नेताजी सुभाष प्रौद्योगिकी संस्थान
NETAJI SUBHAS INSTITUTE OF TECHNOLOGY
(An Institution of Govt. of NCT of Delhi-Formerly, Delhi Institute of Technology)
Azad Hind Fauj Marg, Sector -3, Dwarka, New Delhi-110078
Tel. : 25099050, Fax : 25099025, Website : http://www.nsit.ac.in

**B.E. PROJECT ON**

# LoRa based Untethered Water Pollution Monitoring System

Submitted By

| | |
|---|---|
| **APOORV ARYAN** | **2017UEC2039** |
| **SWAPNIIL KANWAR** | **2017UEC2099** |
| **HARSH SHARMA** | **2017UEC2172** |

Under the Guidance of
Prof. Dhananjay V. Gadre

Project-I in partial fulfilment of requirement for the award of
B.E. in
Electronics & Communication Engineering

Division of Electronics & Communication Engineering
NETAJI SUBHAS INSTITUTE OF TECHNOLOGY
(UNIVERSITY OF DELHI)
NEW DELHI-110078
2021

# CERTIFICATE

This is to certify that the report entitled **"LoRa based Untethered Water Pollution Monitoring System"** being submitted by **APOORV ARYAN(2017UEC2039)**, **SWAPNIIL KANWAR(2017UEC2099)**and **HARSH SHARMA(2017UEC2172)** to the Division of Electronics and Communication Engineering, NSIT, for the award of bachelor's degree of engineering, is the record of the bonafide work carried out by them under our supervision and guidance. The results contained in this report have not been submitted either in part or in full to any other university or institute for the award of any degree or diploma.

**Dhananjay V. Gadre**
Associate Professor
ECE Division
Netaji Subhas Institute of Technology
Sector-3, Dwarka. New Delhi, India
website: http://www.dvgadre.com
e-mail: dvgadre@gmail.com

# ACKNOWLEDGEMENT

*'No great work is ever done in a hurry. To develop a greeat scientific dicovery, to print a great picture, to write an immortal poem - to do anything great requires time, patience, and perseverance'*

*W.J. Wilmont Buxton*

We would sincerely like to thank Prof. Dhananjay V. Gadre for being our project guide. We are also grateful to him for providing this very simple yet so vast topic for a thesis project. We are also thoroughly thankful to him for providing us with all the other necessary facilities in this difficult time which had led to the completion of this project. His passion for teaching is exceptional and was very motivating for us.

Besides our guide, we would like to thank our senior Mr. Mrityunjay Kumar and Mr. Ravi Kumar for their continuous support and help regarding the various application and new ideas regarding this project in the starting phase.

We would also like to extend our gratitude to junior Naman Puri for helping us out in this tough situation whenever we had asked him.

Last but not the least, we would like to thank our family, our parents for constantly believing in us and keeping us motivated throughout our life despite the problems or the regular failures. We express our gratitude to them.

It was a great learning experience to work on this topic and develop it as a project. This project has given us the learning that there can not be any end of gaining knowledge even for simpler topics. There is always something to know more about.

# ABSTRACT

*'He who will not reason is a bigot; he who cannot is a fool; and he who dares not is a slave'*

*Lord Byron*

This thesis describes the design and implementation of a LoRa Based Untethered Water Pollution Monitoring System using ESP32 Heltec board and our approach to help suggest a solution for more direct approach to counter the void of availability of the information to the public. Our solution aims to provide water quality data access to anyone anywhere with an internet connection. The thesis is not just limited to LoRa and the technology used but also covers the comparison and advantages of using Lora over conventional methodologies and different ways of implementation with their advantages and disadvantages respectively. Finally the designing and individual sensor circuit is looked into and discussed viz. pH, turbidity, electrical conductivity of water.

# PLAGIARISM REPORT

ORIGINALITY REPORT

| %7 | %7 | %5 | % |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | www.intechopen.com<br>Internet Source | %4 |
| 2 | cvc.ca<br>Internet Source | %1 |
| 3 | unstats.un.org<br>Internet Source | %1 |
| 4 | Paul Brooker. "Chapter 6 Comparisons with Criminality", Springer Science and Business Media LLC, 2010<br>Publication | <%1 |
| 5 | elearning.vtu.ac.in<br>Internet Source | <%1 |
| 6 | idoc.pub<br>Internet Source | <%1 |
| 7 | upcommons.upc.edu<br>Internet Source | <%1 |
| 8 | www.oak.go.kr<br>Internet Source | <%1 |

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

Social policy making involved with technological innovations are the need of the hour for the current climatic crisis we all are going through. Deploying poorly judged technology with no view over sustainability can result in public reprisal, worsening climate crisis and or in less extreme cases, poor adoption rates among the masses, eventually failing. The need of the hour has been to anticipate these impacts correctly, without bias, and work alongside developers by shaping the eventual design of the technology. First step of which is to bring unanimous access to public data, concerned with national and climate interests.

## 1.1 Water Pollution Monitoring

It has been pretty evident as a fact that water pollution has been a critical issue not being handled with seriousness worldwide and the upcoming climatic crisis would add to the issue. As from the study published in 2017 by a greek journal[1], over 70
According to a report by the World Bank under 2030 water resource group[2], the river's basin covers ¼ of the geographical area and has over 450 million people with 60
In this thesis, we have discussed the primary parameters concerned with water quality and implementation for the same; assigning a fitness value to measure quality of water and have discussed the design for an unmoored water buoy with the usage flow for Lora with the web for viewing data from anywhere in the world with just an internet connection. Moreover, later as mentioned in the future work, we will open source our board layouts and the final 3D design for the structure to make it accessible to anyone who wants to implement the system under BSD-3 clause of open source licences. Our implementation is inclusive of any water body in any demographic concerned but for the sake of simplicity and an easier access to test, throughout the thesis, we will keep our discussion focussed over river Ganga as which is where failed projects have taken place since the last decade.

It is evident by the current water quality management systems implemented on the river data monitor only at a specific area along the course of the river. On river Ganga, for example, there are 36 Real-Time Water Pollution Monitoring locations. However, this has a fundamental drawback, that apart from monitoring the water quality sporadically,

it doesn't pinpoint the particular patch responsible for contamination through the flow. By a recent report published by WEF[4], reliable data and use of analytics to drive out essential insights using machine learning, can help overturn the water pollution issue. We propose capturing reliable data by using an untethered water buoy, inspiration of which has been taken from the hydrophone used in the Naval defense system, which here, will float with the current, sending water quality data continuously at a particular rate, being tracked of its location remotely through GPS and would be retract from its location after the required data has been transmitted or is let to relay the assigned score for water quality and location corresponding to it as the buoy floats through the required region. All the data received would be parallely parsed into our hosting servers to display on the website real time, over which later analysis and tests could be driven at to find a pattern, or simply be printed as a CSV file for governmental consumption.

Figure 1.1: Manually collecting sample for Water Pollution Measurement

As of now, on Indian rivers water pollution test are done sporadically with time taken more than five days with frequency of once a month. Tests are conducted on random samples and the reports take as long as five days with no real time location with quality of water being given. Ten years ago when internet observing innovation was all while creating, CPCB (Central Pollution Control Board) had done a comparative test which wasn't very fruitful with the observation. But now due to much more accessibility to the internet, setting up a network to view real time data is doable and makes analytics accessible.
The Central Pollution Control Board (CPCB) has set up a network of observing stations on rivers across the nation with 870 stations accross the country with water pollution test being done sporadically with time taken more than five days with frequency of once a month. Monitoring is done on a monthly or quarterly basis in surface waters. The water quality data are reported in Water Quality Statistics yearbooks. We would be covering the techniques and the premise used in the project below:

## 1.2 Classification of water

There are two major categories of water based on the source from where it is available. These are 'surface water' and 'groundwater'. However, both these sources are equally likely to get contaminated by pollutants through domestic, agricultural and industrial effluents. These pollutants include fertilizers/pesticides, soap water, waste discharge from washrooms, industrial hazardous chemical waste. Thus, on the basis of level of pollutants, the water is classified into four ways as follows:

1. **Contaminated water:** Water which has presence of harmful physical, biological, chemical or radiological substances. It is considered unfit for drinking as well as domestic purposes.

2. **Palatable water:** It comprises that water quality which does no harm on consumption. It lacks any kind of odour, taste and is very close to transparent on seeing.

3. **Potable water:** It is safe to drink and usable for domestic purposes. This water tastes a little sweet due to the presence of minerals.

4. **Infected water:** It is infected by harmful, disease-causing microorganisms.

## 1.3 LoRa


Figure 1.2: LoRa Logo

LoRa is short for Long-Range, is a communication medium for IoT (Internet of Things) gadgets. It is developed and patented by Semtech Technology. LoRa has special modulation scheme based on CSS. CSS, Chirp Sread Spectrum uses Chirp pulses to encode information using linear frequency wide band. Increasing linear frequency in time is called an up-chirp and decreasing linear frequency is called a down-chirp. Following is an example of LoRa modulated signal with up-chirps and down chirps.

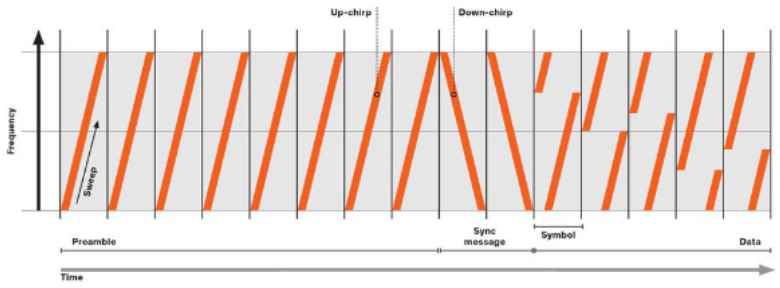


Figure 1.3: LoRa Modulated signal with up chirp and down chirp

Following is the preamble for a real signal.

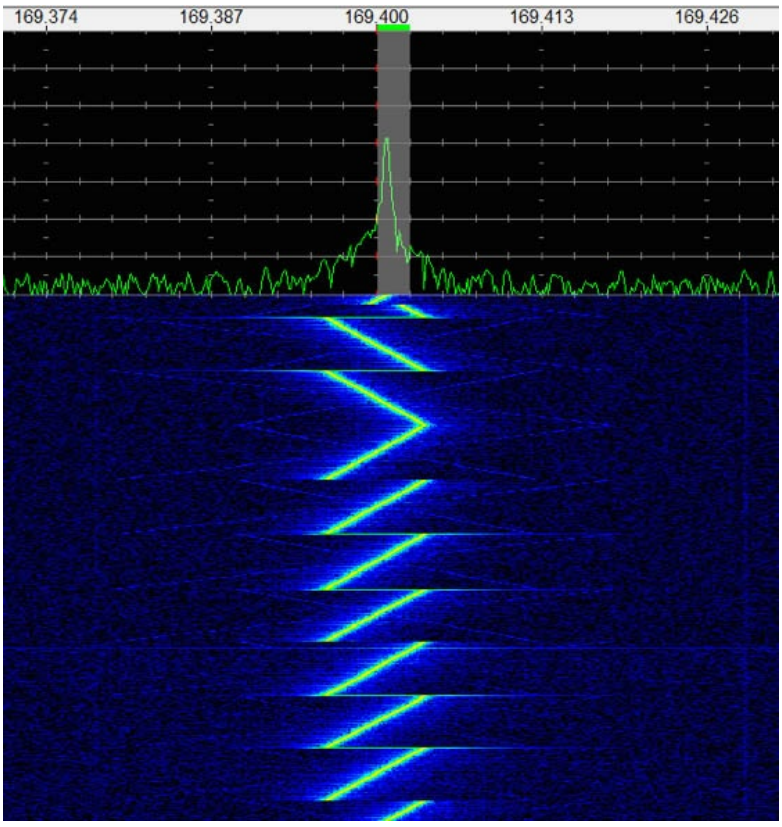


Figure 1.4: Preamble of real signal

The sweep rate is also called Spreading Factor. The European system uses 6 different sweep rates from 7 to 12. Higher spreading factor means it takes longer for one chirp to complete and so less data can be sent in a limited time but range of the communication is increased. On the contrary, lesser spreading factor implies more data rate but lesser range.


