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

EEE 416 (July 2023)

Microprocessor and Embedded System Laboratory

Final Project Report

Section: A1 Group: 02

IoT Based Drinking Water Quality Monitoring System

Course Instructors:

Dr. Sajid Muhaimin Chowdhury, Assistant Professor
Ayan Biswas Pantha, Adjunct Lecture

Signature of Instructor: ______________________________

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature. ***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: Muzakkir Ahmed** **Student ID: 1906002**	**Signature:** ____________ **Full Name: Md. Rowatul Jannat** **Student ID:1906016**
Signature: ____________ **Full Name: Md. Tareq Mahmud** **Student ID:1906024**	**Signature:** ____________ **Full Name: Mithon Rahman** **Student ID:1906028**

Table of Contents

1 Abstract

The main focus of this project is to development and implementation of an IoT-based system for monitoring drinking water quality. As concerns rises over water contamination and the need for real-time monitoring solutions is becoming more inevitable day by day, the integration of IoT technology offers a great method to measure the quality of water and it's sustainability. The samples which will be used for monitoring can be collected from household sources and the parameters like pH, TDS and temperature can be measured using various sensors and micro controllers and observers can monitor the real time analysis via IoT based servers. Ensuring pure and safe drinking water is a basic human right and clean water and sanitation is one of the SDG goals devised for ensuring sustainable development. This project will help us to achieve SDG goals and minimize the risk of water-borne diseases.

2 Introduction

Environment sustainability and public health heavily depends on safe and drinkable water. Water contamination may result in a number of health risks. It can cause infections and hazardous chemicals exposure to long-term health consequences. Conventional approaches to monitoring water quality often depend on recurring sampling and laboratory analysis, which could not provide up-to-date information on possible contamination events or changes in water quality. Also it is a very time consuming process.

The proposal suggests creating and implementing an IoT-based drinking water quality monitoring system in order to get around these restrictions. This strategy gathers data in real time from sensor nodes positioned throughout the water distribution network by using the potential of IoT technology. Through the employment of sophisticated analytics algorithms to examine this data, the user may observe the quality of the water and can take decision from it to whether he/she will drink it or not.

3 Design

3.1 Problem Formulation (PO(b))

3.1.1 **Identification of Scope:** Water is very essential for our life. 70% of our body consists of water. We need to drink a lot of water so that our body can function properly. But nowadays the quality of drinking water is degrading. Almost in every rural area, water is being polluted every day. The contaminants in these water mix with fresh water and degrades the quality of it. This water contains heavy metals, acidic solution or basic solution. Due to the bad quality of water, people are suffering in various diseases. In some places, underground water also contains these. So, monitoring the drinking water quality is now very much needed. We identified these problem and decided to work to find a solution of this through our project.

3.1.2 **Formulation of Problem:** The degradation of drinking water quality due to pollution poses a significant threat to public health and environmental sustainability. Traditional methods of water quality monitoring are often limited in their ability to provide real-time surveillance and early detection of contaminants, leading to delayed response times and increased health risks for communities. Simultaneously monitoring several sensor parameters poses challenges. Besides, calibrating the sensor complexes the problem further.

3.1.3 Analysis

After sorting out the problem, we tried to solve it. The common method for checking water quality is a very long process. It will take time to find out whether it is drinkable or not. So, we tried to solve this problem using obtained knowledge. Some parameters are set by WHO which give us the amount of solids, pH and temperature that can be accepted by our body. Taking these designed a prototype that will help us to solve the problem

3.2 Design Method (PO(a))

Hardware components:

- ESP32 WiFi Module
- Analog TDS Sensor
- DS18B20 Temperature Sensor
- OLED Display
- pH Sensor
- Resistors
- Connecting wires

Software components:

- Arduino IDE (for ESP32 firmware development)
- IoT Platform account (ThingSpeak)
- Platform-specific libraries for integrating ESP32 with the IoT platform
- (One-Wire Library, Dallas Temperature Library, ADS1015 Library, DFRobot ECP EC Library, Adafruit GFX Library, Adafruit SSD1306 Library)

Application:

- Application is developed using MIT App Invertor website
- Application retrieves data from the server

