BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 416 (January 2022)

Microprocessor and Embedded Systems Laboratory

Final Project Report

Section: A2 Group: 05

IoT Based Water Quality Monitoring System

Course Instructors:

Dr. Md Zunaid Baten
Bejoy Sikder

Signature of Instructor: ____

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. <u>Type the student ID and name, and put your signature</u>. *You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.*

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: ____________________ **Full Name: Abrar Md. Mahir** **Student ID: 1906039**	**Signature:** ____________________ **Full Name: Jonaidul Islam Sikder** **Student ID: 1906041**
Signature: ____________________ **Full Name: A. S. Al Mahmud Sajid** **Student ID: 1906050**	**Signature:** ____________________ **Full Name: Manzoor Elahi Tamjeed** **Student ID: 1906046**

Table of Contents

1 Abstract

This project is aimed at the ensuring the hygienic safety of water that is used for the purpose of drinking and performing household tasks through real time monitoring of different aspects linked to the indication of water quality including temperature, electrical conductivity and pH of water. Samples can be collected from household sources and the aforementioned parameters have been measured using dedicated sensors and ESP32 WiFi Module Micro controller and observers can monitor the real time analysis via the IoT based ThingSpeak server and the server data is also fed to their mobile phones by developing an app on MIT App Inventor.

2 Introduction

Water is one of the key elements of sustaining any form of life known to man. Access to clean drinking water and sanitation is regarded as a basic human right and SDG goal no. 6. Despite the fact that around ¾-ths of the surface of our planet is covered by water, drinkable water is scarce and water pollution and climate change have been adding more complexity to this already acute crisis. Currently around 771 million people lack access to safe drinking water.[2] Bangladesh, despite being a riverine country with abundant rainfall, is not safe from the grasp of safe water crisis. 98% of her population has basic water coverage, but, only 58.5% of the population have access to safely managed water 86% poorest household's water is E Coli contaminated and 16.7% of the population have to use arsenic contaminated water.[3] Water pollution is only adding fuel to the fire of this drinking water shortage conundrum, wreaking havoc on public health and sustainable development. According to UNESCO 2021 WORLD WATER DEVELOPMENT REPORT, about 829,000 people die each year from diarrhea including approximately 300,000 children aged below five.[4] Worldwide research has been going on in order to address this issue. Water quality can be interpreted from measurements involving different parameters including electrical conductivity, pH level, BOD, COD, TDS etc. These measurements can be uploaded to dedicated IoT servers where observers can be granted access based on the concept of cloud contribution. These measurements can be taken using sensors and micro controllers and then the upon the observations, necessary steps can be taken if water is deemed unsafe for use.

3 Design

3.1 Problem Formulation (PO(b))

The project's main goal is to monitor important characteristics including pH, electrical conductivity, and temperature in real-time in order to address the urgent problem of water safety for drinking and domestic usage. The problem at hand is the scarcity of efficient, real-time monitoring equipment, which causes responses to suspected water contamination to be delayed. Because tainted water may go unnoticed this gap presents significant health hazards, especially those related to waterborne infections. In order to enable communities to take active measures to protect their health and well-being, the project intends to address the demand for complete and accurate information on water quality.

3.1.1 Identification of Scope

In addition to being scarce since water supplies are running out at a frightening rate, water is also contaminated due to population growth and widespread industrialization.

Because there is a dearth of clean drinking water in our nation, waterborne illnesses are very widespread in both urban and rural locations.
Therefore, there is plenty of room to implement a remote water quality monitoring system that uses micro-controllers to alert people when the quality of the drinking water drops below the standard values

3.1.2 Literature Review

IoT-based water quality monitoring systems collect data on various water parameters, such as pH, temperature, Total Dissolved Solids (TDS), and so on, and send it to a cloud platform for analysis and visualization.
Several research have suggested and deployed IoT-based water quality monitoring systems for a variety of applications, including aquaponics, drainage, and natural water bodies, and have shown that they are more accurate, efficient, and scalable.
However, there are significant obstacles and limits to IoT-based water quality monitoring systems that must be solved before they can be widely used and integrated. These include data quality, security, privacy, and interoperability.