Figure 1.5: Various Spreading Factors

The devices equipped with this protocol are long-range and use low power to carry out receive, transmit over the spectrum without a license. They transmit and receive data over the unlicensed frequency spectrum. They can be best used when sensor data has to be transmitted at a low rate. There are lots of LoRa deployments worldwide, offered as free or paid services. LoRa operates in different bands in different parts of the world as shown in the table below:

Table 1.1: LoRa bands in different parts of the world

| **Region** | **Frequency** | **Regulation** |
|---|---|---|
| Europe | 863 to 870 MHz | CEPT Rec. 70-03 |
| USA | 902 to 928 MHz | None |
| China | 470-510 MHz and 779-787 MHz | None |
| India | 865.9850 MHz 865.0625 MHz, 865.4025 MHz | Use of low power wireless transmission in frequency band of 865 to 867MHz for RFID |

The LoRaWAN, having low power is protocol with wide-area coverage, intended to remotely inteface over the spectrum in commercial and educational usage without a license. Thus, good for IoT operation and projects operating on a battery. LoRaWAN addresses the network architecture: how devices connect to gateways, how gateways process the packets, and how packets make their way to network servers and application servers. A typical LoRaWAN structure is shown below.

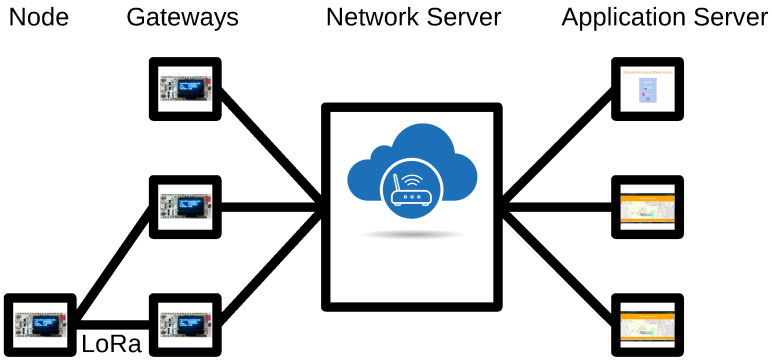


Figure 1.6: LoRa Network

There are three types of LoRaWAN devices out of which we are using a class A device.

1. **Class A:** Stands for “All”. These devices are battery powered and each device uplink to gateway followed by two small downlink windows for reception.

2. **Class B:** Stands for “Beacon”. These are the same as class A but at scheduled times, they provide with extra receive windows.

3. **Class C:** Stands for “Continuous”. These are the same as class A devices but they are continuously listening and hence they require more power and therefore class C devices are not battery powered.

There are various factors that affect the range of communication between node and gateway. One of the primary factors is the presence of obstacles like buildings and trees in the path. Buildings being more conductive in nature are poorer obstacles than trees. Best case scenario for LoRa communication is line of sight communication. Since a perfect line of sight communication is not entirely possible so it is important to consider the link budget of the communication. Link budget is the difference of maximum transmission power and minimum power required by receiving antenna to demodulate the signal and retrieve the message perfectly. This difference is basically the amount of power one can afford to lose in transmission to ensure perfect reception.

# CHAPTER 2

# THEORETICAL BACKGROUND

## 2.1 Microcontroller

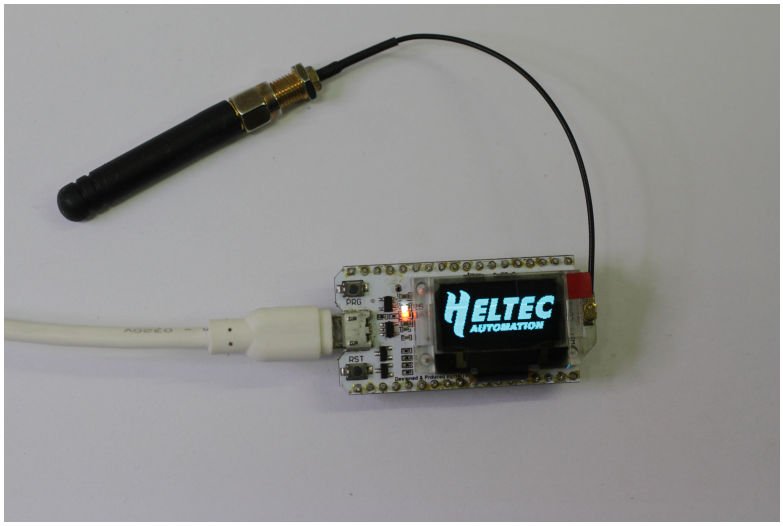


Figure 2.1: Heltec LoRa board

This project uses Heltec board that comprises of ESP32 Microcontroller and LoRa module. This module can be used as a node in a network or as a gateway to recieve data and forward the packet to website.

### 2.1.1 What is ESP32?

ESP32 is a 32-bit microcontroller with means for providing wireless connectivity: WiFi transceiver unit that uses 802.11 b/g/n protocol and a Bluetooth v4.2 unit. It comprises an Xtensa dual-core 32-bit LX6 microprocessor that can run from 160MHz to 240MHz clock rate with internal SRAM-520kB and ROM-448 kB. It can support upto 4 x 16 MB flash/SRAM via QSPI interface. It has Caches, MPU and MMU to manage on-chip memories, off-chip memories, and peripheral access. Support for BLE and other convetional Bluetooth protocols is provided, making it compatible with old devices with

old technology as well as newer ones. It has a sleep current of less than 5 μA, making it suitable for handheld applications.

### 2.1.2 Comparison with oher conventional alternatives

Table 2.1: ESP32 vs Arduino

| **ESP32** | **Arduino** |
|---|---|
| Tensilica Xtensa LX6 dual core microprocessor | Atmel AVR microcontroller |
| 32-bit | 8-bit |
| Can work in range between 160-240MHz | Works at 16MHz |
| Has on chip WiFi and Bluetooth facility | Newer boards require additional modules for wireless connectivity |
| Economically feasible for IoT projects | May require costly modules for IoT implementation. |

Table 2.2: ESP32 vs Raspberry Pi

| **ESP32** | **Raspi** |
|---|---|
| Tensilica Xtensa LX6 dual core processor | Quad-core Arm Cortex-A53 processor |
| 32-bit | 64-bit |
| Controlled and programmed using IDE by external PCs | Has its own OS. Attaching a keyboard and a mouse makes it an independent PC. |
| Has on chip WiFi and Bluetooth facility and on board Hall effect, capacitive touch and temperature sensor. | Has graphic and networks integrated chips, and HDMI, full speed WiFi, and possibility to add directly a keyboard and mouse, make it easy to embed a PC inside something. |
| Very cheap, around 3 dollars | Very Costly, 35 dollars and above |

### 2.1.3 Extra Features

This multifunction controller not only has general peripherals as other cheap controllers do, but also has embedded sensors:

1. -40°C to 125°C temperature sensor.
2. Capacitive-touch sensors on all GPIOs
3. Hall sensor

Some of the available peripherals are:

1. 34 GPIO pins.
2. 18 channels 12-bit SAR ADCs.
3. Two 8-bit DAC channels
4. MAC Interface
5. MMC/SD/SDIO host controller
6. SPI/SDIO Slave Controller
7. Three UART interfaces
8. Two I2C bus interfaces
9. Two standard I2S interfaces
10. Eight channels of IR remote transmitting and receiving Pulse Counter
11. Hardware accelerators
12. Pulse Width Modulation (PWM)
13. DMA Controller
14. Two timer groups having 2 x 64-bit timers

## 2.2 LoRa

LoRa, abbreviated for Long-Range, is a communication medium for IoT (Internet of Things) gadgets. It is developed and patented by Semtech Technology. LoRa has special modulation scheme based on CSS. CSS, Chirp Sread Spectrum uses Chirp pulses to encode information using linear frequency wide band.

### 2.2.1 Comparison to other standard technologies

Table 2.3: LoRa vs other conventional technologies

| **Protocol** | **Range** | **Operating Frequency** | **Data Rate** |
|---|---|---|---|
| Bluetooth | 100m | 2.4GHz | 3Mbps |
| Zigbee | 90m | 2.4GHz | 250kbps |
| WiFi | 80m | 2.4GHz, 5GHz | 60Mbps |

All these popular IoT protocols aim to provide higher data rate but at a cost of higher power consumption and range of not more than 100m.
LoRa and LoRaWAN protocols operate at subGHz with data rate upto 21kbps at SF7 and bandwidth of 500 KHz. Under ideal conditions i.e. line of sight communication, SF12 and perfectly tuned antenna, range of upto 10Km can be achieved. Unlike other

protocols LoRa provides an extremely long range of communication and With cosiderably lower data rate and lower current consumption. These characteristics make it ideal for IoT projects which require a long range of communication between sensors and gateways.

### 2.2.2 TTN: The Things Network

The Things Network provides a globally open-sourced free and decentralised platform for Internet of Things application with full security. This network allows us to connect to the web using negligible power and data. This network relies on LoRaWAN technology where many LoRa devices are connected to each other through a Gateway. A single LoRaWAN gateway can support upto ten thousand devices in a range of 10Kms. It provides a platform where the databases of the devices can be stored securely. The data stored on the network can then be accessed/viewed from anywhere around the world in an interactive user interface format. One can also keep their data private, having access to only limited devices.


Figure 2.2: Gateways provided by TTN

## 2.3 GPS

GPS (Global Positioning System) is a standard protocol to pin point 3d position is a system made up of 29 satellites of which 5 are reserved. It is weather proof available anywhere on the earth all the time for free
Theoretically, by using the time taken to backtrack the distance between each of the three satellites using 3-point system, it is able to pinpoint quite a precise user position in 2-D.
The receiver can determine the position with altitude, with more than 4 satellites in horizon(latitude, longitude and altitude). Using GPS, we can calculate the speed, trip distance and distance to destination. The frequency at which GPS works is:
L1 - 1,575.42 MHz and L2 - 1,227.60 MHz
Let's discuss about the GPS positioning system:
To improve accuracy: a fourth satellite is used to compute the relational position for the reciever.

### 2.3.1 GPS signals

Indoor: GPS signals have issues while catching signal inside the buildings. Can be accounted for.
Ionosphere: The ionosphere is the region between the thermosphere and the exosphere, about 45Km. In this region, the propagation velocity is lower, hence chances of error. Cannot be controlled.
Troposphere: This 10Km layer is 5Km above the ground level with no clouds hence having a drier atmosphere which causes radio reflections which results in GPS position error.
Multipath propagation: There is a probability of reflection of a GPS signal when it hits on the ground. This leads to multipath propagation resulting in wrong readings.
Strength of GPS hampers the reception which can happen due to presence of noise sources near the reciever. Sometimes, obstacles like buildings and trees can lower reception.
We are using NEO-6M GPS module for location tracking for our buoy system.
NEO-6M GPS tracks over 30 satellites and has a -161 dB tracking, consuming 50mA supply current. The system is capable of reducing power consumption. This dramatically reduces power consumption of the module to just 11mA. It supports baud rate uptill 230400bps; stadard rate = 9600bps.

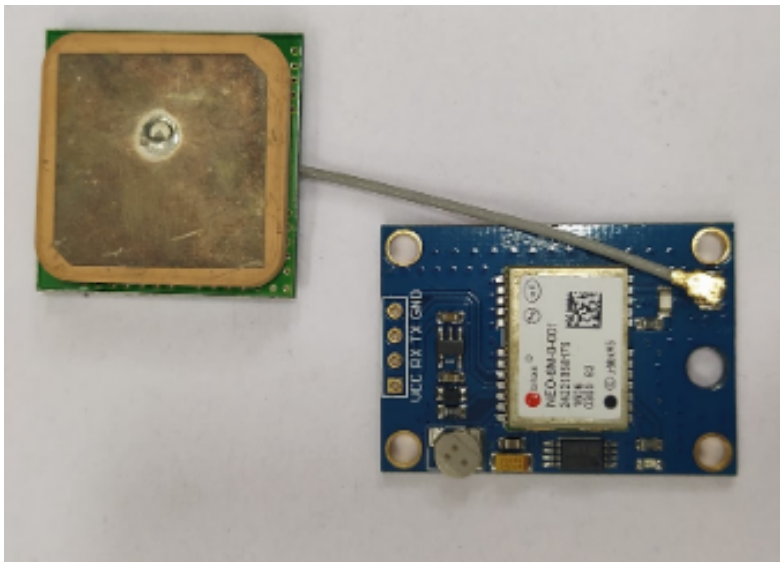


Figure 2.3: GPS module (Neo-6M)

Pinout:
GND – Ground Pin.
TxD – the transmitter pin for UART
RxD – the receiver pin for UART.
VCC – supplies power for the module(2.7V - 5V).