1. ESP32 WROOM Development module:

- Processor: Dual-core 32-bit operating at a frequency of 80Mhz
- RAM: 520kB
- Power Supply: 2.2 – 3.6V

- ADC 12-bit: 18
- DAC 8-bit: 2
- WiFi available

2. pH Sensor Analog Meter Kit:

- Input Voltage: 5V
- Accuracy: +-0.1pH
- Response Time: <=1min
- Operating Range: 0-14 pH

3. Gravity Analog TDS Sensor:

- 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

4. DS18B20 Temperature Sensor:

- 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
- 4.7 kΩ resistor needed to be used as a pullup resistor between Data and VCC line

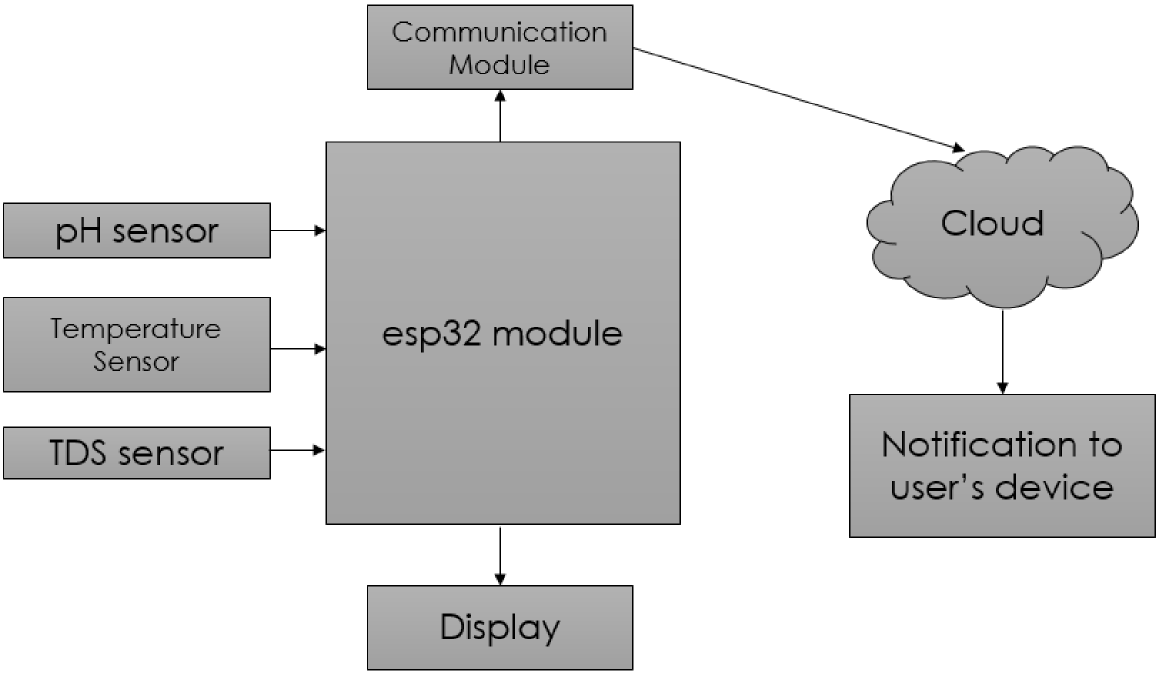


Fig. Flow chart

3.3 Circuit Diagram

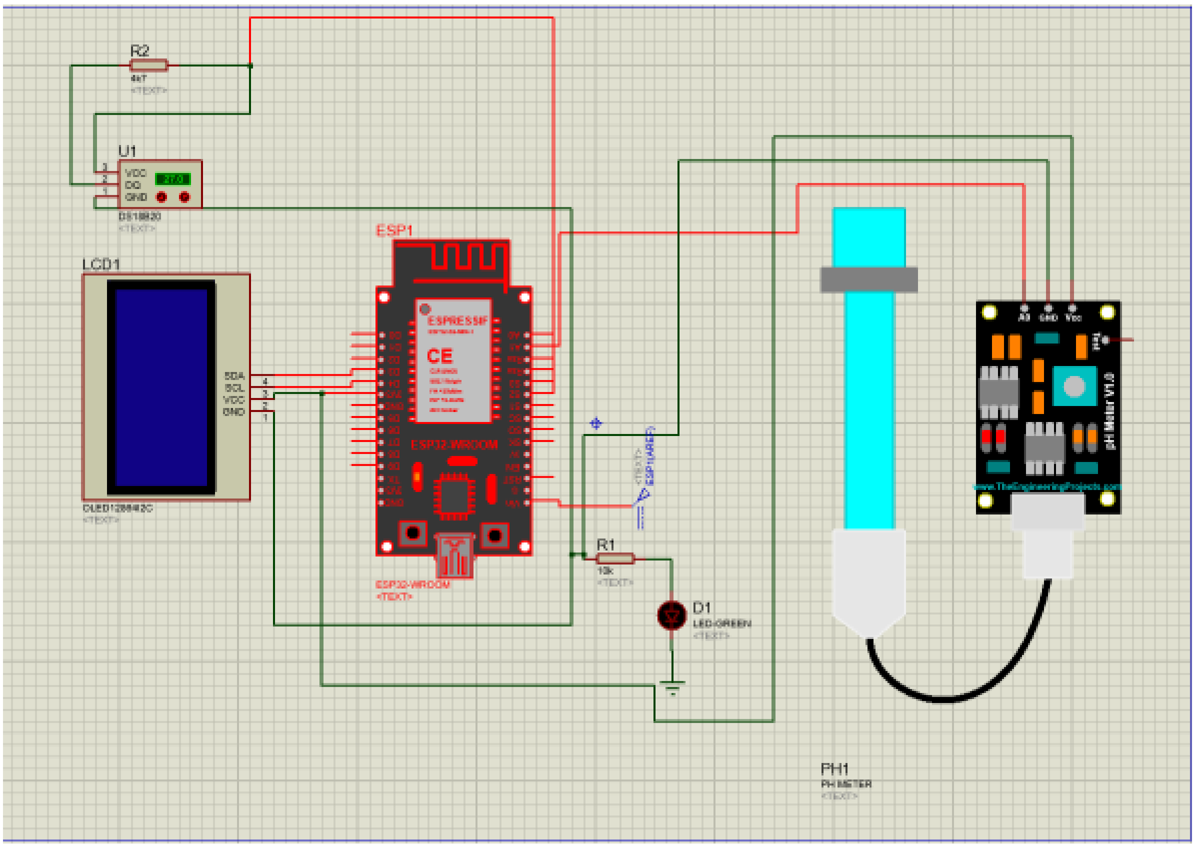


Fig. Circuit Diagram

3.4 CAD/Hardware Design

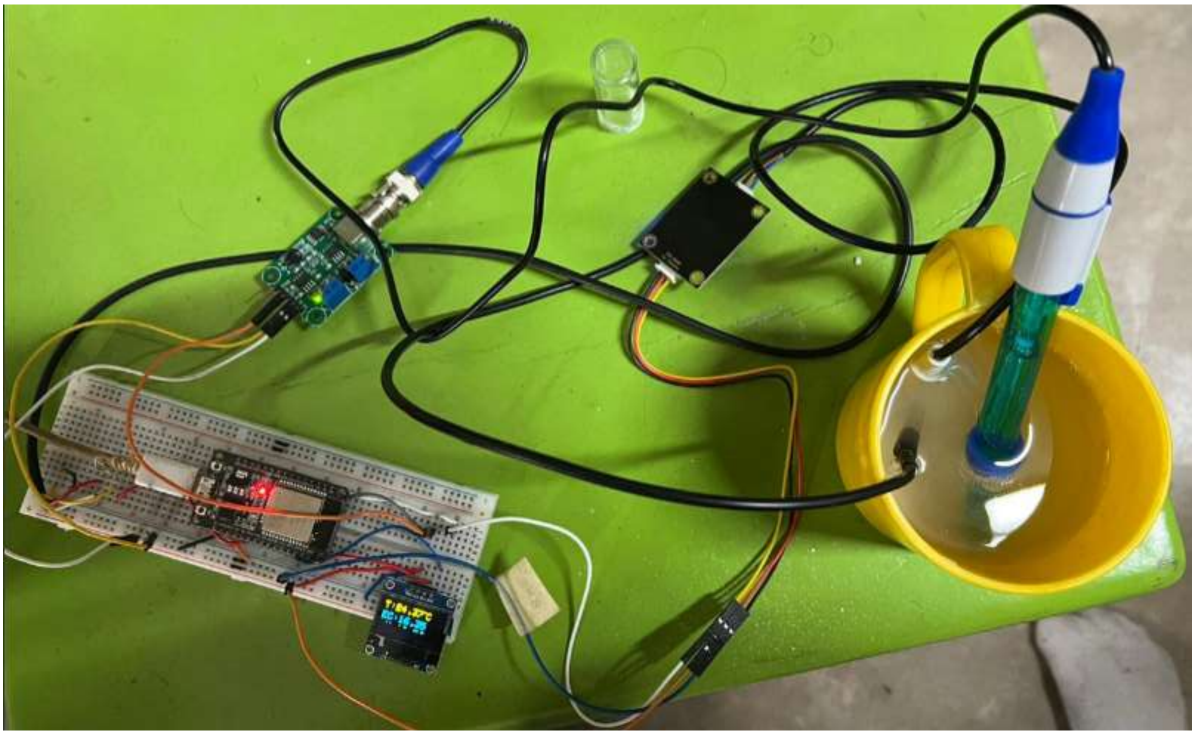


Fig. Hardware Setup

3.5 Full Source Code of Project