3.1.3 Formulation of Problem

Regarding the safety of the drinking water that is provided to our residences, there is widespread public concern.
But the general public still cannot quantify the amount of safety or contamination.
Even once a small system is put into place, it might not be possible for every person to have all the tools they need to measure various water safety factors. Not just people but industries at small large scales as well

3.1.4 Analysis

From the problem formulation, it is evident that implementation of remote monitoring system is imperative to reap the benefits offered by the project.
The parameters can be measured using a micro controller and dedicated sensors and then uploaded to Internet servers for implementing the monitoring system. Considering how commonplace mobile phones with Internet connection have become, creating a mobile application for remote monitoring is also essential.

3.2 Design Method (PO(a))

Required Hardware:

- ESP32 WiFi Module
- Gravity Analog TDS Sensor
- DS18B20 Temperature Sensor
- DFRobot SEN0161 pH Sensor
- 128 × 64 I2C Interface OLED Display
- 4.7 kΩ Pull-Up Resistor
- Connecting Wires
- Breadboard

Required Software and Applications:

- Arduino IDE
- MIT AI2 Companion
- **Required Libraries:**
- One-Wire Library
- Dallas Temperature Library
- ADS1015 Library
- DFRobot ECP EC Library
- Adafruit GFX Library
- Adafruit SSD1306 Library

Required Websites:

- IoT Platform account (ThingSpeak)
- MIT App Inventor Website

1. Identifying Parameters:

Temperature: 20-30°C
TDS: Less than 500 mg/L
pH sensor: 6.5-8.5

2. Hardware Setup:

Connecting the selected sensors to the ESP32 using suitable interfaces in breadboard according to the circuit diagram shown in the picture

3. Set Up IoT Platform:

Creating account on IoT platform like ThingSpeak and obtaining necessary credentials like API keys & creating channel

4. Programming:

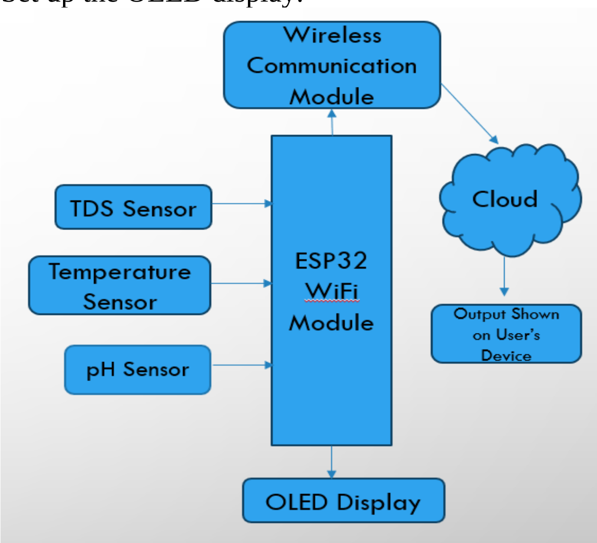
The source code/program is written in C/C++ on Arduino IDE

5. Implement communication:

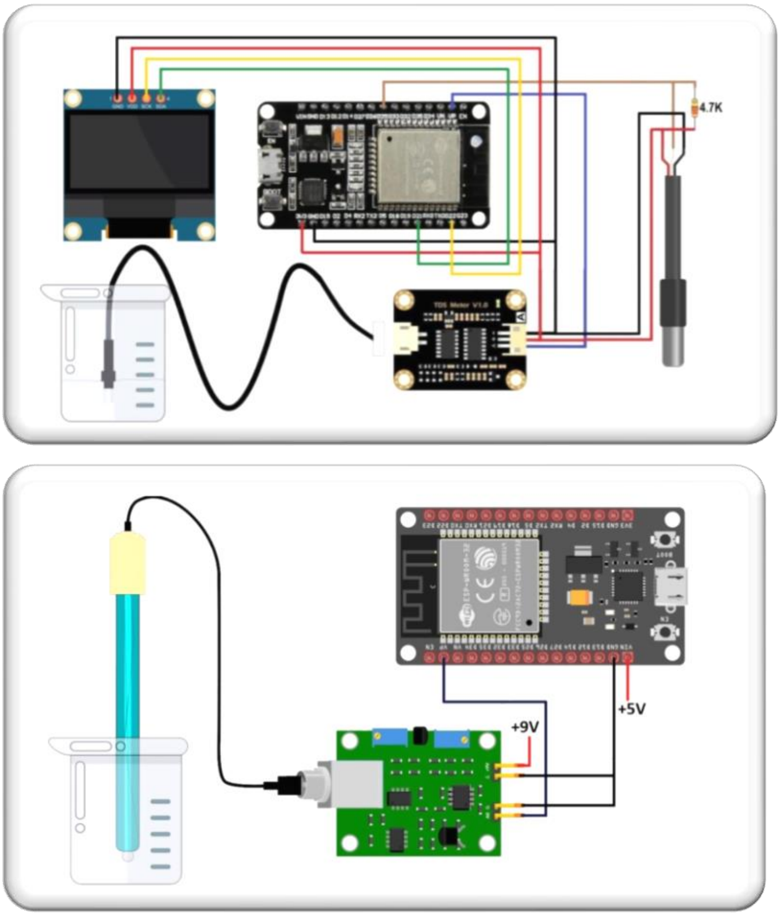
Integrate ESP32 with the IoT platform using relevant libraries.
Configure the ESP32 to send sensor data to the IoT platform at regular intervals.

