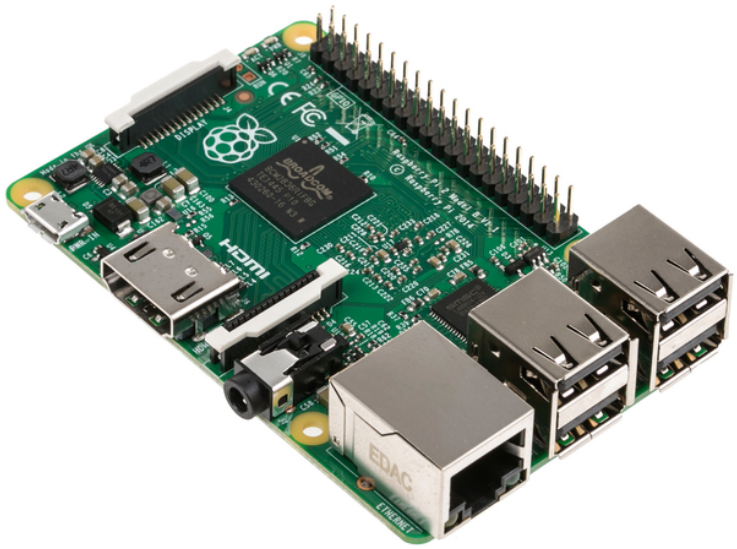

Figure 1.1: Raspberry Pi 2

Raspberry Pi is a dynamic microcontroller that is capable of just about anything a computer is. It runs with the Python programming language, and is a great way to learn about hardware hacking and coding. Python is a wonderful and powerful programming language that's easy to use (easy to read and write) and with Raspberry Pi lets you connect your project to the real world. Python syntax is very clean, with an emphasis on readability and uses standard English keywords. The Python programming language actually started as a scripting language for Linux. Python programs are similar to shell scripts in that the files contain a series of commands that the computer executes from top to bottom. Python is a very useful and versatile high level programming language, with easy to read syntax that allows programmers to use fewer lines of code than would be possible in languages such as assembly, C, or Java. Python programs dont need to be compiled before running them, as you do with C programs. However, you will need to install the Python interpreter on your computer to run them. The interpreter is the program that reads the Python file and executes the code. There are programs like Py2exe or Pyinstaller that can package Python code into stand-alone executable programs so you can run Python programs on computers without the Python interpreter installed.

Like shell scripts, Python can automate tasks like batch renaming and moving large amounts of files. Using IDLE, Pythons REPL (read, eval, print, loop) function can be used just like a command line. However, there are more useful things you can create with Python. Programmers use Python to create things like:

Web applications, Desktop applications and utilities, Special GUIs, Small databases and 2D games. Python also has a large collection of libraries, which speeds up the development process. There are libraries for everything you can think of game programming, rendering graphics, GUI interfaces, web frameworks, and scientific computing.

Python 2 and Python 3 come pre-installed on Raspbian, but to install it on another Linux OS or to update it, simply run sudo apt-get install python3 or sudo apt-get install python. To access Python from the command prompt, type python or python3 depending on which version you want to use. This opens up the Python REPL (read-eval-print-loop),

from which you can enter Python commands just like you use the command line.

The terminal (or 'command-line') on a computer allows a user a great deal of control over their system. This tool allows a user to directly manipulate their system through the use of commands. These commands can be chained together and/or combined together into complex scripts that can potentially complete tasks more efficiently than much larger traditional software packages. On the Raspberry Pi (running Raspbian), the default terminal application is "LXTerminal". This is known as a 'terminal emulator', this means that it emulates the old style video terminals (from before graphical user interfaces were developed) in a graphical environment. The application can be found on the Raspberry Pi desktop

One powerful feature of the Raspberry Pi is the row of GPIO (general purpose input/output) pins along the top edge of the board. These pins are a physical interface between the Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on or off (input) or that the Pi can turn on or off (output). Of the 40 pins, 26 are GPIO pins and the others are power or ground pins plus two ID EEPROM pins.

You can program the pins to interact in amazing ways with the real world. Inputs don't have to come from a physical switch; it could be input from a sensor or a signal from another computer or device, for example. The output can also do anything, from turning on an LED to sending a signal or data to another device. If the Raspberry Pi is on a network, you can control devices that are attached to it from anywhere and those devices can send data back. Connectivity and control of physical devices over the internet is a powerful and exciting thing, and the Raspberry Pi is ideal for this.

1.3 Why Ultrasonic Sensor?

Figure 1.2: UltraSonic Sensor

Ultrasonic level sensors work by the 'time of flight" principle using the speed of sound. The sensor emits a high-frequency pulse, generally in the 20 kHz to 200 kHz range, and then listens for the echo. The pulse is transmitted in a cone, usually about 6 degree at the apex. The pulse impacts the level surface and is reflected back to the sensor, now acting as a receiver, and then to the transmitter for signal processing.

The sensor sends pulses toward the surface and receives echoes pulses back. Basically, the transmitter divides the time between the pulse and its echo by two, and that is the distance to the surface of the material.

Unlike other methods of water level sensing, Ultrasonic and Infrared Sensors are contact less methods. In case of a Flood a Sensor using contact method might get destroyed or washed away but a contact less sensor would survive as it is setup in a higher position than the actual water level.