```
#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 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;
int buf[10],temp;

String apiKey = "QLH883C5C51W5UIJ";     // API key from ThingSpeak

const char *ssid =  "Redmi Note 8";     // wifi ssid and password
const char *pass =  "abcd5425";
const char* server = "api.thingspeak.com";
const int potpin = A3;

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)
```

```
void loop()
{
  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

 for(int i=0;i<10;i++)       //Get 10 sample value from the sensor for smooth the value
  {
    buf[i]=analogRead(potpin);
    delay(10);
  }
  for(int i=0;i<9;i++)        //sort the analog from small to large
  {
    for(int j=i+1;j<10;j++)
    {
      if(buf[i]>buf[j])
      {
        temp=buf[i];
        buf[i]=buf[j];
        buf[j]=temp;
      }
    }
  }

  Value=0;
  for(int i=2;i<8;i++)                      //take the average value of 6 center sample
    Value+=buf[i];
  float phValue=(float)Value*3.3/4095/6; //convert the analog into millivolt
  phValue=3.3*phValue;



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

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

  Serial.print("pH:");
  Serial.println(phValue, 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(phValue, 2);
  display.display();
  delay(500);
  display.clearDisplay();


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

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

```
{
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
}
    postStr += "&field3=";
    postStr += String(phValue, 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();
}
```