6. Data visualization:

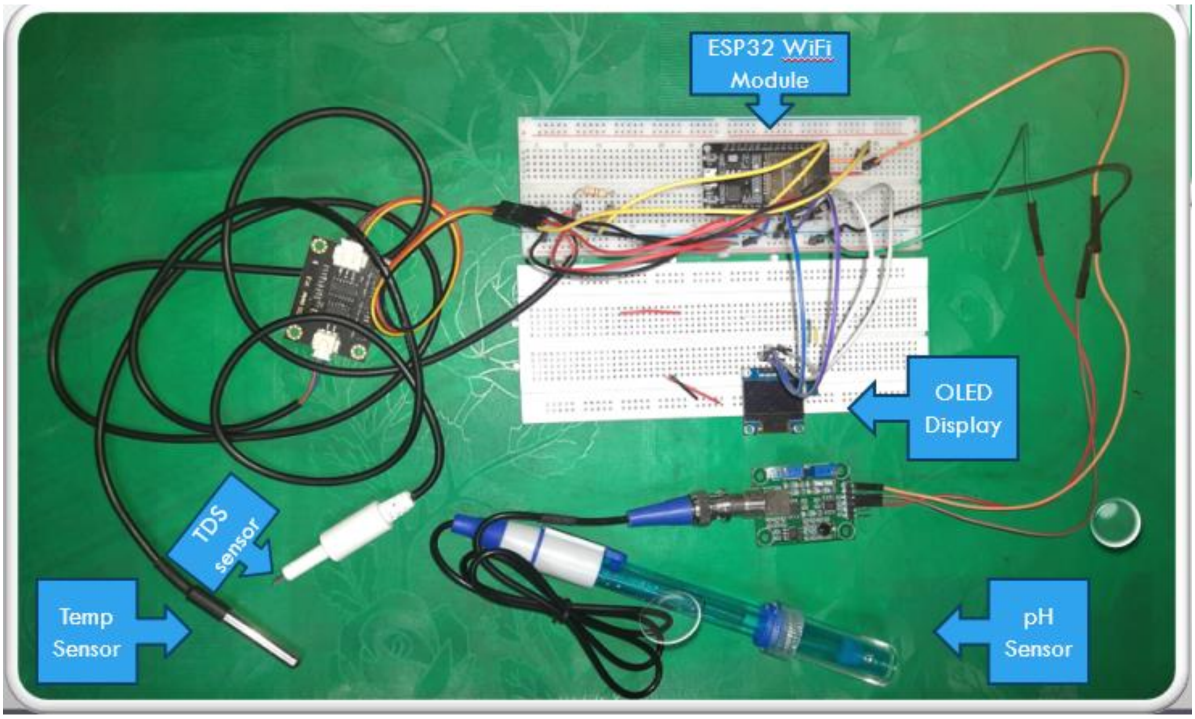
Create dashboards on the IoT platform to visualize real-time and historical data.
Set up the OLED display.