Now if you compare Ultrasonic And Infrared Sensors, Ultrasonic sensors are far more accurate than Infrared Sensors in measuring distance of an obstacle. Moreover Infrared Sensors cannot be used outside, as the Infrared isn't detectable properly in presence of sunlight whereas Ultrasonic Sensors work properly both indoors and outside. Since the Ultrasonic Sensors use sound as a medium the echo signal might not be detected if the obstacle is a sound absorbent. But the application for which it is being used might not encounter such issue. Also Ultrasonic Sensors are slightly more costlier than Infrared Sensors but certainly a better choice in terms of accuracy.

1.4 System Working

The Project uses 'Raspberry Pi 2' and 'Ultrasonic Sensor to create a Flood Precautionary System. The Ultrasonic Sensor measures the distance from the top of the water level. The Raspberry Pi 2 is used as medium of communication between the Ultrasonic Sensor and the various channels of warning transmission. The various channels include the likes of Twitter, Facebook And Email. WhatsApp is another channel of communication which is used a bit differently. WhatsApp is used like a server channel. Where data can be monitered at each and every instant. It also gives a more accurate value of distance as the value sent in the message is Floating Point type.

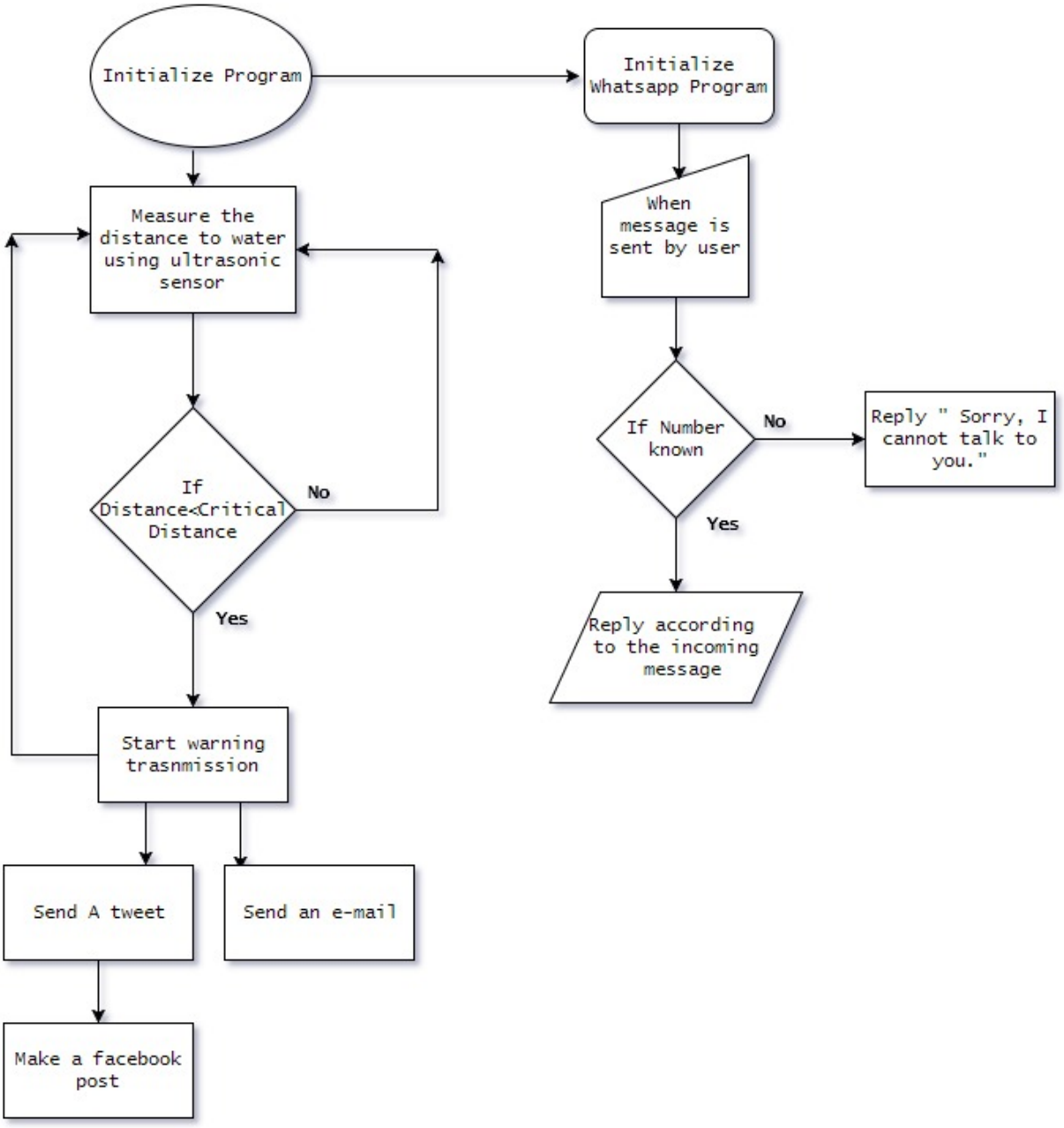

Figure 1.3: System Block Diagram

Chapter 2

Literature Survey

2.1 Local Literature

Mumbai Floods

Any discussion of floods in Mumbai begins with a ritual invocation of one fateful date:- 26 July 2005. On this day, the megacity received 944mm of rainfall the average amount for the entire season, and a 100-year high. This, combined with high tides, set off a devastating flood in the city, much of which is built on low-lying land reclaimed in the 19th century. In the catastrophe that ensued, electricity, water supply, communication networks and public transportation were totally shut down; 1,493 people perished, more than 14,000 homes were destroyed, and the city incurred losses amounting to euro 1.2bn.