## 2.4 Electrical Conductivity

The amount of dissolved salts lead to salinity. But it's a known fact that these salts will disintegrate into ions and these ions then will rise to the ability of water to conduct electricity. Thus, salinity is a solid supporter of conductivity. So it can be also thought as the measure of a liquid to conduct electricity. Even though, salinity can be physically estimated by substance examination, this method is and tedious. We are using a probe to measure elctrical conductivity which gives us practical salinity, a standard farming parameter
Units of its measurement are as follows:
S.I. units = milliSiemens/m (mS/m)
where (mS/m) = 10 umho/cm (1000 uS/cm = 1 dS/m)

Standard conductivity of(in S/m):
Distilled water: 5.5 e-6;
Drinking water: 5e-3;
Seawater: 5;

The electrical conductivity can be used to estimate the TDS value of water as follows:
TDS(mg/L)=EC(dS/m or umho/cm)×(0.55–0.7)
TDS can be estimated using conductivity values by:
TDS = -0.7+EC*0.55
Over ten different ions including bromine, calsium, sulphate results in salinity. But, in the case of Ganga, for example, these salts are in much smaller quantities and the major thing mixed in the water is silt. In a more general context:
Most lakes and rivers: have chloride carbonates majorly
Fresh Water: higher bi-carbonate
Sea-water: higher Na+ concentration

### 2.4.1 The probe

Figure 2.4: Electrical Conductivity probe

### 2.4.2 Measurement

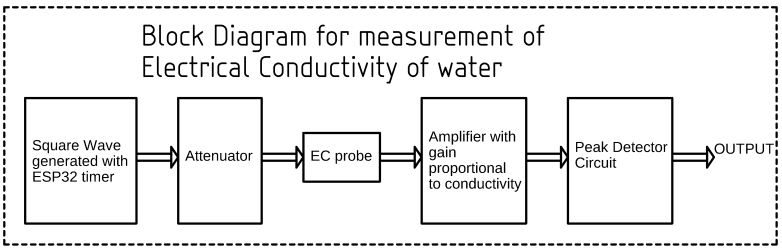


Figure 2.5: Electrical Conductivity measurement block diagram

The circuit starts by getting a square wave of 3.3V Vpp and attenuating it to 200mV. This signal is fed to an inverting amplifier, the gain of which is proportional to the conductivity of the probe. The measuring probe has two electrodes. The distance between these two are fixed and is equal to 5mm. Now when this alternating voltage is applied to the electrodes, the ions dissolved in the water will cause current to flow. (We gave this alternating voltage because a DC voltage causes polarisation of the probe and then it will become dysfunctional.) The more dissolved ions, higher is the conductivity and higher is the gain of the amplifier. This amplified voltage is passed through a peak detector circuit. The final output DC value corresponding to the conductivity of the water is read by ESP32 microcontroller.

## 2.5 pH

pH is a measurement of the acidity or alkalinity of water. It is derived from latin word 'potenz Hydrogen'. Levels of pH indicate hydrogen ion concentration in a liquid in a logarithmic scale. The pH scale goes from 0, for most acidic, to 14, for most basic. pH=-log[H+]. A safe level of pH for potable water lies somewhere between 6.5 to 8.5. pH levels higher than 8.5 become highly basic, while pH levels below 6.5 become highly acidic for potable water. A 1-unit change in pH represents a ten times change in the hydrogen ion concentration.

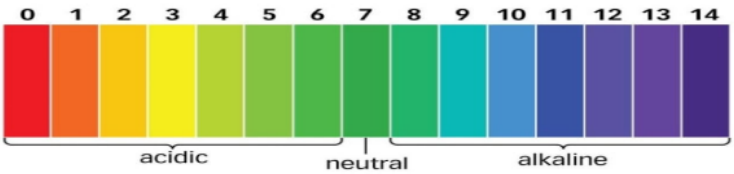


Figure 2.6: pH scale

The pH values in the river changes because of human activities, wastes of submerged plants and animals, industrial efflux. for example, have a more acidic level which directly impacts aquatic lifes and hinders. It may cause fatal consequences to aquatic life if they are exposed to high differences in pH with their tolerable value. In many rivers, fishes have been totally extinct. Excessively high and low pHs values of water can be harmful to a user too. A high pH water tastes bitter. It also decreases the efficiency of the chlorine disinfection process. The amount of oxygen in water increases as pH increases. Low-pH means acidic, which can corrode or dissolve metals.

The effect of pH levels on plants and animals in water can be summarized as: Most aquatic plants and animals have adjusted their bodies in water having a particular pH and they may suffer from even a slight pH change. Water with extremely low or high pH is lethal. pH level below 4 or above 10 may kill many fishes, and very few species of animals can tolerate water with a pH below 3 or above. If this comes in contact with human skin, it can cause severe burns, infections. Amphibians are extremely endangered by low pH water because their skin is very sensitive to contaminants. According to some scientists, reduction in numerous amphibians across globe is caused by acid rains. Heavy metals such as lead, chromium, and cadmium are easily dissolved in acidic solvent (water). This is hazardeous when many heavy metals gets dissolved in rivers. A change in the pH changes the forms of some chemicals in water. For instance, ammonia is harmless to animals in neutral or acidic water. But if the water becomes basic (as the pH increases), ammonia becomes toxic. Carbon dioxide from the atmosphere or from the respiration of aquatic organisms increases the acidity due to formation of carbonic acid.

### 2.5.1 The probe

pH probe has a semi permeable bulb at the bottom attached with two wires going inside the tube. The tube contains a buffer solution to normalise the pH and have a continuos H+ ion movement when the probe is dipped in the solution. Two pins of the probe are pH- and pH+. ph- is held at a constant voltage. Potential difference is produced at the ends of the probe according to the pH of the solution. This potential difference can be measured with the help of micro controller.

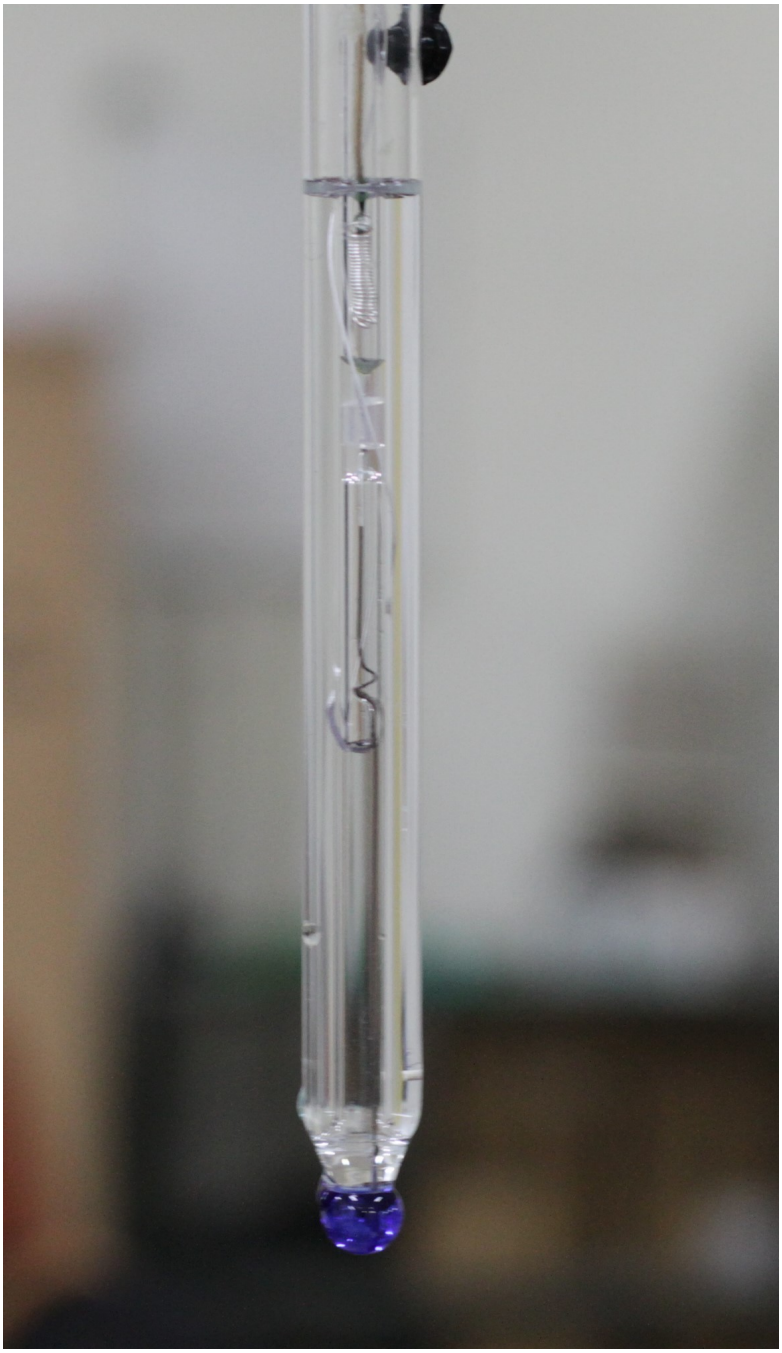

Figure 2.7: pH probe

### 2.5.2 Measurement

The measurement principle used here is the 'Potentiometric pH measurement', that is, to measure voltage equivalent to the pH of the solution. This comes from the Nernst equation that relates pH value to its equivalent potential difference between electrodes. Then converting this voltage back to pH by the processor/controller.
In a glass electrode sensor, which we have used, the pH sensitive element is attached to a glass bulb at the end of the glass tube. The glass tube is filled with a neutral solution of Potassium Chloride (buffered at pH 7). Along with it contains a silver chloride wire that forms an electrode. This tube is surrounded by another glass tube having a reference solution with silver chloride wire. These two are attached with a junction. When the tube is immersed in a liquid, the hydrogen ions penetrate ions in the liquid are able to penetrate the boundary area of the glass bulb, the same process occurs on the inside of the sensor having buffer solution. If there is a difference in the H+ between inside and outside, it creates a potential difference between the two electrodes. If outer concentration is more, the resulting solution is acidic with pH less than 7 and vice-versa. Thus, by measuring this potential difference and applying some suitable mathematical equation, we can get the pH value of the solution.
This method has a drawback that the output resistance of the measuring probe is very high (around 6-8M ohms). So a very less current should be drawn from the probe otherwise there will be a voltage drop due to loading effect. For this we have taken a special Op-amp with very less input bias current (around 15pA) and formed a voltage follower with it in the first stage.
Secondly, the potential difference can go upto 500mV with one of the electrodes at -0.5V and the other at 0V. Since our controller cannot read negative voltage, we have clamped one electrode to 2.5V. This will give us a voltage swing from 2V to 3V which can now be measured easily.

## 2.6 Turbidity

Turbidity is a measurement that tells how much clear (visually) or how much transparent the water is. With presence of suspended solids and colloidal particles, the light coming through the water gets obstructed and hence water starts to appear translucent. More the sediments in the water, the more turbid the water is. Along these lines, drinking water is low in turbidity in contrast to water in the lakes/ rivers/ ponds etc. Major sources of turbidity are wastewater discharge, soil erosion, urban runoff (rainwater flowing from paved surfaces into waterways), excessive algae growth, and eroding stream banks.
Keeping track of turbidity can be fruitful in monitoring increasing soil erosion from construction work, industrial wastes expulsion in the region.

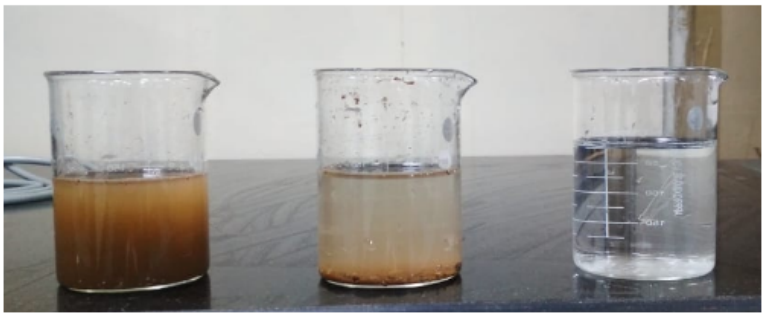

Figure 2.8: Most turbid(left), Mildly Turbid (Middle) and Clear Water (right)

The suspended particles also tend to absorb heat and thus increase the water temperature. This will lead to less dissolved oxygen content in water as warm water holds less oxygen. Thus, impacting water organisms sensitive to oxygen. Suspended materials such as organic products, silt, plankton, clay, and other particulate matters in water are responsible for high turbidity. Low turbidity in drinking water is acceptable.