3.3 Circuit Diagram



3.4 CAD/Hardware Design



3.5 Full Source Code of Firmware

If appropriate for your project. Otherwise, remove this section

```
#include <Arduino.h>  //basic function for arduino
#include <Wire.h> //allows communication with I2C devices; I2C has two lines
SCL and SDA, SCL is used for clock and SDA is used for data.
#include <EEPROM.h> //provides functions to read and write to the EEPROM
(Electrically Erasable Programmable Read-Only Memory)
#include <WiFi.h> //enables Wi-Fi connectivity
#include <OneWire.h>
#include <DallasTemperature.h> // typically used to interface with OneWire
temperature sensors, such as the DS18B20
#include <Adafruit_ADS1X15.h>  //supports the ADS1015 and ADS1115 analog-to-
digital converters (ADCs) from Adafruit.
#include <DFRobot_ESP_EC.h> // used for interfacing with electrical
conductivity (EC) sensors
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h> //support OLED displays, such as the SSD1306,
and provide graphics functions and text on the display.

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

#define ONE_WIRE_BUS 4                // this is the gpio pin 4(ADC4)  on
esp32.
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

DFRobot_ESP_EC ec;
Adafruit_ADS1115 ads;

float voltage, ecValue, pH, Value=0, temperature = 25;

String apiKey = "0IUGZ3PZ67DMYS6L";     // API key from ThingSpeak

const char *ssid =  "Dot_theTF";     // wifi ssid and password
const char *pass =  "Saha.69@mf";
const char* server = "api.thingspeak.com";
const int potpin = A3;   // GPIO39 (ADC3),VN pin of the ESP32 Module
const int sampleSize = 20; // Number of samples to average
const float alpha = 0.2; // Exponential moving average smoothing factor

WiFiClient client;

void setup()
{
  Serial.begin(115200);   //Begins serial communication at a baud rate of
115200 for debugging.
  EEPROM.begin(32);//nitializes the EEPROM to store calibration data.
  pinMode(potpin,INPUT);
  ec.begin();
  sensors.begin();
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { //specifies the I2C
address of the display
    Serial.println(F("SSD1306 allocation failed"));
    for (;;);   //infinite loop
  }
  delay(2000);
  display.clearDisplay();

  Serial.println("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
```

```
  }
  Serial.println("");
  Serial.println("WiFi connected");
}



void loop()
{
  float pH = 0; //this is the analog voltage of pH sensor
  float pH_value; //this is the pH Value
  voltage = analogRead(A0); // VP in esp32 is the gpio 36
  sensors.requestTemperatures();
  temperature = sensors.getTempCByIndex(0);  // read temperature sensor to
execute temperature compensation
  ecValue = ec.readEC(voltage, temperature); // convert voltage to EC with
temperature compensation

 // Collect samples
  for (int i = 0; i < sampleSize; i++) {
    pH += analogRead(potpin) * (3.3 / 4095.0); // Convert analog reading to
voltage (assuming 3.3V reference)
    delay(10);
  }

  // Average the samples
  pH /= sampleSize;

  // Apply exponential moving average (EMA)
  static float prevpH = pH; // Initialize previous pH value
  pH = alpha * pH + (1 - alpha) * prevpH; // Apply EMA
  prevpH = pH; // Update previous pH

  pH_value= -12.7 * pH + 36;  // this is for manual calibration

  Serial.print("Voltage:");
  Serial.print(pH, 2);

  Serial.print("Temperature:");
  Serial.print(temperature, 2);
  Serial.println("ºC");

  Serial.print("EC:");
  Serial.println(ecValue, 2);

  Serial.print("pH:");
  Serial.println(pH_value, 2);

  display.setTextSize(2);
  display.setTextColor(WHITE);

  display.setCursor(0, 0);
  display.print("T:");
  display.print(temperature, 2);
  display.drawCircle(85, 0, 2, WHITE); // put degree symbol ( ° )
  display.setCursor(90, 0);
  display.print("C");

  display.setCursor(0, 20);
  display.print("EC:");
  display.print(ecValue, 2);

  display.setCursor(0, 40);
  display.print("pH:");
  display.print(pH_value, 2);
  display.display();
  delay(500);
  display.clearDisplay();


  ec.calibration(voltage, temperature); // calibration process

  if (client.connect(server, 80))  //   api.thingspeak.com
  {
```

```
    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(temperature, 2);
    postStr += "&field2=";
    postStr += String(ecValue, 2);
    postStr += "&field3=";
    postStr += String(pH, 2);
    postStr += "\r\n\r\n\r\n";
    delay(500);

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n\n");
    client.print(postStr);
    delay(500);
  }
  client.stop();
}
```

Table: Source Code for the main program

4 Implementation

4.1 Description

Our project aims to create an IoT-based water quality monitoring system for pH, electrical conductivity and temperature levels. The system utilizes an ESP32 microcontroller for data acquisition and communication with various sensors, an OLED display for real-time data visualization, Wi-Fi module for sending information to the ThingSpeak server and suitable libraries for sensor interfacing and communication.