```
{

    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(temperature, 2);
    postStr += "&field2=";
    postStr += String(ecValue, 2);
}
```

Table: Source Code for the main program

4 Implementation

4.1 Description

ESP32 Microcontroller: ESP32 which will process all data and serve as communication center is principally the device. Moreover, it serves the functions of data acquisition from the senors as well as data transmitting to the server.
pH Sensor: The PH sensor establishes the measure of acidity or alkalinity in the water. It is basic for the evaluation of water microbiology parameters and revealing the sources of a possible infection.
Temperature Sensor: The temperature senor measures the water temperature which serves as a red flag for quality parameters of water and aquatic life in the environment.
TDS Sensor: The TDS topic refers to the total dissolved solids sensor that gives an indication of the amount of dissolved solids in water. It serves as the instrument for tracing the basic quality and naturalness of the water.
The ESP32 microcontroller is the core part of the system circuit. It connects all sensor modules and the server. They all communicate to each other through it. pH sensor, temperature sensor, and TDS sensor are connected with ESP32 using analog or digital pins. ESP32 interface with these sensors over such interfaces. ESP32 grasp measurements from the sensors once a certain time gap.
The next step will be to collect the data from the sensors and then the ESP32 would utilize secure protocols such as MQTT and HTTP for connection establishment of the server. The transmitted data is either during a live or regular period, according to the system settings.
The sensor serves as the central vessel for storing, modeling, and representing the acquired information. It gets the data packets coming from the monitoring stations set up in various spots across the territory. This server is capable of specific data processing operations which consist of aggregation, anomaly detection and offering related analysis.

5 Design Analysis and Evaluation

5.1 Novelty

Real Time Surveillance: The project is based on a new approach of water quality monitoring by using the IOT devices mechanism which gives in time imperialistic surveillance of water parameters. Contrary to the established means that conduct regular sampling and laboratory investigation, the system provides for unending as a result of which contaminants and anomalies are almost immediately discovered.

Multi Parameter Monitoring: Many characteristics of this project are the following but, the most up-to-date feature is the ability of monitoring more than one parameters at the same time. The system is made up of IoT sensors, which can monitor different factors, like pH level, turbidity, dissolved oxygen and microbes. Thus, it is providing a complete and detailed data about water quality along water distribution lines.

Integration with existing Infrastructure: The project places emphasis on the necessity of the integration the IoT-based monitoring system with standing water management infrastructure. The system is structured such that it has the ability to closely tie with SCADA's, data logging equipment and regulatory compliance tools, which enhances its usefulness in water quality management practices as it supports interoperability and efficiency.

Smart Monitoring with mobile app: Mobile app which exhibits the water monitored datawas made by us and we also developed the app that will show whether the water is drinkable or not..

5.2 Design Considerations (PO(c))

5.2.1 **Considerations to public health and safety:**

The public health, safety and security that the IoT-based water quality monitoring system should provide include early warning for water contaminants, continuous surveillance, compliance with regulations and transparency in water quality data, community involvement, planning for responses in emergency cases, privacy protection, and equal access. These principles guarantee rapid reaction to health hazards, adherence to rules, community participation, privacy protection and equitable water supply for the safety of the populace.