In response, in 2007, the municipality formulated the Greater Mumbai disaster management action plan, which identified the risks and vulnerabilities the city could face, including floods, earthquakes and cyclones; it also formed the Disaster Management Cell to co-ordinate relief and rescue efforts, and widened and deepened the Mithi River, which drains into the Arabian Sea. But environmentalists and researchers contend that too many factors which make Mumbai vulnerable to flooding remain: unabated construction on floodplains and coastal areas, as well as the continuing problem of storm-water drains and waterways being clogged by plastic garbage.

The floods have been the subject of research by scientists and social scientists attempting to understand the causes, impacts, and short/long term consequences. Scholars have studied the floods in Mumbai from the perspectives of climate change, disaster management / mitigation, urban health, vulnerability and adaptation, hydrology, environmental degradation and encroachment etc.

2.2 Local Studies

Dadasaheb K. Mane, in his International Journal of Engineering Research and Technology (IJERT) journal titled 'Disaster Flood Alert System Using Ultrasonic Sensor' Disaster flood alert system using GSM and ultrasonic frequency sensors has described a important technology which is useful to make the people alert from disaster flood, in this project ultrasonic transducers are used to find out the water level of the flood . And then information given to the controller and GSM, this system continuously send the messages towards control room about the level of the flood when water level will change.

The project is installed in the river by creating a supportive basement in the river and to decide the water levels in the river ultrasonic sensor will sense the continuously the water level of the selected river.

2.3 Foreign Studies

Iowa severe flooding in 2008 demonstrated the need for more extensive monitoring of the states rivers and streams in real time. To address this, the Iowa Flood Center (IFC) developed and maintains a statewide network of stream stage sensors designed to measure stream height and transmit data automatically and frequently to the Iowa Flood Information System (IFIS). Data collected from the sensors are automatically sent to the Iowa Flood Information System (IFIS), where the real time monitoring information is integrated into an advanced hydrological model. System data and river stage hydrographs are shared with the public and emergency management officials.

With easy online access to the water level data from the sensors, Iowa residents and state agencies can now get the reliable, real-time information they need to manage flooding in their locales.

The sensors were developed as a student project to design an affordable, yet effective way to measure stream and river heights. The sensors are solar powered and attached to the side of bridges. A sonar signal is used to measure the distance from the water surface to the sensor and data is transmitted via a cell modem to IFIS where the data is publicly available.

Before the system was in place, it was common for emergency personnel to be dispatched to assess the flooding in threatened locations. But with the stream gauges collecting the data in real-time, emergency responders can focus on helping people instead of tracking flood waters.

Real Time Wireless Flood Monitoring System Using Ultrasonic Waves another project by Abubakr Rahmtalla Abdalla Mohamed and Wang Guang Wei of Tianjin University of Technology and Education, Department of Electronics Engineering. The aim of this project is to develop prototype of water level detection that can be viewed as a part of control system of river flow management system. The system consist of two parts, transmitter and receiver modules. Transmitter module detect water level automatically, then transmit the data to receiver. Ultrasonic sensor is used to detect the distance between sensor and the water surface. Water level detection is performed without physical contact between the sensor and water surface. Ultrasonic sensors utilize the principle of sound reflection to measure the level of the water. Elapsed time required to transmit and receive the reflected ultrasonic wave is multiplied by the rapid propagation of sound in water in order to obtain the distance value.

The calculation is performed by high level language program that reside in microcontroller. The distance value then is transmitted using wireless network (RF transmitter). In receiver module, distance value received through the wireless network (RF receiver) is passed to another microcontroller to display the water level using an LCD. Then depending on the measurements of the previous years for the same river we also have a set of LEDs to show that the current value of the water level in that area.

Rivers in Honduras flood frequently, causing a major trouble to people and their

lives there. Seeing the pain that the Honduras citizens had to suffer from on a daily basis prompted Robert Ryan-Silva, the director of DAI maker lab, to take on the "Hidrosnico project".

Considering villages and farms in Honduras are located on river banks, they're highly prone to flooding. This makes it an extremely tedious task to design a full proof solution for the same. Making things worse is the fact that mobile phones have a mere 20 percent penetration in Honduras. Hence, a solution had to be invented that couldn't not only be able to detect rising flood waters in time and alert the families to evacuate, but also one thats rugged, affordable and easy to install.

For the unaware, Sonar IoT sensors make use of the same principal to detect water level that the bats use to see. Bats make use of echolocation in order to see with the help of sound. They do so by sending out "bat call"(sounds), which bounce off nearby objects. The bats can detect nearby objects and their speed by sensing how long it takes for the echo to return. The same way the sensors mounted on the Honduras bridge send high-frequency sounds onto the water and measure the time that it takes for the echo to reflect back. As and when the water level rises, the rivers flood, making the return time for the echo to shrink as there's a shorter distance to cover.

Hidrosnico is a stream gauge using a MaxSonar HRXL MB7369 sonar rangefinder, a Seeeduino Stalker v3 Arduino-compatible microcontroller platform, and an Adafruit FONA 800 GSM module. It reads the distance between the sensor (specifically from where the housing meets the threading) and the water's surface directly below and sends this data at specified intervals to cloud-based storage, SMS, and/or email as determined by the user. It was designed to aid in collection of hydrological data and for flood early warning in developing countries.