Components Used:

1. **ESP32 Microcontroller:** The ESP32 serves as the central processing unit for the monitoring system. It handles sensor data acquisition, processing, and communication tasks.

2. **Analog pH Sensor:** We employ a DFRobot_ESP_PH pH sensor for measuring the pH level of the solution. This sensor is specifically designed for compatibility with ESP32 microcontrollers and offers nearly accurate pH measurements.

   Its data wire is connected to VN pin, which is ADC1_3 pin of ESP32 module .

3. **Analog TDS Sensor:** We employ a DFRobot_ESP_EC pH sensor for measuring the TDS (Total Dissolved Solid) level of the solution. Then the imported library convert it to EC using the following equation.

   $$\mathbf{EC}=\frac{\boldsymbol{raw\ EC}}{\mathbf{1+0.0185*(\boldsymbol{T}-25)}}$$

- EC represents Electrical Conductivity.

- rawECrawEC represents the raw electrical conductivity measurement obtained from a sensor, typically in units of microsiemens per centimeter ($\mu S/cm$).

- T represents the temperature at which the EC measurement is taken, typically in degrees Celsius (°C).

- The constant 0.01850 is a coefficient used for temperature compensation, and it has no units.

- The value 25 is the reference temperature (typically 25°C) for temperature compensation.

  The units of the electrical conductivity (EC) calculated using this equation are microsiemens per centimeter ($\mu S/cm$).

  Its data wire is connected to VP pin, which is ADC1_0 pin of ESP32 module .

4. **DS18B20 Temperature Sensor:** Our temperature sensor is based on the Dallas temperature sensor technology, utilizing one-wire communication protocol. It is connected to pin D4 of the ESP32, configured as an ADC pin, enabling analog communication for temperature readings.

   Its data wire is connected to D4 pin of the ESP32 module which is basically ADC2_0 pin working in normal mode.

5. **OLED Display:** An OLED display is integrated into the system for real-time visualization of pH and temperature data. This display provides a user-friendly interface for monitoring and allows for quick assessment of environmental conditions.

   It uses I2C connection(SCL for clock & SDA for data) for connecting with the ESP32 module.

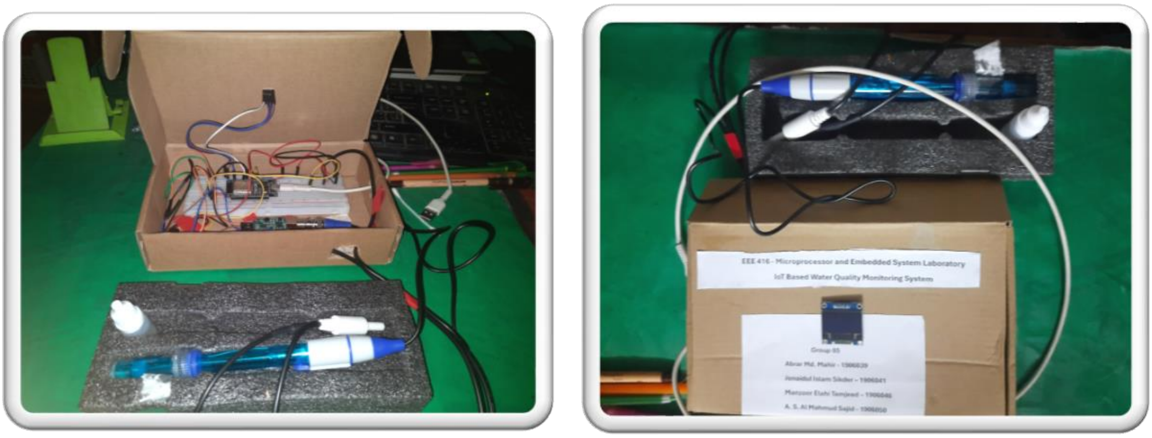


Figure 2: Implementation of Design

Video Demonstration Link:
https://www.youtube.com/watch?v=pY5yzwIFPcM&ab_channel=JonaidulIslamSikder