The impact of turbidity can be summarized as:

High turbidity can increase the cost of water treatment, as more particles have to be filtered using various processes.

The particulates can shelter harmful microorganisms and thus protecting them from the disinfection process.

Suspended materials may clog or completely damage fish's gills, thereby, decreasing their immune system, making them more prone to diseases. It also reduces fertility rates, affecting their egg and maturation of larvae.

High turbidity increases water temperature and thus dissolved oxygen content decreases affecting aquatic life.

Turbidity is measured by 'Nephelometric Turbidimeter', expressing turbidity in units of NTU or TU. 1 mg/L of silica in a suspension is defined to be 1 TU.

Humans can easily identify turbidity of 5NTU or above. Turbidity of muddy water is more than 100 NTU. Groundwater has very low turbidity due to the natural filtering process by layers of soil.

### 2.6.1 The probe

Turbidity probe is simplest of all three, which has a transmitter and reciever LED. Higher amount of light recieved corresponds to low turbidity value.

Figure 2.9: Turbidity probe

### 2.6.2 Measurement

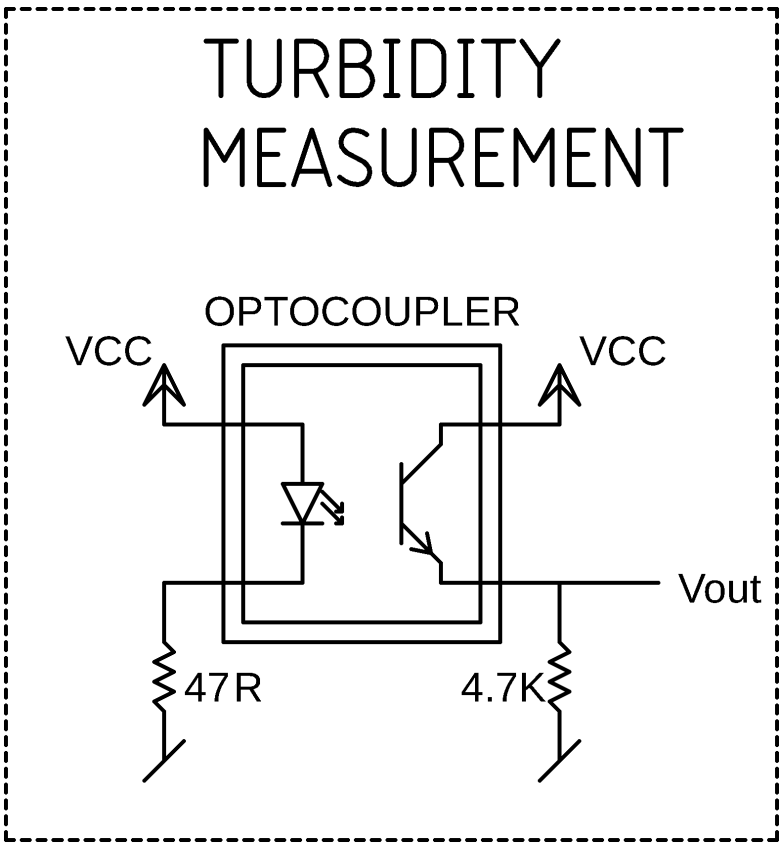


Figure 2.10: Circuit to measure Turbidity

To measure turbidity, light is passed on one end and its intensity is measured on the other end. Light intensity is directly proportional to current passing through the transistor. The amount of light received is inversely proportional to turbidity. Higher voltage received at the output implies low turbidity.

# CHAPTER 3

# IMPLEMENTATION AND APPROACH

As covered in the theoretical concepts, the individual sensors were then interfaced with the microcontroller and the receiver is separately handling the integration with the web app which we will discuss here in this subsection. Finally, we will also discuss the mechanical design and the related challenges we faced throughout the project.
Firstly, let's talk about the connection between the two LoRa modules. The Heltec boards were integrated and the range was tested by varying the spreading factor of the antenna.

## 3.1 LoRa Range testing

Connectivity between two LoRa modules was established and their range of communication was tested under different circumstances. Following are the results we obtained from it:

Table 3.1: LoRa Range Testing

| **Conditions** | **Range observed** |
|---|---|
| Spreading Factor 7: Antennas at right angle to each other | 300m |
| Spreading Factor 7: Antennas held upright | 420m |
| Spreading Factor 9: Antennas held upright | 600m |
| Spreading Factor 11: Antennas held upright | 750m |

## 3.2 Linking gateway to website

Initially, we used The Things Network (TTN) which provides public gateways and many other tools for IoT projects, at no cost to receive data on the web. We were able to receive data on it's servers and even interfaced the TTN gateway to receive data

on the same. But there were some of its limitations

1. There are not many gateways in India at present and so the coverage is extremely poor. So, we used Heltec ESP32 LoRa module board as the gateway.
2. This single channel gateway is a cost effective alternative and best suited for a project like this.
3. Another problem with TTN was that according to its fair access policy it allows only 30 seconds uplink time on air per 24 hours. For our usage, we require much more freedom with the uplink time.
4. Therefore, an alternative for TTN was required with all the features and a separate database with easy usage.

The best alternative to TTN was to build a website from scratch with a NoSQL database at the backend to store the data received from gateway.

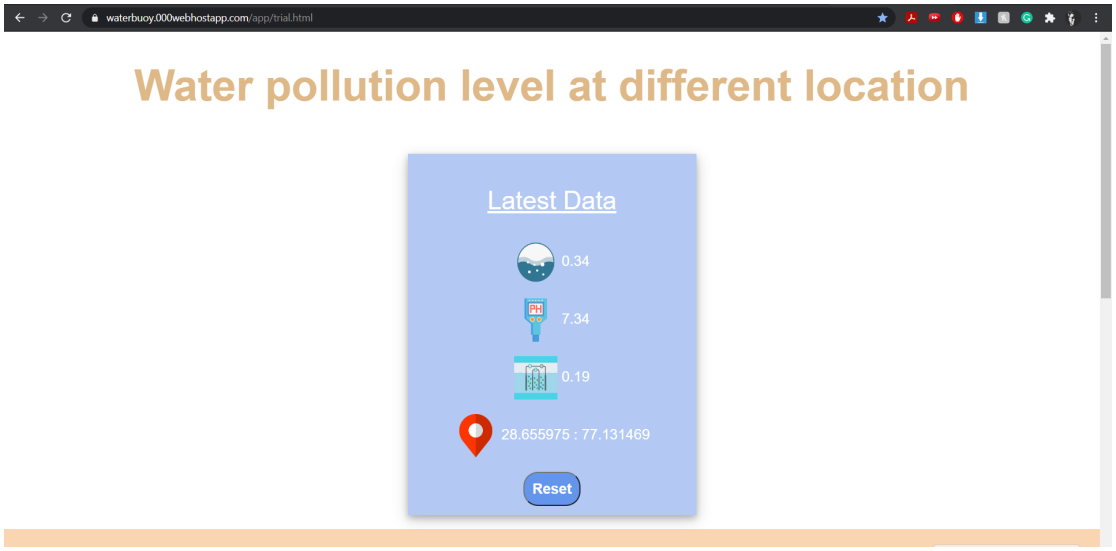


Figure 3.1: Last set of data recieved by gateway shown on website

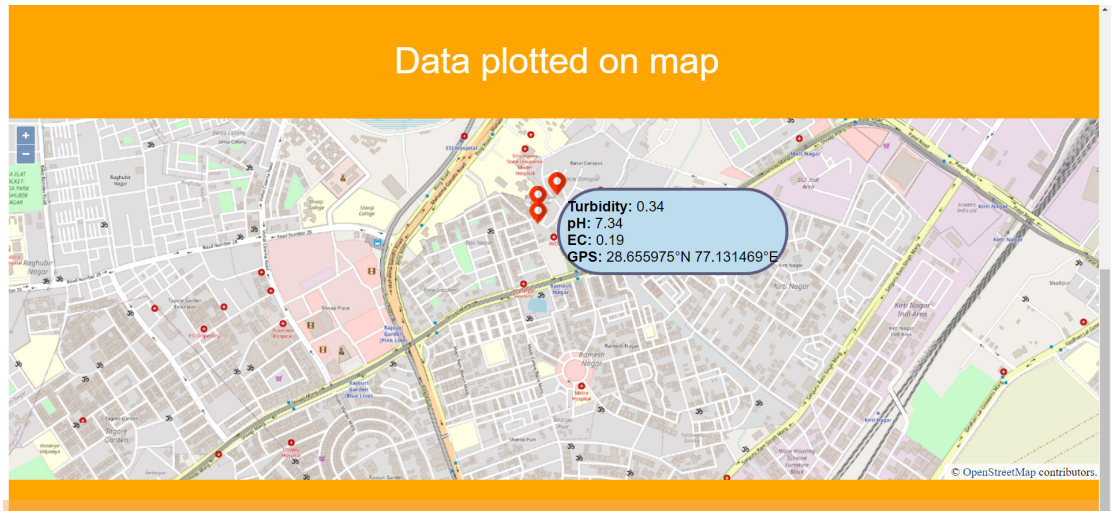


Figure 3.2: All the packets of data shown on map

1. The website shows the last set of values received from the node and also plots earlier sets of values on a map with the help of GPS data sent by the node in real time.

2. The gateway sends data by HTTP request handled in the database by parsing using Javascript.

3. The website shows the latest set of data received using which a marker is placed in real time on the map to pinpoint location of the Lora node.

4. When a marker is clicked, it pops up the details of pollution and the GPS coordinates of that location as shown below.

Now, once the connection with the web app was established, adding new sensor points is easy as only the objects are needed to be defined during the json parsing, which is like two words added in the command. Hence the modularity is achieved.

## 3.3 Interfacing sensors

As discussed in the last chapter, the four sensors were interfaced as shown in the images below.

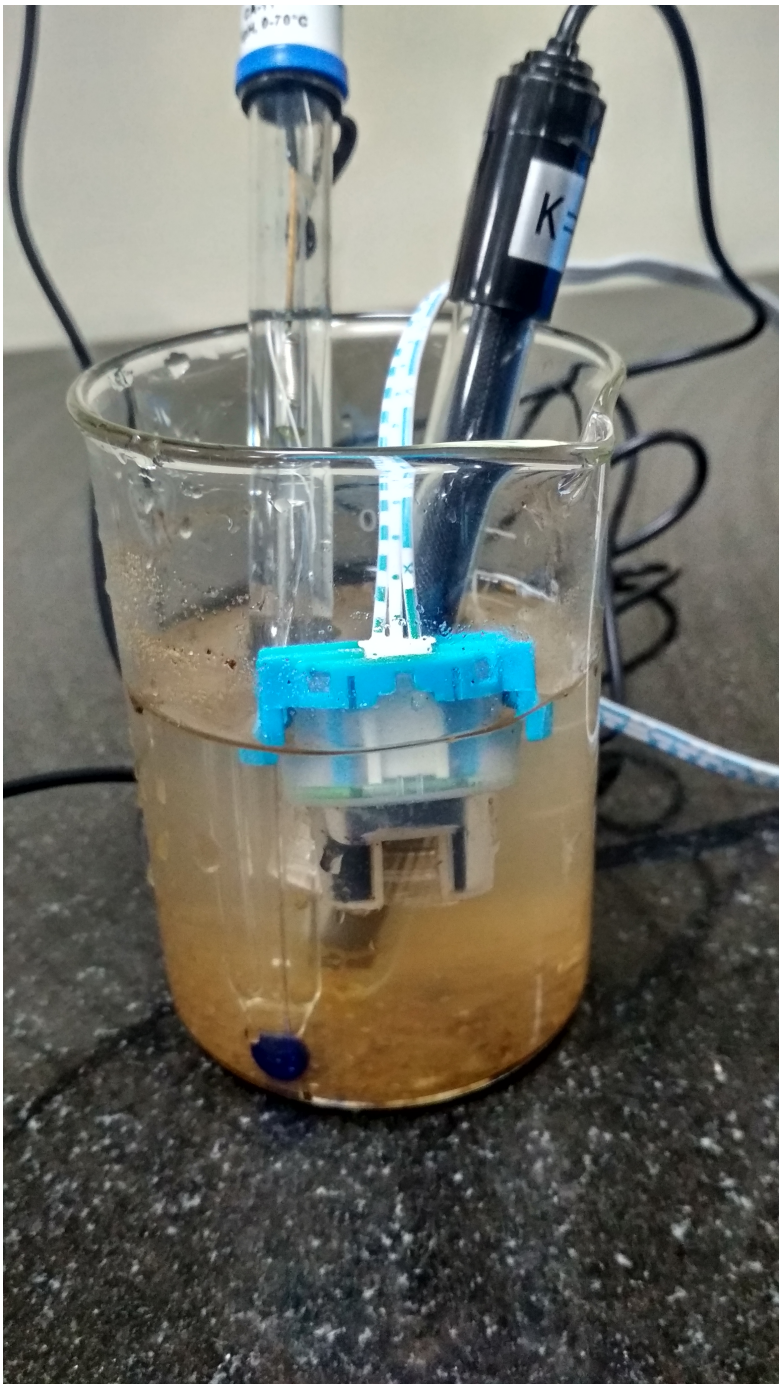

Figure 3.3: Final PCB containing, connectors of pH, turbidity, and conductivity

### 3.3.1 pH

The pH meter value was interfaced using a high precision low input bias current of 50pA using TL082 as a voltage buffer followed by a clamper circuit to be finally read by the ADC of our ESP32. Using 1st order curve fitting on data from a commercial available pH meter, the slope and intercept of the fitted line was calculated which gave slope to be -23.67 and 8.34 as intercept after calibration at 20 degree celsius.