Chapter 3

Proposed Methodology

3.1 WhatsApp AutoReply Chat Bot

We use the following File structure for the Chat bot:

1. PROJECTDIR/run.py
2. PROJECTDIR/layer.py

For writing a Yowsup project we need at least 2 files (well we can put everything in 1 file but that wouldn't be so readable would it). Inside layer.py we are going to define our logic for the auto reply chat bot. Inside the run.py we are going to initialize the yowsup stack with the Chat Bot layer on top.

layer.py

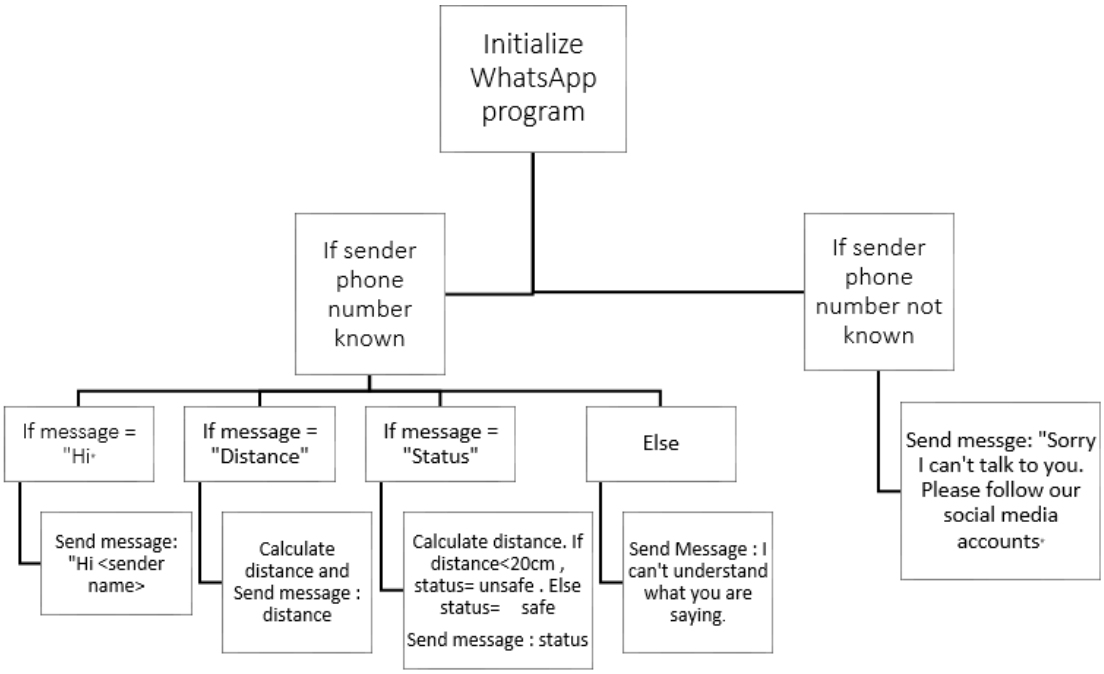

Figure 3.1: layer.py Flowchart

For a project to use Yowsup, it needs to integrate itself as a layer in the Yowsup stack. So this is where we define our Layer. Because we know the layer is going to be placed in the stack directly above the protocol layers, we know it's going to receive data of type ProtocolEntity. However, since we are only looking for incoming messages, we can filter those by checking the return value of 'protocolEntity.getTag()'. And then we let the 'onMessage' method take over. To send the message back we create a TextMessageProtocolEntity object and send it to the layer below using 'toLower' method.

While chat bot of the above implementations for the Layer is enough to get the client functioning, we will notice every time we reconnect using Yowsup we will receive the same messages we previously received. This is because we need to send a receipt to WhatsApp including the Id for the received messages. This is what makes double ticks appear in WhatsApp on the sender's phone.

When we send a message, we receive a receipt from WhatsApp indicating delivery to server and to recipient. We need to send an 'Ack' for that receipt to prevent receiving this receipt over and over, and to prevent WhatsApp sending us "stream:error" and closing the connection for not sending those acks.

run.py

This is the code that will initialize the stack with the new Layer on top. Here we create the stack and we manually include all layers. For the protocol layers, our Layer is interested only in Authentication (YowAuthenticationrotocolLayer), sending and receiving text messages (YowMessagesProtocolLayer), sending and receiving receipts (YowReceiptProtocolLayer) and sending Acks for receipts (YowAckLayer). If we were interested for example in media messages, we would have added YowMediaProtocolLayer.

3.2 Twitter, Facebook and Email

First, we import the necessary libraries. Also, we need to assign import the access tokens and access secret generated for our twitter app. We need to add the email id and password of the sender (Raspberry in our case) and the receivers. Start the Gmail email server using SSMTP. Since we need to measure the water level until the program execution is stopped, we put in an infinite loop wherein the sensor will keep measuring the water level and then transmitting the warnings accordingly.

Next, we need to name our input and output pins, so that we can refer to it later in our Python code and then set your two GPIO ports as either inputs or outputs as defined previously. Then, ensure that the Trigger pin is set low, and give the sensor a second to settle. The HC-SR04 sensor requires a short 10uS pulse to trigger the module, which will cause the sensor to start the ranging program (8 ultrasound bursts at 40 kHz) in order to obtain an echo response. So, to create our trigger pulse, we set out trigger pin high for 10uS then set it low again.