5.2.2 **Considerations to environment:**

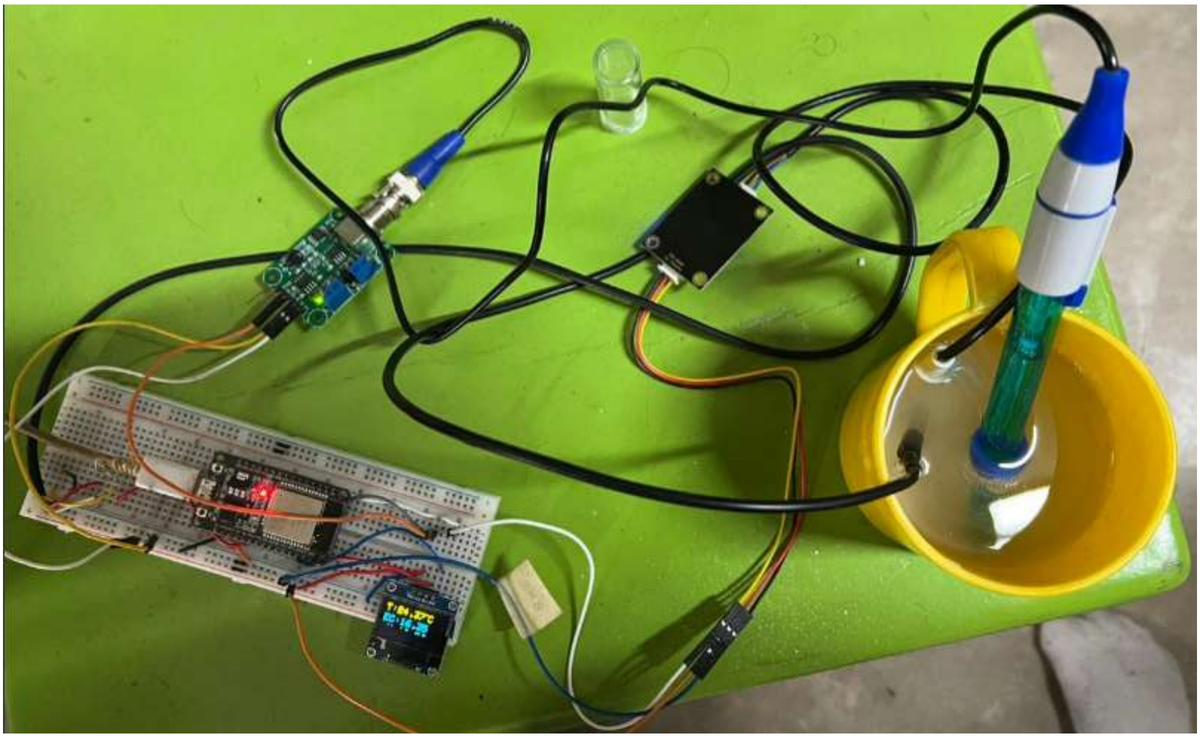
In IoT-based water quality monitoring systems, environmental considerations are related to a delicate balance between economic impact mitigation, support of sustainable solutions, and preservation of natural resources. This spreads environmental awareness, applies eco-friendly materials, limits energy consumption, slacks in reducing waste generation, and reduces dubious harm to aquatic ecosystems. Besides that, the system has to be so designed that it caters to the needs of endorsing biodiversity conservation, habitat safeguarding, and normal ecosystems as well as making their longevity sustainable. By reaching a sufficiently big-scale project with this tool, we will also be able to detect the pollution sources in the environment.

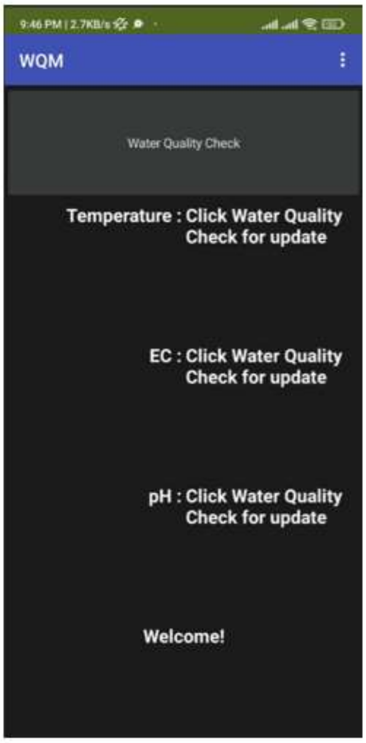
5.2.3 **Considerations to cultural and societal needs:**

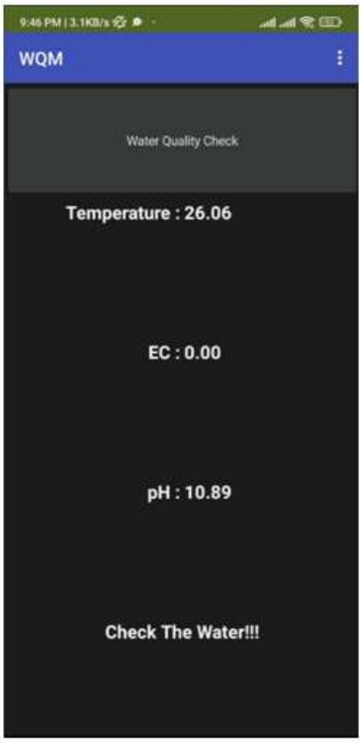
Likewise, for the IoT-based water quality monitoring system, issues will be ranged from public health matters to environmental management till societies/cultures. This types of monitoring calls for early detection of toxicants for health preservation, eco-friendly strategies for environmental conservation, and cultural coordination for diverse community interaction. An interaction of these factors will be realized through the system and achieve efficient surveillance, protection of natural resources, build relationship at the community level and cooperation.

5.3 Investigations (PO(d))

5.3.1 **Results and Analysis**







We have performed the experiment on our test samples and the result are shown here. We can see that in the first picture, the OLED display shows the temperature, eletrical conductivity and pH value of our test sample. We have also seen these results in our application which is accessable from any mobile device

5.3.2 Interpretation and Conclusions on Data

From testing on the samples, we can see the EC, pH and temperature of our sample. Sensors were working and giving the value. Sensitivity of the pH sensor was a bit high but other sensors sensitivity was in acceptable limit. App was updating data.

5.4 Limitations of Tools (PO(e)):

1. Sensitivity of the sensor can affect the result of the value

2. TDS sensor does not give the TDS of the sample water. It gives the value of electric conductivity. We do not get the exact value of how much dangerous element like arsenic, iron, copper etc. mixed with the water.

5.5 Impact Assessment (PO(f))

5.5.1 Assessment of Societal and Cultural Issues:

An IoT-based water quality monitoring system improves the wellbeing of the society by revealing the instantaneous information about the water safety, foiling the risks of waterbourn diseases and guaranteeing the access to the clear water even in underprivileged neighborhoods. Socially, it creates environmental consciousness and care, sharing about the importance of, and overseeing of, community's natural resources.

5.5.2 **Assessment of Health and Safety Issues:**

The project is an addition to the daily health outcomes of the individuals by monitoring the water parameters of quality which in turn decreases the intake of dangerous pollutants. The project further regulates the risk associated with consumption of the water once exposed to their way ends up being contaminated. The safety system enhanced by the detection of toxic elements in timely hours allowing the immediate movement which completely put off the possibility of health crises and positively impacted the entire community which depends on clean water sources for their general well-being.

5.5.3 **Assessment of Legal Issues:**