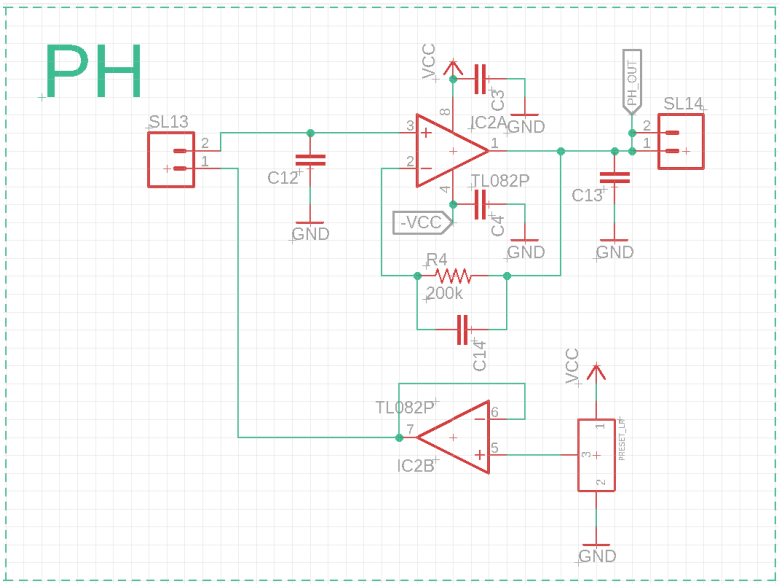


Figure 3.4: Circuit diagram for pH sensor

### 3.3.2 Conductivity

As discussed in the theoretical chapter, where interfacing was achieved. Below is the schematic for the same involving a square wave generation using a PWM pin on ESP32.

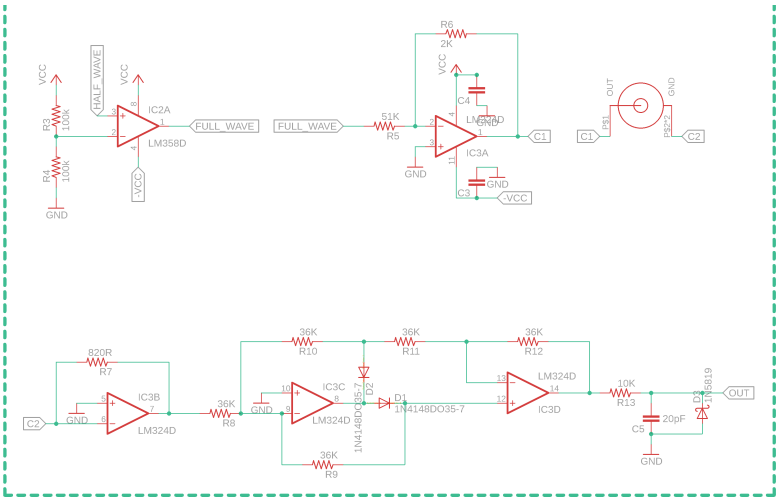


Figure 3.5: Electrical Conductivity Schematic

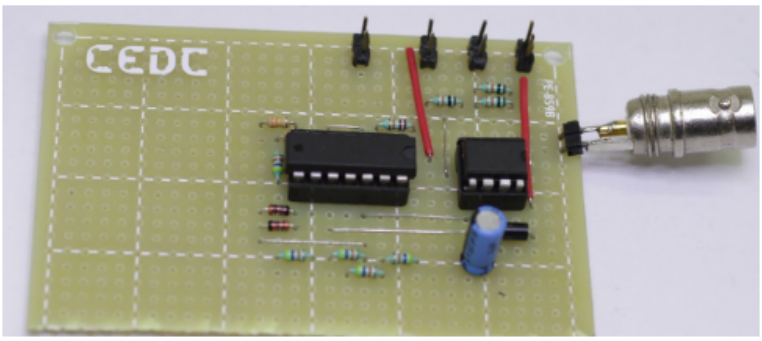


Figure 3.6: Electrical Conductivity Circuit Implemented

### 3.3.3 Turbidity

Turbidity, the simplest of the three sensor was interfaced using just a resistor to pull down at optocoupler pin and read directly by the ADC of ESP32. Below is the schematic for same.

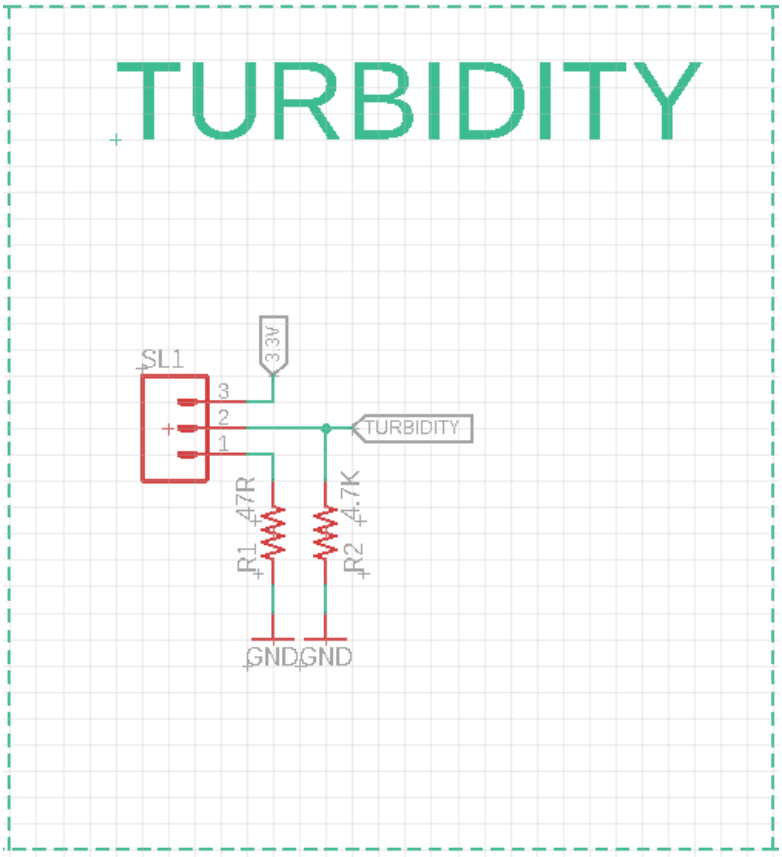


Figure 3.7: Circuit diagram for turbidity sensor

## 3.4 Mechanical Design

First, we had to test whether the 3D printed material floats and stands sturdy on water or not. Below is the 3D model picture where we have made the diameter to 8inch. We

have created this design on Blender, a 3D modelling software.

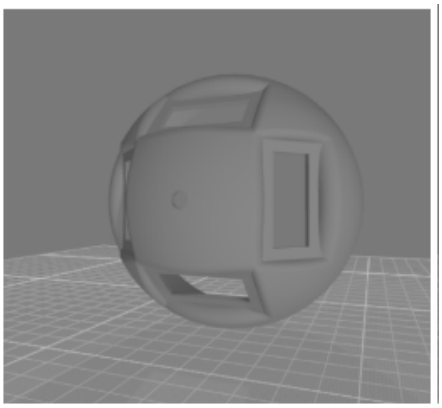

Figure 3.8: Upper Hemisphere

The open spaces seen in the upper hemisphere are for the solar panels.
These two hemispheres, upper and lower, are to be joined by threads, as we have seen on bottle caps. This will help in opening the buoy if there is any error/ problems in the circuit placed inside. This thread was added around the rim of the bottom hemisphere. The same thread was attached on the upper hemisphere and subtracted from the side walls so as to create a depression. We used the 'boolean' merging option provided in Blender.
Our 3D model should be light enough to float in water with electronic sensors and circuits residing in it. Upon estimating it, we need to find adequate volume of the buoy which will be able to float using archimedes' principle, where buoyant force (Fb) is equal to weight of liquid displaced by the object.

## 3.5 Final Implementation

The final implementation of the project has onboard GPS, pH and Turbidity sensor with a solar panel setup to charge the buoy during presence of sunlight. As you can see in the picture above, the first version of the prototype is meant to house inside the 3D buoydesign which is discussed below. The circuit for the sensors discussed in chapter 2 is integrated in a single power supply. The circuit is powered by 3.7V 200mAH LiPo battery boosted by BL8530 IC. The result of all sensor value can be seen on the OLED display of Heltec board. Below is the picture of final PCB for node.