5 Design Analysis and Evaluation

5.1 Novelty

- Application of a micro controller (ESP32 WiFi Module) in the field of water safety.
- Combination of 3 different kinds of sensors.
- Real Time tracking facility on an Internet server.
- Development of a user-friendly mobile application

5.2 Design Considerations (PO(c))

5.2.1 Considerations to public health and safety

- Considering the effects of water pollution hazard and the need of ensuring water safety.
- Designing a system for alerting people about the 3 key features of safe drinking water.
- Addition of the remote monitoring feature to ensure a 24 × 7 × 365 accessibility to water safety data and potential hazard alert.

5.2.2 Considerations to environment

- Designing a low power consuming system so that the environmental impact is minimized.
- Eliminating the need to use potentially hazardous chemical reagents to measure water safety level.

5.2.3 Considerations to cultural and societal needs

- Water is a key element to healthy living and thus ensuring a balanced societal and cultural life.
- Thus, this project addresses this necessity and helps improve the present socio-cultural scenario involving safe drinking water.

5.3 Investigations (PO(d))

5.3.1 Design of Experiment

Here, in this project, we implemented our testing procedure by analyzing four liquids. These are, normal water, salty water, detergent and lemon juice.

5.3.2 Data Collection

5.3.3 Results and Analysis

We tested our samples in 26 degree and 70 degree Celsius and the temperature sensor worked correctly.

Here, in normal water,
Electric conductivity= 5.2
pH= 6.7
In salty water,
Electric conductivity= 16.5
pH= 7.02
In detergent,
Electric conductivity= 15.6
pH=11.35
In lemon juice,
electric conductivity= 15.11
pH=4.75

5.3.4 Interpretation and Conclusions on Data

Normal water and salty water are neutral in nature. So, the pH value is almost 7 in both cases. But for electric conductivity and TDS, the salty water's conductivity is very high because it has free ions that can carry electrons.

Both for acidic and basic solution, the water conductivity is very high for free ions. But for acid the pH is under 7 and for base, the pH is 11.2 or over 7.

5.4 Limitations of Tools (PO(e))

The sensor's sensitivity may have an impact on the value's outcome. The TDS of the sample water is not provided by the TDS sensor. It provides the electric conductivity value. The precise number of hazardous elements, such as arsenic, iron, copper, etc., combined with the water is unknown to us.

5.5Impact Assessment (PO(f))

5.5.1 Assessment of Societal and Cultural Issues

Waterborne diseases can be harmful for public health sector and hence creating societal instability. The project needs to be able to minimize such risks by training and alerting people to measure water quality parameters using simple instruments and observing them remotely.

The use of Internet surveillance will also educate people to use the Internet service more productively.

5.5.2 Assessment of Health and Safety Issues

Safe drinking water is scare in Bangladesh despite it being a riverine country, as mentioned earlier. Electrical conductivity and pH level of water are two features related to this issue that everyone can easily be trained to identify. Hence these features are necessary to be measured for ensuring water safety alongside water temperature.

5.5.3 Assessment of Legal Issues

This project enables the user to remotely monitor water safety features in real time. Hence the authorities concerned with supplying water can be legally accountable to ensure the safety of water they are supplying. Any hazardous situation involving drinking water can be legally contested using this real time monitoring data if necessary.

5.6 Sustainability Evaluation (PO(g))

The sustainability evaluation of this water quality monitoring project reveals its positive impact on various dimensions. Environmentally, the project contributes to the sustainable use of water resources by enabling early contamination detection, which helps in preserving ecosystems and maintaining water quality. Socially, the system promotes public health by ensuring access to safe drinking water and fostering community engagement through awareness. Economically, the project supports cost reductions in water management by addressing issues promptly, preventing potential health crises that could incur higher expenses. Governance-wise, it aligns with policies related to water quality and facilitates efficient water management. Moreover, the technological aspect showcases innovation in utilizing IoT for real-time monitoring, demonstrating adaptability to advancements. In essence, the project proves to be a sustainable initiative, fostering resilience, equity, efficiency, and innovation across environmental, social, economic, governance, and technological domains.

