SMART WATER QUALITY MONITORING SYSTEM USING IOT TECHNOLOGY

A Main Project Report

submitted by

MOHAMMED AFNAN PP(MES20MCA-2026)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

July 2022

DECLARATION

I undersigned hereby declare that the project report **Smart Water Quality Monitoring System Using Iot Technology**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Ms. Febin Aziz., Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:Kuttippuram

....

Date:06-07-2022

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **Smart Water Quality Monitoring System Using Iot Technology** is a bona fide record of the Main Project work carried out by **MOHAMMED AFNAN PP(MES20MCA-2026)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

First of all I thank and praise god almighty, for giving me the strength and power to complete my project work without any hindrance and for molding me into what I'm. With great respect I express my sincere thanks to Dr. Rahumathunza.I, Principal, MES College of Engineering, providing facilities for this project. I extend my heartfelt thanks and gratitude to Mr. Hyderali.K, Associate Professor, Head of the Department of Computer Applications.I would also like to express my gratitude towards my supportive and encouraging project coordinator, Ms. Priya J.D, Assistant Professor, Department of Computer Applications.I have great pleasure to express my gratitude to Ms. Febin Aziz, Assistant Professor, internal guide for her valuable suggestions and encouragement. I extend my sincere thanks to all my teachers in the department for their constant encouragement and never ending support throughout the project. Lastly by no means the least, I shall be amiss in not proffering my loving thanks to my dear parents and friends for their encouragement, support, cooperation and good wishes, without which I would have been very sorely tested in leading this project to its successful completion. Well-wishers also gratefully acknowledged.

MOHAMMED AFNAN PP(MES20MCA-2026)

Abstract

The economical and effective system of water quality observation is the most robust implementation of impure water. Drinking water could be precious for all people as water utilities face more challenges. These challenges arise due to the high population, fewer water resources, etc. So, different methods are used to monitor in the real-time water quality. To make sure that safe distribution of water is done, it must be observed in real time for a new method in the "Internet of Things (IoT)" based water quality has been projected. Real-time water quality observation is examined by data acquisition, method, and transmission with an increase in the wireless device network method in the IoT. Microcontroller and the processed values remotely to the core controller ARM with a WI-FI protocol are used to interface the measured values from the sensors. This project the water quality observation interface sensors with quality observation with IOT setting. WQM selects parameters of water like temperature, pH level, water level ,multiple different device nodes. This methodology sends the information to the web server. The data updated at intervals within the server may be retrieved or accessed from anyplace within the world.

Keywords: *pH sensor, Turbidity sensor, Temperature sensor, Waterlevel sensor, Ardurino Board, ESP-32 module.*

Contents

List of Figures

Chapter 1

Introduction

Wireless communication developments are creating new sensor capabilities. The current developments in the field of sensor networks are critical for environmental applications. Internet of Things (IoT) allows connections among various devices with the ability to exchange and gather data. IoT also extends its capability to environmental issues in addition to automation industry by using industry . As water is one of the basic needs of human survival, it is required to incorporate some mechanism to monitor water quality time to time. Around 40 Percentage of deaths are caused due to contaminated water in the world. Hence, there is a necessity to ensure supply of purified drinking water for the people both in cities and villages. Water Quality Monitoring (WQM) is a cost-effective and efficient system designed to monitor drinking water quality which makes use of Internet of Things (IoT) technology. In this paper, the proposed system consists of several sensors to measure various parameters such as pH value, the turbidity in the water, level of water in the tank, temperature and humidity of the surrounding atmosphere. And also, the Microcontroller Unit (MCU) interfaced with these sensors and further processing is performed at Personal Computer (PC). The obtained data is sent to the cloud by using IoT based ThinkSpeak application to monitor the quality of the water.

1.0.1 Motivation

The traditional method for monitoring of the water quality is such that the water sample is taken and sent to the laboratory to be tested manually by analytical methods. Although by this method the chemical, physical, and biological agents of the water can be analyzed, it has several drawbacks. Firstly, it is time consuming and labor intensive. Secondly, the cost for this controlled, displayed, and transferred. Compared to the conventional water quality testing techniques, sensor based water quality testing has many advantages such as accurate, high sensitivity, good selectivity, speed, fast response, low cost etc.