Now that weve sent our pulse signal we need to listen to our input pin, which is connected to ECHO. The sensor sets ECHO to high for the amount of time it takes for the pulse to go and come back, so our code therefore needs to measure the amount of time that the ECHO pin stays high. We use the "while string to ensure that each signal timestamp is recorded in the correct order. The time.time() function will record the latest timestamp for a given condition. For example, if a pin goes from low to high, and were

recording the low condition using the time.time() function, the recorded timestamp will be the latest time at which that pin was low.

Our first step must therefore be to record the last low timestamp for ECHO (pulsestart) e.g. just before the return signal is received and the pin goes high. Once a signal is received, the value changes from low (0) to high (1), and the signal will remain high for the duration of the echo pulse. We therefore also need the last high timestamp for ECHO (pulseend). We can now calculate the difference between the two recorded timestamps, and hence the duration of pulse (pulseduration). With the time it takes for the signal to travel to an object and back again, we can calculate the distance using the following formula.

$$Speed = Distance/Time \tag{3.1}$$

The speed of sound is variable, depending on what medium its travelling through, in addition to the temperature of that medium. However, some clever physicists have calculated the speed of sound at sea level so well take our baseline as the 343m/s.

$$34300 = 2 \times Distance/Time \tag{3.2}$$

$$17150 = Distance/Time \tag{3.3}$$

$$Distance = 17150 \times Time \tag{3.4}$$

Store the measured distance in a variable and calculate whether it is less than the predefined critical distance. If the distance is more, go back. Else proceed further. Send a tweet using api.updatestatus. We have synced our Facebook page and Twitter account, whenever there is a Tweet the same text is posted on the Facebook page. Then we send an email using s.sendmail. Then the process is repeated. Finally, we clean our GPIO pins to ensure that all inputs/outputs are reset.