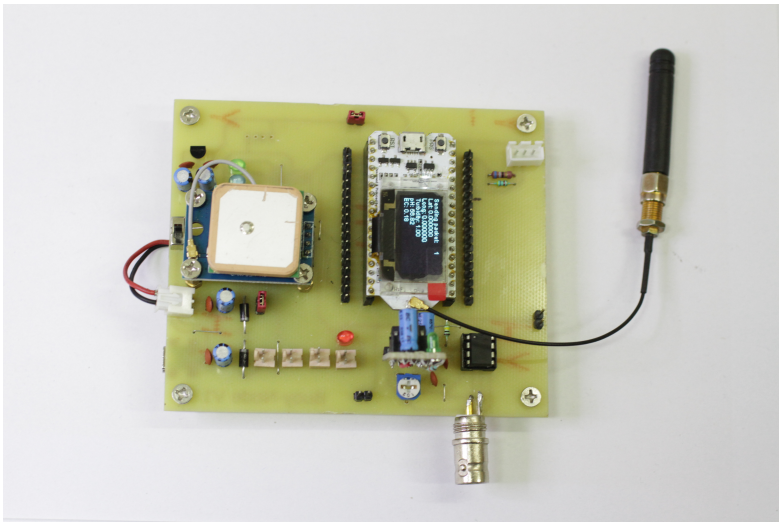

Figure 3.9: Final PCB containing, connectors of pH, turbidity, and conductivity

# CHAPTER 4

# RESULTS

## 4.1 Website

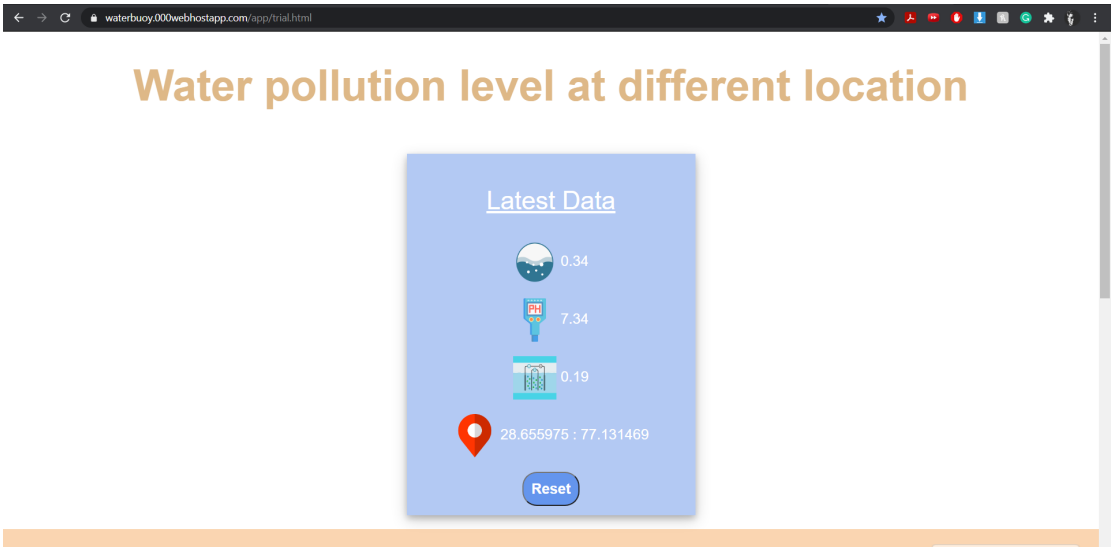


Figure 4.1: Last set of data recieved by gateway shown on website

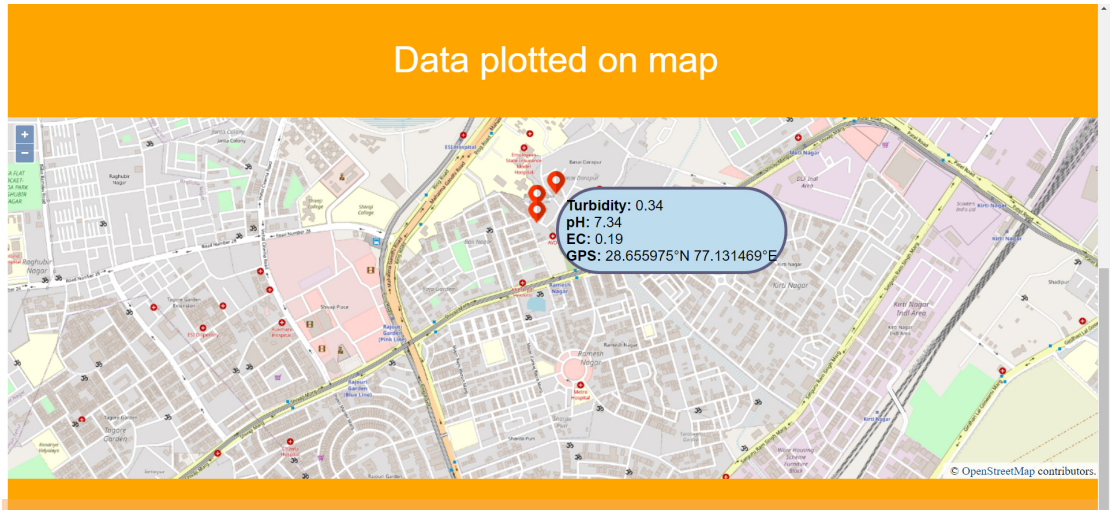


Figure 4.2: All the packets of data shown on map

.

## 4.2 pH

Table 4.1: pH meter result

| pH | Voltage Observed (V) |
|---|---|
| 0 | 0.358 |
| 1 | 0.306 |
| 2 | 0.242 |
| 3 | 0.197 |
| 4 | 0.158 |
| 5 | 0.109 |
| 6 | 0.053 |
| 7 | 0.0 |
| 8 | -0.046 |
| 9 | -0.111 |
| 10 | -0.142 |
| 11 | -0.212 |
| 12 | -0.239 |
| 13 | -0.313 |
| 14 | -0.347 |

These results were verified by a commercial pH meter as shown.


Figure 4.3: Verifying pH result with commercial pH meter

## 4.3 Electrical Conductivity

Results for electrical conductivity was observed with different amount of salt dissolved in 100mL water.

Table 4.2: Electrical conductivity result

| **salt(grams)** | **Conductivity Observed (mS/cm)** |
|---|---|
| 0 | 0.190 |
| 0.5 | 5.776 |
| 1.0 | 12.934 |
| 1.5 | 18.934 |

After dissolving 1.5 grams of salt in 100mL water, readings were saturated. These results were verified by a commercial electrical conductivity module as well.

# CHAPTER 5

# CONCLUSIONS AND FUTURE WORK

*'Art without engineering is dreaming. Engineering without art is calculating.'*

*Steven Robert*

## 5.1 Conclusions

To conclude, we hope our thesis implementation and the challenges faced can help demonstrate the usage of technological solutions to bring a more improved ecosystem for open access of public interest data like the climate changes effect and have an unbiased reporting of same.
In this project, an untethered water pollution monitoring system is implemented using ESP32 microcontroller based Heltec board that comes with LoRa module. LoRa technology was used for transmission of data gathered by the node to the network gateway. This Network gateway was implemented using same Heltec board based on ESP32 microcontroller. This gateway works with single channel for reception of data. The range of communication from one LoRa module to another was tested for various spreading factors. Various sensors, pH, Electrical conductivity and turbidity sensor were addded to the node to gather relevant information regarding water pollution measurement. The results of these sensors were verified by reliable commercial devices as well. Other than these, a GPS was added to track the location of node in real time. A website was developed to store the data received from the gateway. The website contains a world map showing the live location of the node using GPS coordinates.
We have obtained good results for pH measurement and electrical conductivity measurement. This kind of project can be used for regular monitoring of water bodies by the government to detect the root cause of pollution in the entire course of river.

## 5.2 Future Work

This project was intended to provide us with various water pollution details on the web and it was successfully achieved but there could have been some improvements for enhanced performance and presentation of the project.
There could have been many improvements regarding the designing of the structure to hold the PCB and various sensors. Some additional benefits could have been achieved in terms of network layout as well. Future work would require a deeper understanding of some sensors and their working, new ideas to try different methods or maybe a simple curiosity to improve. However, there are some ideas or improvements that we would have liked to try to improve the functionality and applicability of the project given the required time and opportunities.

1. Mesh Network of nodes could have been implemented along with the already implemented star network, to increase the range of data transmission with a drawback that nodes would be no longer untethered. There exist a open source project, named 'meshtatic' to create a mesh network from LoRa technology.

2. A proper robust enclosure could have been made for this project. It would allow the entire node with all the circuitries to float in water and send data to the gateway from there only. This enclosure should hold the PCB of the node, pH sensor probe, Electrical conductivity sensor probe, turbidity probe, solar panels and this structure should be waterproof to save electronics. It should also not be permanently closed so that changes can be made whever desired. These constraints increases the complexity of designing of enclosure.

3. Website could have been manipulated with additional features of showing and downloading CSV file of all the data recieved for better user experience.

# APPENDIX A

# CODES

## A.1 LoRa Node Transmission Code

```
#include <SoftwareSerial.h>
#include "heltec.h"
#include "images.h"
#include <TinyGPS++.h>
SoftwareSerial gpsSerial(23,17); //tx of the gps is connected to 23; rx of the gps is connected to 17
TinyGPSPlus gps;
float lattitude,longitude,turbidity;
unsigned int counter = 0;
#define BAND 868E6
#define turbPin 37
#define phpin 38
#define ecpin 39
#define LED_PIN 25
hw_timer_t * timer = NULL;
bool stat = true;
float ph;
float ec = 0.19;
void onTime() {
  stat = !stat;
  digitalWrite(LED_PIN, stat);
}

void logo()
{
 Heltec.display->clear();
 Heltec.display->drawXbm(0,5,logo_width,logo_height,logo_bits);
 Heltec.display->display();
}

void setup() {

```

```
32 gpsSerial.begin(9600);
33 Serial.begin(9600);
34 Heltec.begin(true /*DisplayEnable Enable*/, true
      ↪ /*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial
      ↪ Enable*/, true /*PABOOST Enable*/, BAND /*long BAND*/);
35 pinMode(LED_PIN, OUTPUT);
36   digitalWrite(LED_PIN, LOW);
37
38  timer = timerBegin(0, 40, true);
39   timerAttachInterrupt(timer, &onTime, true);
40
41   // Sets an alarm to sound every second
42   timerAlarmWrite(timer, 1000, true);
43   timerAlarmEnable(timer);
44  Heltec.display->init();
45  Heltec.display->flipScreenVertically();
46  Heltec.display->setFont(ArialMT_Plain_10);
47  logo();
48  delay(1500);
49  Heltec.display->clear();
50
51  Heltec.display->drawString(0, 0, "Heltec.LoRa Initial
      ↪ success!");
52  Heltec.display->display();
53  delay(1000);
54   }
55
56 void loop()
57 {
58 // while (!gpsSerial.available());
59  while (gpsSerial.available()){
60    int data = gpsSerial.read();
61    gps.encode(data);
62    delay(1);
63 // Serial.println("Hello!!!"); //this line helped before
      ↪ adding 1ms delay
64    }
65 // Serial.println("i am here!!!!!!");
66     lattitude = (gps.location.lat());
67     longitude = (gps.location.lng());
68     turbidity = float(analogRead(turbPin));
69     turbidity = 1.0 - (turbidity / 4095.0);
70     ph =
          ↪ (((float(analogRead(phpin))/4095.00)*3.30)-2.50+0.28);
71     ph = ph*(-23.042) + 8.076;
72     ec = (float(analogRead(ecpin))/4095.0)/0.2;
73     Serial.print ("lattitude: ");
74     Serial.println (lattitude,6);
75     Serial.print ("longitude: ");
76     Serial.println (longitude,6);
77     Serial.print ("Turbidity: ");
```

```
78      Serial.println(turbidity);
79      Serial.print ("pH: ");
80      Serial.println (ph);
81      Serial.print ("EC:");
82      Serial.println (ec);
83
84      LoRa.beginPacket();
85  LoRa.setTxPower(14,RF_PACONFIG_PASELECT_PABOOST);
86  LoRa.print(counter);
87  LoRa.print(":");
88  LoRa.print(lattitude,6);
89  LoRa.print(":");
90  LoRa.print(longitude,6);
91  LoRa.print(":");
92  LoRa.print(String(turbidity));
93  LoRa.print(":");
94  LoRa.print(String(ph));
95  LoRa.print(":");
96  LoRa.print(String(ec));
97  LoRa.print(":");
98  LoRa.endPacket();
99      Heltec.display->clear();
100 Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
101 Heltec.display->setFont(ArialMT_Plain_10);
102
103 Heltec.display->drawString(0, 0, "Sending packet: ");
104 Heltec.display->drawString(90, 0, String(counter));
105 Heltec.display->drawString(0, 10, "Lat:
    ↪ "+String(lattitude,6));
106 Heltec.display->drawString(0, 20, "Long:
    ↪ "+String(longitude,6));
107 Heltec.display->drawString(0, 30, "Turbidity:
    ↪ "+String(turbidity));
108 Heltec.display->drawString(0, 40, "pH: "+String(ph));
109 Heltec.display->drawString(0, 50, "EC: "+String(ec));
110 Heltec.display->display();
111 counter++;
112 delay(10000);
113 }
```