1.1 Objective

Water pollution has been an increasing problem over the last few years. Water personal satisfaction may be a standout amongst those primary variables with control well being and the Lakes and waterways would those fundamental wellsprings about drinking water, which impressively rely on upon water personal satisfaction (refers of the physical, chemical, What's more living aspects about water). The objective of this water quality monitoring system using internet of things is to find the quality of the water i.e. how the pH content varies and And show a graphical representation view to the corresponding authorities. We are going to implement this project at municipal water tanks and drinking water reservoir. For that we are using an Arduino(ESP-32) board for finding module for This technique.Traditionally, detection of water quality was manually performed where water samples were obtained and sent for examination to the laboratories which is time taking process, cost and human resources. The proposed water quality monitoring system is consisting of a microcontroller and basic sensors, is compact and is very useful for pH, turbidity, water level detection, temperature and humidity of the atmosphere, continuous and real-time data sending via wireless technology to the monitoring station Finally the user gets message of pH value of water Further we extend this project by sending the sensor data to cloud for global monitoring of water quality.

1.2 Report Organization

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

Water Quality Monitoring for Rural Areas-A Sensor Cloud Based Economical Project.This paper highlights water quality monitoring methods, sensors, embedded design, and information dissipation procedure, role of government, network operator and villagers in ensuring proper information dissipation. It also explores the Sensor Cloud domain.The economical and effective system of water quality observation is the most robust implementation of impure water. Drinking water could be precious for all people as water utilities face more challenges.

These challenges arise due to the high population, fewer water resources, etc. So, different methods are used to monitor in the real-time water quality. To make sure that safe distribution of water is done, it must be observed in real time for a new method in the "Internet of Things (IoT)" based water quality has been projected. Real-time water quality observation is examined by data acquisition, method, and transmission with an increase in the wireless device network method in the IoT. Microcontroller and the processed values remotely to the core controller ARM with a WI-FI protocol are used to interface the measured values from the sensors. This projected the water quality observation interface sensors with quality observation with IOT setting. WQM selects parameters of water like temperature, pH level, water level ,multiple different device nodes. This methodology sends the information to the web server. The data updated at intervals within the server may be retrieved or accessed from anyplace within the world.

While automatically improving the water quality is not feasible at this point, efficient use of technology and economic practices can help improve water quality and awareness among people. This paper describes to ensure the safe supply of drinking water the quality should be monitored in real time for that purpose new approach IOT (Internet of Things) based water quality monitoring has been proposed. In this paper, we present the design of IOT based water quality monitoring system that monitor the quality of water in real time. This system consists some sensors which measure the water quality parameter such as phSensor,turbidity, temperature parameters of water like temperature, pH level, water level ,multiple different device nodes. This methodology sends the information to the web server. The data updated at intervals within the server may be retrieved or accessed from anyplace within the world. Finally, sensors data can view on internet browser application using cloud computing.

Chapter 3

Methodology

3.1 Introduction

The proposed system uses four sensors which are pH, turbidity, ultrasonic, DHT-11, microcontroller unit as the main processing module and one data transmission module ESP32 Wi-Fi module (NodeMCU).

The microcontroller unit is a significant part of the system developed for water quality measurement because The Arduino uno consumes low power, and it is a small size, where the size is a good use for a crucial point-of-sale technology criterion. Among four sensors, two of the sensors collect the data in the form of analog signals the MCU has an on-chip ADC that translates the sensor analog signals into the digital format for further study. So, to get this analog output from the sensor, the sensor's analog output of will be connected to the MCU's analog pins. Whereas the other two sensors output directly connected to the digital pins of the MCU units. All the sensors data processed by the MCU and updated to the ThingSpeak server using the Wi-Fi data communication module ESP32(NodeMCU) to the central server.

3.2 Developing Environment

1.Hardware Requirement

The selection of hardware is very important in the existence and proper working of any software. Then selection hardware, the size and capacity requirements are also important.