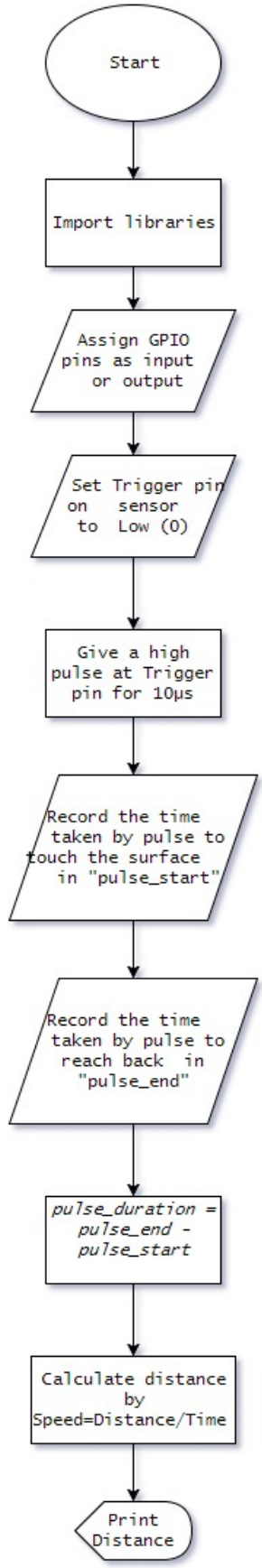

Figure 3.2: Program Flowchart

Chapter 4

System Design

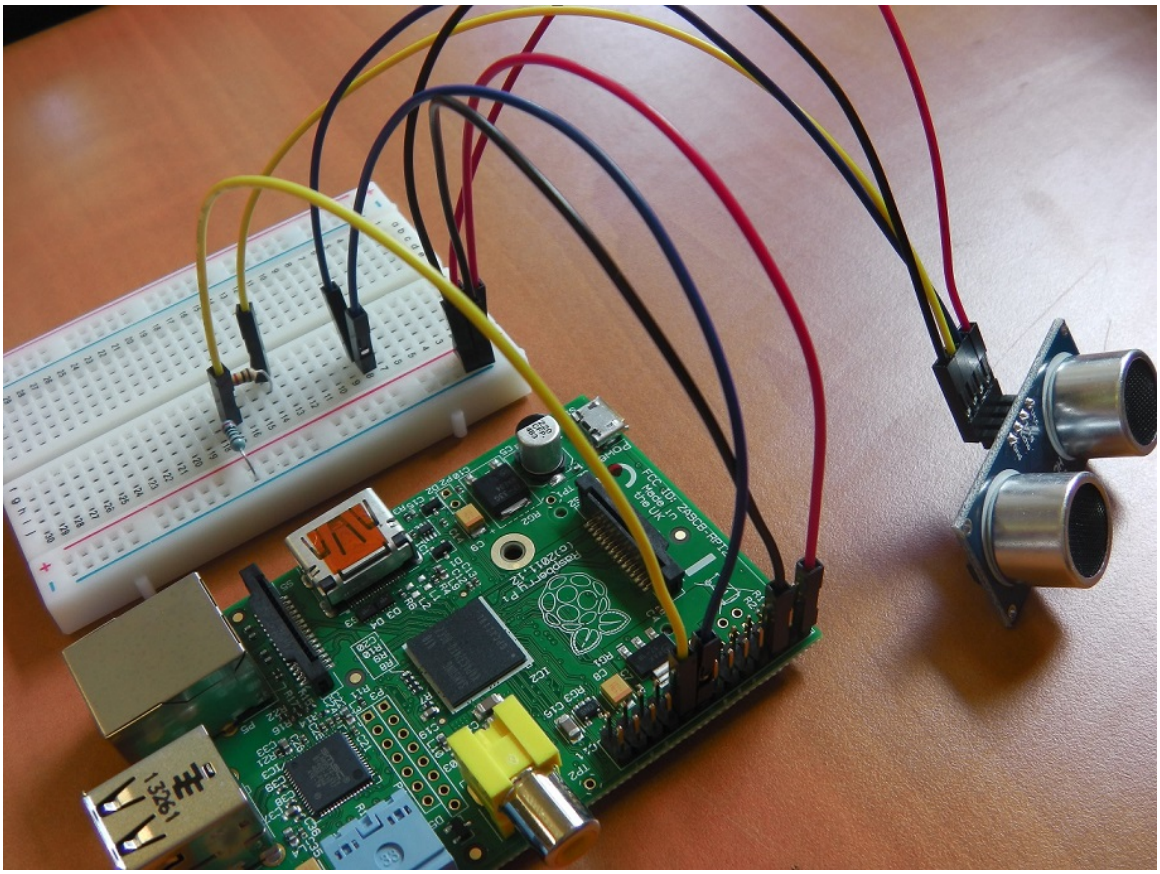

Figure 4.1: Working Model

4.1 Components

4.1.1 Hardware

1. Raspberry Pi Model 2
2. EW-7811Un Wi-Fi USB Adapter
3. HC-SR04 Ultrasonic Sensor

4. Resistor (1kΩ)
5. Resistor (2kΩ)
6. Jumper Wires
7. Bread board

4.1.2 Software

1. Raspbian Wheezy OS
2. Twython (Twitter API)
3. Secure simple mail transfer protocol (SSMTP)
4. Yowsup2

4.2 Descriptions

HC-SR04 Ultrasonic Sensor

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

Test distance = (high level timevelocity of sound (340M/S)) / 2

Pins:

1. VCC: +5VDC
2. Trig : Trigger (INPUT)
3. Echo: Echo (OUTPUT)
4. GND: GND

Twython

Twitter has an Application programming interface (API) and Raspberry Pi can run several software development kits (SDKs) for their API. You can download a bunch of different libraries, for this purpose i.e to access twitter functions. Raspberry Pi can make tweets. You can just connect and make a tweet. Also Raspberry Pi can respond to tweets. So, not only can it send tweets, it can look through tweets and do something if a tweet occurs, or something like that. So, it may look for a tag, maybe you want to search for tags of a different type and look at the text in those or something like that. But, it's a two-way thing. So, Raspberry Pi can send tweets, but also it can receive tweets and scan through tweets.

Twython is the package that is used to access Twitter's API. There are other ones but Twython is pretty good. Twython is a very robust, mature library for Twitter. It has been

maintained for over 2 years. It's actively maintained by its core team and receives patches from the community very regularly. This is not an alpha-status library; it's tested and used in many commercial applications. Twython is the premier Python library providing an easy (and up-to-date) way to access Twitter data. Actively maintained and featuring support for Python 2.6+ and Python 3.

We registered a Twitter app and used the generated access tokens and access secrets in the program to use Twython.

Figure 4.2: Twython Access Token

Yowsup2

Yowsup is a Python library that allows you to login and use the WhatsApp service and provides you with all capabilities of an official WhatsApp client, allowing you to create a full-fledged custom WhatsApp client. By using Yowsup you can make Raspberry Pi a WhatsApp client. Raspberry Pi will respond to messages sent by the person its in converstion with. You can specify certain fixed responses for messages sent.

4.3 Circuit Diagram

1. The Ground Pin of Ultrasonic Sensor is connected to GPIO GND (Pin 6)
2. The VCC is connected to GPIO 5V (Pin 2)
3. Trigger Pin connected to GPIO 23 (Pin 16)
4. The sensor output signal (ECHO) on the HC-SR04 is rated at 5V. However, the input pin on the Raspberry Pi GPIO is rated at 3.3V. Sending a 5V signal into that unprotected 3.3V input port could damage your GPIO pins. Well need to use a small voltage divider circuit, consisting of two resistors, to lower the sensor output voltage to something our Raspberry Pi can handle.