5.7 Ethical Issues (PO(h))

One potential ethical issue associated with this water quality monitoring project revolves around data privacy and security. As the system collects and transmits real-time data about water quality, including potentially sensitive information about households, there is a concern about how this data is handled, stored, and protected. Ensuring the privacy of individuals and safeguarding their data from unauthorized access or misuse becomes paramount. This includes addressing questions about who has access to the collected information, how long it is retained, and whether it is shared with third parties. Other than that it has no ethical issues.

6 Reflection on Individual and Team work (PO(i))

6.1Individual Contribution of Each Member

1906039: Hardware & ThingSpeak Setup

1906041: Hardware, Debugging & App Development

1906050: Coding, Debugging Calibration

1906046: Coding & Debugging

Everyone did a little bit of everything. it was a team effort.

6.2Mode of TeamWork

Initially, we had sat down to discuss the feasibility of the project. Then we had divided the hardware and software parts among ourselves. Every member reviewed articles and papers on this issue in order to get a clearer understanding the features of the project. There was active participation of everyone in every segment of the project.

6.3Diversity Statement of Team

The team was diverse in their way of thinking and attacking challenges

6.4 Log Book of Project Implementation

Date	Milestone achieved
Week-7	Setting up temperature sensor
Week-8	Setting up TDS sensor
Week-9,10	Setting up PH sensor
Week-7,12	Arduino Coding
Week-13	App Making

7 Communication to External Stakeholders (PO(j))

Multidisciplinary aspects of the projects are given below:

- Environmental Science: Environmental factors that can affect water quality, such as pollution sources, pH value temperature etc.
- Electrical and Electronics and IoT: integration of sensors into circuit designing for measuring temperature, electrical conductivity, and pH and making a network of it
- Data Science and Analytics: Data visualization to present meaningful insights to users.
- Chemistry: Measurement of pH, EDS and temperature all require knowledge of fundamental chemistry

Also Computer Coding Knowledge, data science, Embedded System knowledge are essentials

7.1 Executive Summary

Using sensors and an ESP32 WiFi Module, the Internet of Things-based water quality monitoring project measures variables like pH, electrical conductivity, and temperature in real-time. It guarantees quick contamination identification and easy-to-use data access on mobile devices thanks to integration with the ThingSpeak server and an app created by MIT App Inventor. This program encourages community involvement, public health, and sustainable water management. For ethical implementation, data security and privacy must be carefully considered in addition to the environmental, social, and economic factors. All things considered, the project is a creative and all-encompassing solution that promotes the long-term health of communities and ecosystems.

7.2User Manual

1. Gravity Analog TDS Sensor Specifications:
Input Voltage: 3.3 – 5.5 Volts
Output Voltage: 0 – 2.3 Volts
Working Current: 3 – 6 mA
TDS Measurement Range: 0 – 1,000 ppm

TDS Measurement Accuracy: ± 10% at 25°C

2. DS18B20 Temperature Sensor Specifications:
Temperature Measurement Range: -55°C to 125°C (-67 to 257°F)
Temperature Measurement Accuracy: ± 0.5 °C
Required Libraries: Dallas Temperature Sensor Library and One-Wire Library
A 4.7 kΩ resistor needed to be used as a pullup resistor between Data and VCC line

3. DFRobot SEN0161 pH Sensor Specifications:
pH Detection Range : 0 – 14
pH Detection Accuracy: ±0.1
Temperature Range of Operation: 0-60°C
Input Voltage: 5V
Working Current: 5-10 mA
Power Consumption: ≤0.5 W
Response and Stability Time: ≤60 seconds

8 Project Management and Cost Analysis (PO(k))

8.1 Bill of Materials

Item	Price (BDT)
ESP32 30 Pin NODEMCU Development Board with WiFi Module	575
Gravity Analog TDS Sensor	2,069
DS81B20 Temperature Sensor	300
0.96’’ I2C OLED Display	495
Breadboard (2 pieces)	140 × 2 = 280
Jumper Wires (1 set)	180
Arduino Compatible pH Sensor Analog Kit	2,554
Total	**6,453**