Processor - Dual-Core 32-bit LX6 microprocessor

Speed - 1.1 gh

RAM – 2Gb (min)

Hard disk - 500HDD

Microcontroller-ESP32

Power Requirement-Operating Voltage: 2.2V to 3.6V, On-board 3.3V 600mA regulator

2.Software Requirement

One of the most difficult task is selecting software for the system, once the system requirements is found out then we have to determine whether a particular software package fits for those system requirements. The application requirement:

Operating System - Windows 7 or above

Software Used - Arduino IDE,ThingSpeak

3.3 Embedded System

Embedded systems consist of a micro-controller with on-board memory, a power supply, and communication ports for transmitting data to other devices. Embedded software programs tell the micro-controller how to respond in real time to data collected from the environment through peripheral sensors and devices.

3.3.1 Block Diagram

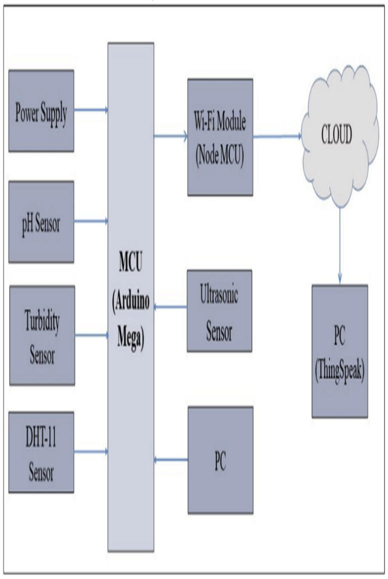

Figure 3.1: Block Diagram

3.3.2 Working structure

The whole system is designed in Embedded-C and simulating the written code using Arduino IDE. Block diagram of system consists of Turbidity sensor, pH sensor, DHT 11 sensor which are connected to processor ESP-32.The ESP-32 accessing the sensors output and processing them. The sensor reading can be viewed on webpage with the help of Wi-Fi.The Turbidity sensor, Ph sensor, DHT 11 sensor check the quality of water. The four sensors are being connected and the values are read into the sensors. The Ultrasonic sensor reads the digital value directly so it is considered as the duration of time in seconds. With the help of the duration, distance is calculated using a equation.

The DHT 11 Sensor reads the analog values of temperature and humidity. Later the same values are sent into the Thing Speak server and the same values are updated in the Serial monitor. The water level detector is used to detect the hight level of water to avoid overflow The data from all sensors is given to ESP32. This data also will be display on webpage. The serial monitor of Arduino is initialized with 115200 baud rate. Later the ESP Wi-Fi module and the Thing Speak Server is also initialized.

Authorized users can access these data using a user ID and password for accessing data on the ThingSpeak server by logging into their accounts. The information is gathered, stored, analyzed and transmitted in real-time.

3.3.3 Target Boards

The target board is a device on which a microcontroller. are fabricated on it. The two target boards are Arduino UNO and ESP32 which are used in the proposed system.