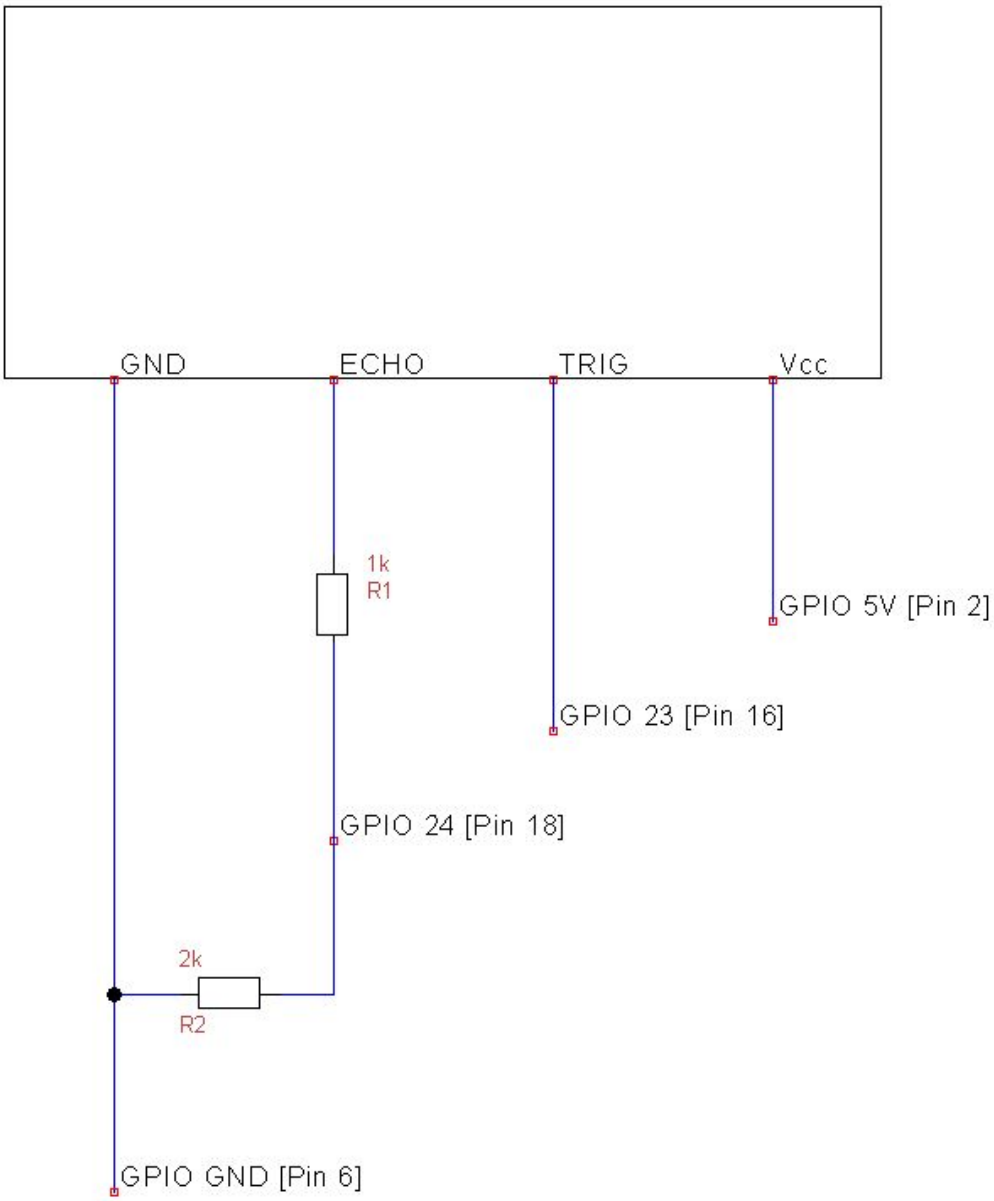

Figure 4.3: Circuit Diagram

Chapter 5

Results

The program was successfully executed. whenever the level water rose to the critical level a warning was transmitted via various channels. The following are the channels:

1. **Email**:- An email alert was automatically sent as the water level reaches the critical level. The alert message has a warning statement 'Stay Away' and states the distance from top.

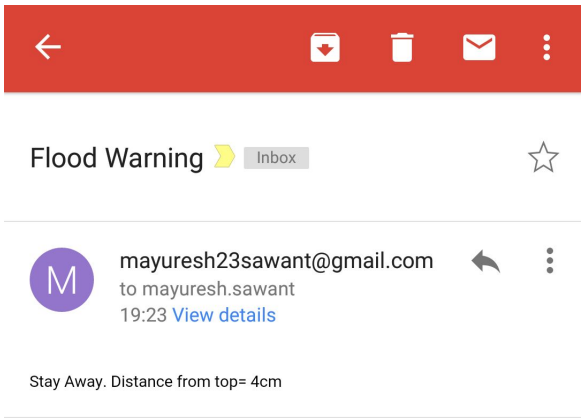

Figure 5.1: Email Output Screenshot

2. **Twitter**:- A Tweet alert was automatically sent as the water level reaches the critical level. The alert tweet has a warning statement 'Please Stay Away' and states the distance from top. Also the time was included in Tweet because if the water level remains the same for sometime and a tweet is delivered then it will be considered as a duplicate tweet which is not allowed in Twitter. This will generate an error in program and effect the execution of program.

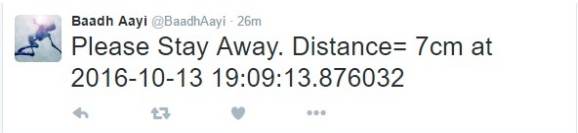

Figure 5.2: Twitter Output Screenshot

3. **Facebook**:- Facebook warning transmission was synced with Twitter. As soon as a warning was send on Twitter the same warning message was posted on the Facebook Page of the project.

4. **WhatsApp**:- WhatsApp was used as a Real Time Monitoring System. When the user types 'Distance' the distance from top was displayed similarly the status and the temperature of the Raspberry Pi 2 was displayed for respective commands.

Figure 5.3: WhatsApp and Facebook Output Screenshot

Chapter 6

Conclusion

The Flood Precautionary System is a modern and efficient way of warning common people and the concerned authorities about the incoming flood with the use of small and effective components with Internet of Things. Real time monitoring system gives accurate predictions than the Met Department. This project will minimise the loss of life and property caused by the flood by issuing alerts well in advance.