## A.2 LoRa Gateway Receiver Code

```
/*
 This is a simple example show the Heltec.LoRa recived data in OLED.

 The onboard OLED display is SSD1306 driver and I2C interface. In order to make the
 OLED correctly operation, you should output a high-low-high(1-0-1) signal by soft-
 ware to OLED's reset pin, the low-level signal at least 5ms.

 OLED pins to ESP32 GPIOs via this connecthin:
 OLED_SDA -- GPIO4
 OLED_SCL -- GPIO15
 OLED_RST -- GPIO16

 by Aaron.Lee from HelTec AutoMation, ChengDu, China
 www.heltec.cn

 this project also realess in GitHub:
 https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series
*/
#include "heltec.h"
#include "images.h"
#include <WiFi.h>

#define BAND 868E6 //you can set band here directly,e.g. 868E6,915E6
String rssi = "RSSI --";
String packSize = "--";
String packet ;

// Defining variable for each data to be recieved
String lati;
String longi;
String turbidity;

const char* ssid = "apoorv";
const char* password = "12345678";
const char* host = "waterbuoy.000webhostapp.com";
void logo(){
 Heltec.display->clear();
 Heltec.display->drawXbm(0,5,logo_width,logo_height,logo_bits);
 Heltec.display->display();
}

void sendData(){
 WiFiClient client;
 const int httpPort = 80;
 if (!client.connect(host, httpPort)) {
```

```
46    Serial.println("connection failed");
47    return;
48   }
49
50   String url = "/api/insert.php?lat=" + lati + "&lon=" + longi
       ↪ + "&turbidity=" + turbidity;
51   Serial.print("Requesting URL: ");
52   Serial.println(url);
53
54   client.print(String("GET ") + url + " HTTP/1.1\r\n" +
55                "Host: " + host + "\r\n" +
56                "Connection: close\r\n\r\n");
57   delay(500);
58
59   while(client.available()){
60    String line = client.readStringUntil('\r');
61    Serial.print(line);
62   }
63   delay(1000);
64 }
65
66 void LoRaData(){
67   Heltec.display->clear();
68   Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
69   Heltec.display->setFont(ArialMT_Plain_10);
70   Heltec.display->drawString(0 , 10 , "counter:- ");
71   Heltec.display->drawString(0 , 20 , "Lat:- ");
72   Heltec.display->drawString(0 , 30 , "Long:- ");
73   Heltec.display->drawString(0 , 40 , "Turbidity:- ");
74
75   int i=0;
76   char str_array[packet.length()];
77   packet.toCharArray(str_array, packet.length());
78   char *p = str_array;
79  char *str;
80  while ((str = strtok_r(p, ":", &p)) != NULL){ // delimiter is
       ↪ the semicolon
81    switch(i){
82     case 0: Heltec.display->drawString(50 , 10 , str);break;
83     case 1: Heltec.display->drawString(50 , 20 ,
          ↪ str);lati=str;break;
84     case 2: Heltec.display->drawString(50 , 30 ,
          ↪ str);longi=str;break;
85     default: Heltec.display->drawString(50 , 40 ,
          ↪ str);turbidity=str;
86    }
87    i++;
88 }
89   sendData();
90   Heltec.display->display();
91 }
```

```
92
93 void cbk(int packetSize) {
94  packet ="";
95  packSize = String(packetSize,DEC);
96  for (int i = 0; i < packetSize; i++) { packet += (char)
       ↪ LoRa.read(); }
97  rssi = "RSSI " + String(LoRa.packetRssi(), DEC) ;
98  LoRaData();
99 }
100
101 void setup() {
102   //WIFI Kit series V1 not support Vext control
103  Heltec.begin(true /*DisplayEnable Enable*/, true
       ↪ /*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial
       ↪ Enable*/, true /*PABOOST Enable*/, BAND /*long BAND*/);
104  Serial.begin(115200);
105  Serial.println("waiting...");
106  Heltec.display->init();
107  Heltec.display->flipScreenVertically();
108  Heltec.display->setFont(ArialMT_Plain_10);
109  logo();
110  delay(1500);
111  Heltec.display->clear();
112
113
114 // ------------------From the ESP's code to establish
       ↪ connection with the wifi network-----------
115
116  delay(100);
117  Serial.println();
118  Serial.println();
119  Serial.print("Connecting to ");
120  Serial.println(ssid);
121
122  WiFi.begin(ssid, password);
123  while (WiFi.status() != WL_CONNECTED) {
124    delay(500);
125    Serial.print(".");
126  }
127
128  Serial.println("");
129  Serial.println("WiFi connected");
130  Serial.println("IP address: ");
131  Serial.println(WiFi.localIP());
132  Serial.print("Netmask: ");
133  Serial.println(WiFi.subnetMask());
134  Serial.print("Gateway: ");
135  Serial.println(WiFi.gatewayIP());
136 // ---------------------ESP's wifi setup
       ↪ ends-----------------------------------
137
```

```
138   Heltec.display->drawString(0, 0, "Heltec.LoRa Initial
          ↪ success!");
139   Heltec.display->drawString(0, 10, "Wait for incoming
          ↪ data...");
140   Heltec.display->display();
141   delay(1000);
142   //LoRa.onReceive(cbk);
143   LoRa.receive();
144 }
145
146 void loop() {
147   int packetSize = LoRa.parsePacket();
148   if (packetSize) { cbk(packetSize); }
149   delay(10);
150 }
```

## A.3 HTML code for website

```
{% load static %}
<!DOCTYPE html>

<html lang="en">
   <head>
   <meta charset="UTF-8">
     <meta name="viewport" content="width=device-width,
        ↪ initial-scale=1.0"/>
     <meta name="Vikkey" content="Vivek Gupta & IoTMonk">
   <meta http-equiv="Access-Control-Allow-Origin" content="*">

  <!-- If you are opening this page from local machine,
     ↪ uncomment belwo line -->

  <script
     ↪ src="http://ajax.googleapis.com/ajax/libs/jquery/2.0.0/jquery.min.js"></scrip
    <!-- Openlayers CSS -->
  <link rel="stylesheet"
     ↪ href="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/en/v
     ↪ type="text/css">
   <!-- OpenLayers JS -->
    <script
       ↪ src="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/en/

   <!--<link rel="stylesheet" href="./ol.css"> -->
   <!-- If you are opening this page from a web hosting server
      ↪ machine, uncomment belwo line -->
   <!--
   <script type="text/javascript">
    document.write([
      "\<script src='",
      ("https:" == document.location.protocol) ? "https://" :
         ↪ "http://",
      "ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js'
         ↪ type='text/javascript'>\<\/script>"
    ].join(''));
    </script>
    -->
    <title>water pollution logging</title>
    <style>
  .footer{
    background:rgba(246, 171, 100,0.5);
    width:100%;
    height:50px;
    position: fixed;;
    bottom:0;
    left:0;
  }

    .center {
    height: 500px;
    width: 400px;
    background: #b3c9f3;
    position: absolute;
       box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0
```

```
        ↪ rgba(0, 0, 0, 0.19);
47   top: 50%;
48   left: 50%;
49   margin-top: -200px;
50   margin-left: -200px;
51  }
52
53  .form{
54   padding-top: 10px;
55   padding-right: 0px;
56   padding-bottom: 50px;
57   padding-left: 0px;
58  }
59  .map {
60      height: 500px;
61      width: 100%;
62     }
63  </style>
64   </head>
65  <body bgcolor="#ffffff">
66
67   <center>
68      <h1 style="font-family: Helvetica;color:
        ↪ burlywood;font-size: 60px;white-space:nowrap">Water
        ↪ pollution level at different location</h1>
69   </center>
70  <div class="center">
71    <div align="center" class="form">
72     <p style = 'line-height: 40px;font-family: Helvetica;color:
        ↪ rgb(255, 255, 255);font-size: 35px;text-decoration:
        ↪ underline;'>
73      Latest Data
74     </p>
75          <p style = 'line-height: 40px;font-family:
            ↪ Helvetica;color: #fff;font-size: 20px;'
            ↪ id="turbidity">
76      <img src = 'turbidity.png' height="40px" width="40px"
            ↪ style='vertical-align: middle' /> 00.00
77     </p>
78     <p style = 'line-height: 40px;font-family: Helvetica;color:
        ↪ #fff;font-size: 20px;' id="ph">
79      <img src = 'ph.png' height="40px" width="40px"
            ↪ style='vertical-align: middle' /> 00.00
80     </p>
81     <p style = 'line-height: 40px;font-family: Helvetica;color:
        ↪ #fff;font-size: 20px;' id="ec">
82      <img src = 'ec.jpg' height="40px" width="40px"
            ↪ style='vertical-align: middle' /> 00.00
83          </p>
84          <p style = 'line-height: 40px;font-family:
            ↪ Helvetica;color: #fff;font-size: 20px;' id="gps">
85             <img src = 'gps.png' height="40px" width="40px"
                ↪ style='vertical-align: middle' /> 00.00
86     </p>
87
88  <button id="reset" style="background-color:
     ↪ cornflowerblue;color: white;font-family: Helvetica;
     ↪ font-size: 20px; border-radius: 20px;padding: 10px
```

```
    ↪ 10px;"><b>Reset</b></button>
89        </div>
90    </div>
91    <!--An orange div containing heading and map-->
92    <div style = 'background: orange; top: 100%; width:
        ↪ 100%;height: 100%; position: absolute'>
93      <div align="center">
94        <p style = 'line-height: 60px;font-family: Helvetica;color:
            ↪ rgb(255, 255, 255);font-size: 50px;'>
95          Data plotted on map
96        </p>
97      </div>
98      <div id="map" class="map"> </div>
99      <div id="popup" class="ol-popup">
100       <a href="#" id="popup-closer" class="ol-popup-closer"></a>
101       <div id="popup-content" style='background:rgb(189, 220,
            ↪ 238);border-radius: 100px;border: 4px solid rgb(93,
            ↪ 93, 128);padding: 10px 10px;font-family:
            ↪ Helvetica;font-size: 20px;'></div>
102     </div>
103     </div>
104    <!--a fixed footer-->
105    <footer class="footer">
106       <h6 style="font-family: Helvetica;color:black;text-align:
            ↪ right;">Apoorv Aryan, Swapniil Kanwar, Harsh Sharma
            ↪ .</h4>>
107    </footer>
108
109    </body>
110
111    <!--script for data display every 5 seconds-->
112    <script>
113      window.onload = function() {
114            loaddata();
115      };
116      var lat;
117      var lon;
118      var turbidity;
119      var oldid = null;
120      var Delhiview = new ol.View({
121    center: ol.proj.fromLonLat([77.1025,28.7041]),
122    zoom: 6
123  });
124
125  //BaseLayer
126  var baselayer = new ol.layer.Tile({
127     source: new ol.source.OSM()
128    });
129
130  var markerSource = new ol.source.Vector({
131    // It is empty initially but later it will be filled with
        ↪ markerSource.addFeature(markerfeature) and it can be
        ↪ removed with markerSource.Clear()
132  });
133
134
135
136
```

```
137 //Markerlayer
138 var markerlayer = new ol.layer.Vector({
139    source: markerSource
140 });
141
142  var map = new ol.Map({
143    target: "map"
144  });
145  map.setView(Delhiview);
146  map.addLayer(baselayer);
147  map.addLayer(markerlayer);
148
149  var iconStyle = new ol.style.Style({
150    image: new ol.style.Icon(/** @type {olx.style.IconOptions} */
         ↪ ({
151     anchor: [0.5, 32],
152     anchorXUnits: 'fraction',
153     anchorYUnits: 'pixels',
154     src: './gpsicon.png'
155    }))
156  });
157
158 // markerfeature.setStyle(iconStyle);
159 // markerSource.addFeature(markerfeature);
160
161
162    //-------------------------------------------------------popup-------------
163 var container = document.getElementById('popup');
164  var content = document.getElementById('popup-content');
165  var closer = document.getElementById('popup-closer');
166
167  var overlay = new ol.Overlay({
168     element: container,
169     autoPan: true,
170     autoPanAnimation: {
171        duration: 250
172     }
173  });
174  map.addOverlay(overlay);
175
176  closer.onclick = function() {
177     overlay.setPosition(undefined);
178     closer.blur();
179     return false;
180  };
181
182  map.on('singleclick', function (event) {
183   // if (map.hasFeatureAtPixel(event.pixel) === true) {
184   var selectedfeature = map.forEachFeatureAtPixel(event.pixel,
185    function(feature) {
186     return feature;
187    });
188    if (selectedfeature){
189     var coordinate = event.coordinate;
190     content.innerHTML = "<b>Turbidity: </b>"
         ↪ +selectedfeature.get('turb') + "<br><b> pH:</b> " +
         ↪ selectedfeature.get('pH') +"<br><b> EC:</b> " +
         ↪ selectedfeature.get('EC') +"<br><b> GPS:</b> " +
```

```
        ↪ selectedfeature.get('gps') ;
191     overlay.setPosition(coordinate);
192   } else {
193     overlay.setPosition(undefined);
194     closer.blur();
195   }
196 });
197 // ----------------------------------------to truncate the
    ↪ database----------------------------------------
198   document.getElementById('reset').onclick = function () {
199     var truncate =
          ↪ "https://waterbuoy.000webhostapp.com/api/truncatetable.php";
200     $.getJSON(truncate, function(data) {});
201     };
202 //-------------------------------------------------------------------------------
203   function loaddata(){
204     var url =
          ↪ "https://waterbuoy.000webhostapp.com/api/read_all.php";
205     $.getJSON(url, function(data) {
206     var val= data;
207     var
          ↪ id=(data['pollutionlevel'][(Object.keys(data['pollutionlevel']).length)-
208     lat=(data['pollutionlevel'][(Object.keys(data['pollutionlevel']).length)-1]['l
209     lon=(data['pollutionlevel'][(Object.keys(data['pollutionlevel']).length)-1]['l
210     turbidity=(data['pollutionlevel'][(Object.keys(data['pollutionlevel']).length)
211     ph =
          ↪ (data['pollutionlevel'][(Object.keys(data['pollutionlevel']).length)-1][
212     ec =
          ↪ (data['pollutionlevel'][(Object.keys(data['pollutionlevel']).length)-1][
213       document.getElementById("turbidity").innerHTML =
            ↪ "<img src = 'turbidity.png' height=\"60px\"
            ↪ width=\"60px\" style='vertical-align: middle'
            ↪ /> " +turbidity;
214     document.getElementById("ph").innerHTML = "<img src =
          ↪ 'ph.png' height=\"60px\" width=\"60px\"
          ↪ style='vertical-align: middle' /> " +ph;
215     document.getElementById("ec").innerHTML = "<img src =
          ↪ 'ec.jpg' height=\"60px\" width=\"60px\"
          ↪ style='vertical-align: middle' /> " +ec;
216     document.getElementById("gps").innerHTML = "<img src =
          ↪ 'gps.png' height=\"60px\" width=\"60px\"
          ↪ style='vertical-align: middle' /> " +lat+" : "+lon;
217     console.log(data['pollutionlevel'][(Object.keys(data['pollutionlevel']).length
218
219     var mypoint = new
          ↪ ol.geom.Point(ol.proj.fromLonLat([parseFloat(lon),
          ↪ parseFloat(lat)]));
220     var markerfeature = new ol.Feature({
221       geometry: mypoint,
222       turb: parseFloat(turbidity),
223     gps: parseFloat(lat) + "N " + parseFloat(lon) + "E",
224     pH: parseFloat(ph),
225     EC: parseFloat(ec)
226     });
227     if(oldid!=id){
228     markerfeature.setStyle(iconStyle);
229     markerSource.addFeature(markerfeature);
230     }
```

```
231         oldid=id;
232         });
233         }
234         window.setInterval(function(){
235         loaddata();
236            }, 10000);
237
238   </script>
239 </html>
```