That is to say the project makes legal system more powerful by providing constant data about water quality in real time, hereinafter controlling the current water quality legislations and enacting new standards on the basis of in depth monitoring. This contributes to the transparency and accountability of the legal process about water pollution cases resulting in the community to respect the rule of law.

5.6 Sustainability Evaluation (PO(g)):

The project is sustainable to some extent as it uses some basic sensors and an esp32 board. These components are very likely not to break down. On an average, after 6 months there might be required to do some inspection. The sensor probes will be indulged in the water for most of the time. We can notice the sensors working on our server and also from app from mobile. Overall, it is sustainable

5.7 Ethical Issues (PO(h))

Ideas of the project were improvised by us. We took help from the internet but we improvised the ideas and merged different ideas for our project. So, no ethical issues from our side.

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

6.1 Individual Contribution of Each Member:

ID	Contribution
1906002	**Worked in the hardware implementation, circuit debugging, app development**
1906016	**Worked on the software part, coding, code debugging, app development**
1906024	**Worked in the hardware implementation, circuit debugging, testing**
1906028	**Worked on the software part, coding, debugging of the code, testing**

6.2 Mode of Teamwork:

At the beginning of the project, we divided into two teams. One team looked at the hardware part, the other looked at the software part. Each team had two members. ID 1906002,1906024 worked on the hardware part and ID 1906016,1906028 worked on the software part. Combining both, we successfully completed our project.

7 Communication to External Stakeholders (PO(j))

7.1 Executive Summary

FOR IMMEDIATE RELEASE

Title: At the focal point of the cutting-edge IoT (Internet of Things) Water Quality Monitoring System is the environmental conservation revolution.

An innovative IoT-based Water Quality Monitoring System by our team. Besides applying ESP32 microcontrollers together with the sensors such as pH, temperature or TDS for data collection and analysis in a real-time mode. Remotely connected to the central server, it lets the user receive and send signals via Bluetooth. This up-to-date solution gives water users including individuals and companies the ability to save water efficiently. With the help of its simple interface and the offering of accurate observations, it completely reinvent environmental protection activities worldwide. Paired with us, technology will contribute to a prosperous environment.

For media inquiries, contact:
Mr. XYZ
General Manager
xyz1920@gmail.com

7.2 User Manual

We just to need to connect the esp to power. The esp32 will connect to the wifi and start updating the sensors data to server. We can check the data from internet or from the mobile app

7.3 Github Link

https://github.com/rowatulrafi/IoT-based-Water-Quality-Monitoring

7.4 YouTube Link

https://youtu.be/xvFZjo5PgG0?si=fdqLYjCa74aGVvLS

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

8.1 Bill of Materials

Item	Cost (tk)
ESP32 Module	420
TDS sensor	1700
pH sensor	2504
Temperature sensor	700
OLED Display	447
Breadboard	150
Wire and Resistor	50
Total	5971

8.2 Timeline of Project Implementation

Week 4	Project proposal submitted
Week 5	Project proposal approved
Week 6	Buying component
Week 7	Project demo presentation
Week 8	Hardware setup
Week 9-10	Software part
Week 11-12	Sensor calibration
Week 13	Final project presentation
Week 14	Final project hardware demonstration

9 Future Work (PO(l))

- Enhancing sensor technology

- Smart water management system combining real time monitoring and water treatment plant)

10 References

- https://how2electronics.com/iot-based-drinking-water-quality-monitoring-with-esp32/

- https://how2electronics.com/ph-meter-using-ph-sensor-arduino-oled/

- https://appinventor.mit.edu