The system can be set up across different sewers, rivers, nullahs, streams and other areas that are usually affected by water logging. A network of such devices across the city will help in bringing the data to the people on their devices. The system can also result in better management of traffic in case of a calamity, as it would alert the motorists about water logging in a particular area.

Cost involved in this project is a fraction of what the existing systems cost. The software used in it is all open source. This makes it absolutely free. Open source also has an interesting prospect of collaboration i.e more people can modify the software to enhance it's functionality. The hardware is easy to set up with only a few components and requires minimal effort. The set up of the is sensor away from the main circuit because, in case of an actual flood, only the sensor will be damaged and not the entire circuit.

This system can be implemented across India and the technology can be further improved upon by enthusiasts.

Chapter 7

Future Scope

This Project is a modern approach for assisting in tackling a Flood Calamity by providing head start information through channels like WhatsApp, Facebook, Twitter and Email. However in a country as diverse as India not everyone use these channels. These shortcomings can be overcome by using other channels and bringing this network to the entire community. Another useful feature would be storing the sensed data for future use.

India being a third world country many people do no have access to internet. This creates a problem when transmitting through channels that require internet. This problem can be taken care of by using off-line channels such as SMS or a radio broadcast. A GSM module or a 2G/3G dongle along with a prepaid sim card can be used to interface the Raspberry Pi to send an SMS. Also, there are many online serives to send SMSs in bulk over the internet. This SMS service can be used by people who do not have access to internet. PiFM package can be installed and programmed in Raspberry Pi to make it a FM radio Transmitter.

If the data collected by the sensors can be stored and tracked by real time data logging. This data can be used to plot graphs and create a map of the whole city using network of sensors of sensors across the city.

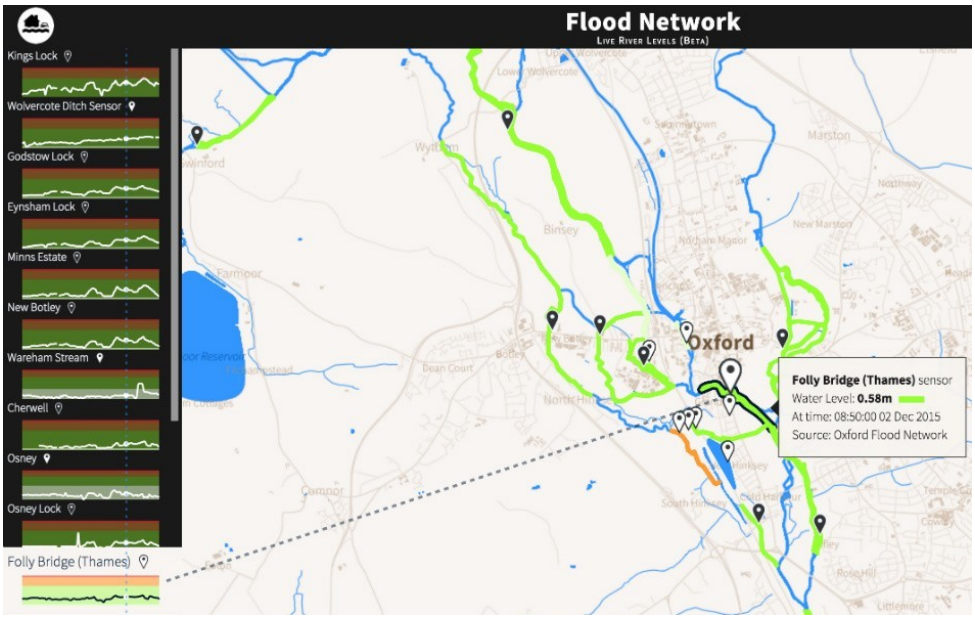

Figure 7.1: Flood Network used in Oxford, UK

The information gathered will be made available to the community as open data and distributed to the community via the channels. The information gathered in the past years across the area can be analysed and put together to create a map of flood levels that will give residents a clearer picture of the situation in their area.

Also by using Machine Learning and Artificial Intelligence a prediction of the actual the time of occurrence of flood can be made by plotting distance versus time graph of the sensor data.This could help the people and the authorities to take preventive measures well in advance.

Bibliography

[1] Google images (http://www.google.com/images)

[2] https://senix.com/water-level-sensors-provide-flood-warning/

[3] http://www.slideshare.net/markanthonymuya/road-flood-sensor-with-web-and-mobile-application-support

[4] https://www.ijsr.net/archive/v3i8/MDIwMTUxNDQ=.pdf

[5] http://iowafloodcenter.org/projects/stream-stage-sensor/

[6] http://www.controlglobal.com/articles/2013/automation-technology-ultrasonic-continuous-measurement/

[7] https://github.com/DAI-Maker-Lab/hidrosonico

[8] http://www.indianweb2.com/2016/07/07/introducing-sonar-iot-system-can-provide-early-warning-floods/

[9] https://www.raspberrypi.org/documentation/usage/

[10] http://www.computerweekly.com/news/2240232979/Oxford-Flood-Network-brings-Internet-of-Things-to-the-community

[11] http://www.circuitbasics.com/how-to-write-and-run-a-python-program-on-the-raspberry-pi/

[12] https://github.com/alaudet/raspi-sump

[13] https://github.com/tgalal/yowsup/wiki/Sample-Application

[14] http://www.makeuseof.com/tag/how-to-build-a-raspberry-pi-twitter-bot

[15] https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi

[16] http://www.instructables.com/id/WhatsApp-on-Raspberry-Pi/