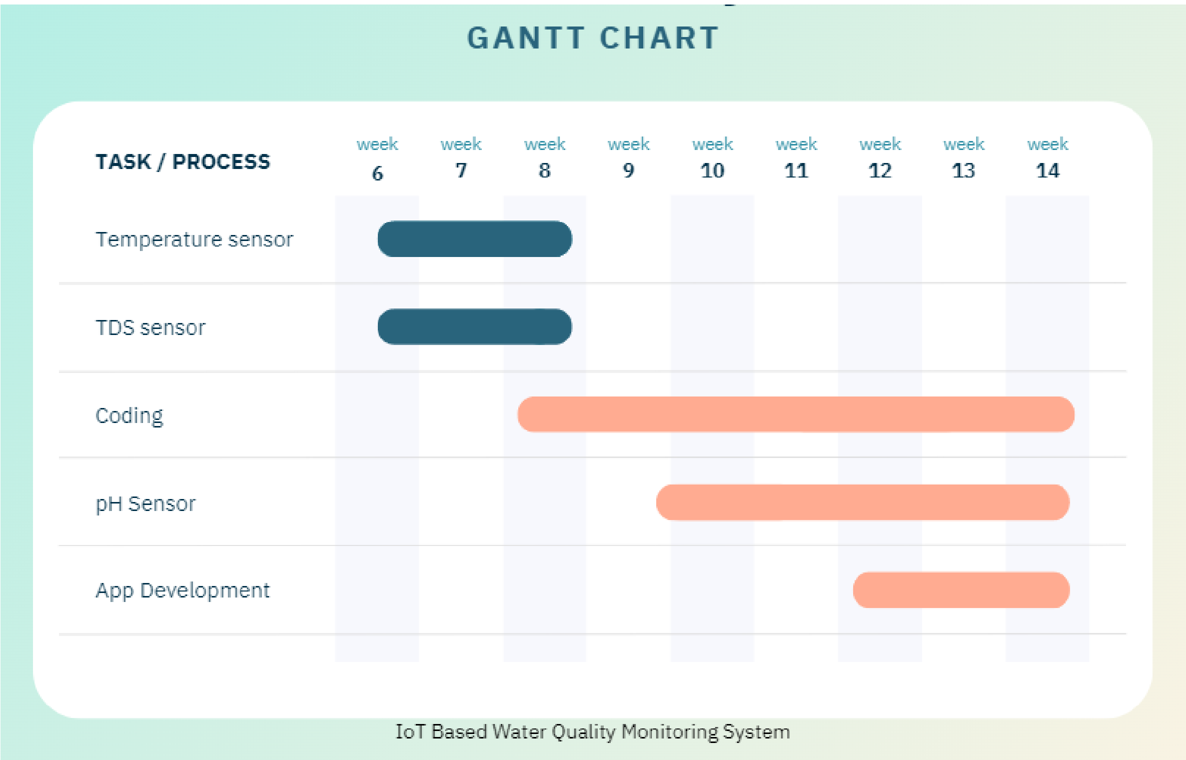
8.2 Calculation of Per Unit Cost of Prototype

6453 BDT

8.3 Calculation of Per Unit Cost of Mass-Produced Unit

No Mass produced unit was found

8.4 Timeline of Project Implementation



9 Future Work (PO(l))

- Inclusion of more precise and cutting-edge sensors.
- Improvement of sensor calibration.
- Development of a more advanced and interactive GUI.
- Scope of working in cohesion with chemical analysts to determine the amount each trace element present in water samples.

10 References

1. https://www.un.org/sustainabledevelopment/water-and-sanitation/#:~:text=Goal%206%3A%20Ensure%20access%20to%20water%20and%20sanitation%20for%20all&text=Access%20to%20safe%20water%2C%20sanitation,in%202030%20unless%20progress%20quadruples.

2. https://water.org/our-impact/water-crisis/

3. https://www.wateraid.org/bd/the-crisis/water

4. https://www.frontiersin.org/articles/10.3389/fenvs.2022.880246/full

5. https://cdn.who.int/media/docs/default-source/wash-documents/wash-chemicals/ph.pdf?sfvrsn=16b10656_4

6. https://randomnerdtutorials.com/esp32-ds18b20-temperature-arduino-ide/?fbclid=IwAR3CnOWbnNx0ryhF59hqaLxR02nO2yf4QCcq0eeFGHBUuAqGkxoJ1vl0o30

7. **Gantt chart:** Canva.com