## A.4 PHP code to connect to server

```php
<?php
class DB_CONNECT {

   // Constructor
   function __construct() {
      // Trying to connect to the database
      $this->connect();
   }

   // Destructor
   function __destruct() {
      // Closing the connection to database
      $this->close();
   }

  // Function to connect to the database
   function connect() {
      //importing dbconfig.php file which contains database
         ↪ credentials
      $filepath = realpath (dirname(__FILE__));
      require_once($filepath."/dbconfig.php");

     // Connecting to mysql (phpmyadmin) database
      $con = mysql_connect(DB_SERVER, DB_USER, DB_PASSWORD) or
         ↪ die(mysql_error());

      // Selecing database
      $db = mysql_select_db(DB_DATABASE) or die(mysql_error())
         ↪ or die(mysql_error());

      // returing connection cursor
      return $con;
   }

  // Function to close the database
   function close() {
      // Closing data base connection
      mysql_close();
   }

}

?>
```

## A.5 PHP code containing configuration for connection

```php
<?php
/**************************************************/
/************** Created by : Apoorv Aryan ************/
/**************************************************/
define('DB_USER', "id15017330_water_buoy"); // Your database
    ↪ user name
define('DB_PASSWORD', "72zb}/#1dQUiYzA&");   // Your database
    ↪ password (mention your db password here)
define('DB_DATABASE', "id15017330_waterbuoy"); // Your
    ↪ database name
define('DB_SERVER', "localhost");   // db server (Mostly will
    ↪ be 'local' host)
?>
```

## A.6 PHP code for deletion from database

```php
<?php

header('content-type: application/json; charset=utf-8');
header("access-control-allow-origin: *");

//Creating Array for JSON response
$response = array();

// Check if we got the field from the user
if (isset($_GET['id'])) {
    $id = $_GET['id'];

    // Include data base connect class
    $filepath = realpath (dirname(__FILE__));
   require_once($filepath."/db_connect.php");

    // Connecting to database
    $db = new DB_CONNECT();

    // Fire SQL query to delete pollution level data by id
    $result = mysql_query("DELETE FROM pollutionlevel WHERE id
        ↪ = $id");

    // Check for succesfull execution of query
    if (mysql_affected_rows() > 0) {
        // successfully deleted
        $response["success"] = 1;
        $response["message"] = "Data successfully deleted";

        // Show JSON response
        echo json_encode($response);
    } else {
        // no matched id found
        $response["success"] = 0;
        $response["message"] = "No pollution level data found by
            ↪ given id";

        // Echo the failed response
        echo json_encode($response);
    }
} else {
    // If required parameter is missing
    $response["success"] = 0;
    $response["message"] = "Parameter(s) are missing. Please
        ↪ check the request";

    // Show JSON response
    echo json_encode($response);
}
```

```
47 ?>
```

## A.7 PHP code for insertion to database

```php
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
//Creating Array for JSON response
$response = array();

// Check if we got the field from the user
if (isset($_GET['lat']) && isset($_GET['lon']) &&
    ↪ isset($_GET['turbidity']) && isset($_GET['ph']) &&
    ↪ isset($_GET['ec'])) {

    $lat = $_GET['lat'];
    $lon = $_GET['lon'];
    $turbidity = $_GET['turbidity'];
    $ph = $_GET['ph'];
    $ec = $_GET['ec'];

    // Include data base connect class
    $filepath = realpath (dirname(__FILE__));
   require_once($filepath."/db_connect.php");

    // Connecting to database
    $db = new DB_CONNECT();

    // Fire SQL query to insert data in weather
    $result = mysql_query("INSERT INTO
       ↪ pollutionlevel(lat,lon,turbidity,ph,ec)
       ↪ VALUES('$lat','$lon','$turbidity','$ph','$ec')");

    // Check for succesfull execution of query
    if ($result) {
       // successfully inserted
       $response["success"] = 1;
       $response["message"] = "pollution level logged.";

       // Show JSON response
       echo json_encode($response);
    } else {
       // Failed to insert data in database
       $response["success"] = 0;
       $response["message"] = "Something has been wrong";

       // Show JSON response
       echo json_encode($response);
    }
} else {
    // If required parameter is missing
    $response["success"] = 0;
```

```
45      $response["message"] = "Parameter(s) are missing. Please
            ↪ check the request";
46
47      // Show JSON response
48      echo json_encode($response);
49  }
50  ?>
```

## A.8 PHP code to read all entries from database

```php
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
//Creating Array for JSON response
$response = array();

// Include data base connect class
$filepath = realpath (dirname(__FILE__));
require_once($filepath."/db_connect.php");
 // Connecting to database
$db = new DB_CONNECT();

 // Fire SQL query to get all data from pollutionlevel
$result = mysql_query("SELECT *FROM pollutionlevel") or
    ↪ die(mysql_error());

// Check for succesfull execution of query and no results found
if (mysql_num_rows($result) > 0) {

  // Storing the returned array in response
   $response["pollutionlevel"] = array();

  // While loop to store all the returned response in variable
   while ($row = mysql_fetch_array($result)) {
      // temperoary user array
      $pollutionlevel = array();
      $pollutionlevel["id"] = $row["id"];
      $pollutionlevel["lat"] = $row["lat"];
      $pollutionlevel["lon"] = $row["lon"];
      $pollutionlevel["turbidity"] = $row["turbidity"];
      $pollutionlevel["ph"] = $row["ph"];
      $pollutionlevel["ec"] = $row["ec"];
     // Push all the items
      array_push($response["pollutionlevel"], $pollutionlevel);
   }
   // On success
   $response["success"] = 1;

   // Show JSON response
   echo json_encode($response);
}
else
{
   // If no data is found
  $response["success"] = 0;
   $response["message"] = "No data on pollutionlevel found";

   // Show JSON response
   echo json_encode($response);
```

```
49 }
50 ?>
```

## A.9 PHP file to read a specific entry from database

```php
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
//Creating Array for JSON response
$response = array();

// Include data base connect class
$filepath = realpath (dirname(__FILE__));
require_once($filepath."/db_connect.php");
 // Connecting to database
$db = new DB_CONNECT();

// Check if we got the field from the user
if (isset($_GET["id"])) {
   $id = $_GET['id'];

    // Fire SQL query to get pollutionlevel data by id
   $result = mysql_query("SELECT *FROM pollutionlevel WHERE id
      ↪ = '$id'");

  //If returned result is not empty
   if (!empty($result)) {
      // Check for succesfull execution of query and no
         ↪ results found
      if (mysql_num_rows($result) > 0) {

       // Storing the returned array in response
         $result = mysql_fetch_array($result);

       // temperoary user array
         $pollutionlevel = array();
         $pollutionlevel["id"] = $result["id"];
         $pollutionlevel["lat"] = $result["lat"];
         $pollutionlevel["lon"] = $result["lon"];
       $pollutionlevel["turbidity"] = $result["turbidity"];
         $pollutionlevel["ph"] = $result["ph"];
         $pollutionlevel["ec"] = $result["ec"];

         $response["success"] = 1;
         $response["pollutionlevel"] = array();

       // Push all the items
         array_push($response["pollutionlevel"],
            ↪ $pollutionlevel);

         // Show JSON response
         echo json_encode($response);
      } else {
         // If no data is found
```

```
47          $response["success"] = 0;
48          $response["message"] = "No data on pollutionlevel
               ↪ found";
49
50          // Show JSON response
51          echo json_encode($response);
52       }
53    } else {
54       // If no data is found
55       $response["success"] = 0;
56       $response["message"] = "No data on pollutionlevel found";
57
58       // Show JSON response
59       echo json_encode($response);
60    }
61 } else {
62    // If required parameter is missing
63    $response["success"] = 0;
64    $response["message"] = "Parameter(s) are missing. Please
         ↪ check the request";
65
66    // echoing JSON response
67    echo json_encode($response);
68 }
69 ?>
```

## A.10 PHP code to truncate database

```php
<?php

header('content-type: application/json; charset=utf-8');
header("access-control-allow-origin: *");

//Creating Array for JSON response
$response = array();

   // Include data base connect class
   $filepath = realpath (dirname(__FILE__));
  require_once($filepath."/db_connect.php");

   // Connecting to database
   $db = new DB_CONNECT();

   // Fire SQL query to delete pollution level data by id
   $result = mysql_query("TRUNCATE TABLE pollutionlevel");

      // successfully deleted
      $response["success"] = 1;
      $response["message"] = "Table Successfully Truncated";

      // Show JSON response
      echo json_encode($response);
?>
```

# APPENDIX B

# LIST OF RESOURCES

The list of datasheets and necessary reference material associated with the project.

1. **Heltec board:** https://heltec.org/project/wifi-kit-32/
2. **ESP32:** https://www.espressif.com/sites/default/files/documentatio esp32_technical_reference_manual_en.pdf https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
3. **Adafruit Oled Display:** https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf
4. **TL082:** https://www.ti.com/lit/ds/symlink/tl082-n.pdf?ts=1609499920262&ref_url=https%253A%252F%252Fwww.google.com%252F
5. **pH probe:** https://robu.in/product/analog-ph-sensor-kit/
6. **LM358:** https://www.ti.com/lit/gpn/lm358-n
7. **GPS Module:** https://components101.com/sites/default/files/component_datasheet/NEO6MV2%20GPS%20Module%20Datasheet.pdf
8. **Conductivity Probe:** https://wiki.dfrobot.com/Gravity__Analog_Electrical_Conductivity_Sensor___Meter_V2__K%3D1__SKU_DFR0300#More_Documents

# REFERENCES


[1] https://www.veluda.com/en/blog/15-endiaferon-dedomena-gia-ti-mo

[2] .https://www.2030wrg.org/india/background/#:~:
text=Water%20Challenges,on%20agriculture%20for%
20their%20livelihoods.

[3] https://www.thehindu.com/news/national/
rivers-in-india-a-reality-check/article27510799.ece/
photo/1/

[4] https://www.weforum.org/agenda/2019/10/
water-pollution-in-india-data-tech-solution/

[5] http://www.who.int/water_sanitation_health/dwq/
chemicals/tds.pdf

[6] https://www.downtoearth.org.in/news/
instant-quality-check-of-rivers-33502

[7] .BiologicalWaterqualityassessmentofriverGanga:https:
//factchecker.in/wp-content/uploads/2019/04/CPCB_
Ganga.pdf

[8] .http://www.sulabhenvis.nic.in/Database/
WaterQualityStatus_6984.aspx

[9] https://devopedia.org/lora

[10] https://www.semtech.com/lora/what-is-lor

[11] 'WaterqualitymonitoringinIndia:Areview\OT1\
textquoterightbySnehGangwar,DeptofGeography,
DelhiUniversity:http://www.irphouse.com/ijict_spl/
17_ijictv3n8spl.pdf

[12] https://www.youtube.com/playlist?list=
PLPk6Wn69oEjJNX-Ug7-UV1PoN15TrRB84