3.3.4 Arduino Uno

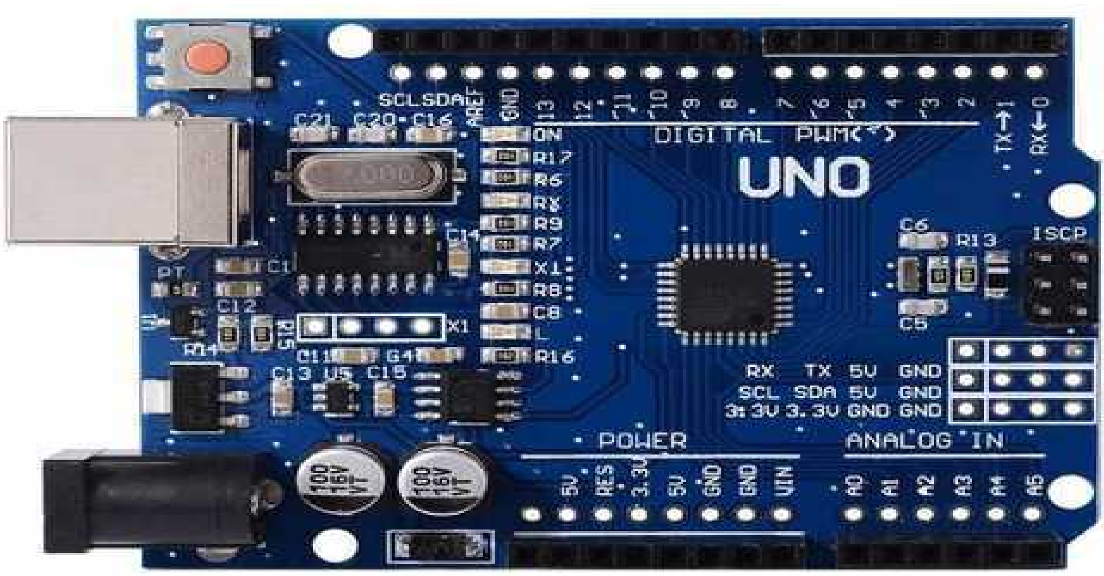

Figure 3.2: Arduino Uno

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins.Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards.

3.3.5 ESP32

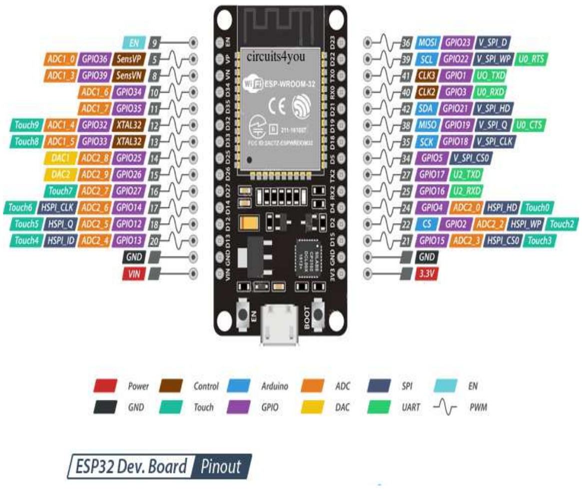

Figure 3.3: ESP32

ESP32 is a series of small and fast microcontroller boards like Arduino but comes with inbuilt Wi-Fi and Bluetooth features.The difference is that the Arduino Zero uses an AT-SAMD21G18 chip with a clock rate of 48 MHz. By comparison, the Esp32 runs on a Tensilica Xtensa LX6 microprocessor chip with a clock rate of between 160 and 240 MHz.

3.4 Sensors

3.4.1 pH sensor

Figure 3.4: pH sensor

A pH sensor helps to measure the acidity or alkalinity of the water with a value between 0-14. When the pH value dips below seven, the water starts to become more acidic. Any number above seven equates to more alkaline. Each type of pH sensor works differently to measure the quality of the water.

3.4.2 Turbidity Sensor

Figure 3.5: Turbidity Sensor

The sensor operates on the principle that when light is passed through a sample of water, the amount of light transmitted through the sample is dependent on the amount of soil in the water. As the soil level increases, the amount of transmitted light decreases.

3.4.3 DHT11 Sensor

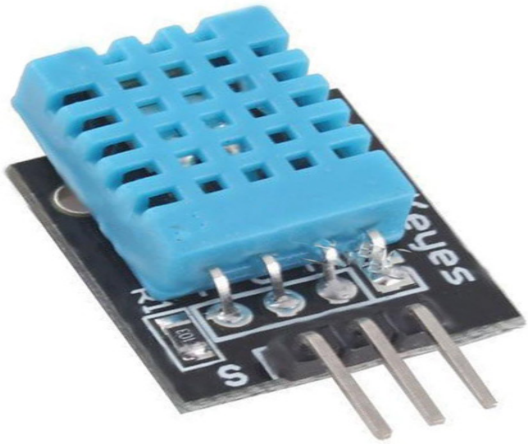

Figure 3.6: DHT11 Sensor

The DHT sensors are made of two parts, a capacitive humidity sensor and a thermistor. There is also a very basic chip inside that does some analog to digital conversion and spits out a digital signal with the temperature and humidity. The digital signal is fairly easy to read using any microcontroller.

3.4.4 Ultrasonic Sensor

Figure 3.7: Ultrasonic Sensor

Ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology . Ultrasonic sensors are also used in water level in a object, as well as manufacturing technology.

3.5 ThingSpeak server

ThingSpeak is an IoT data collection application for analysis of various sensors, e.g. pH, turbidity, voltage, temperature, moisture, distance, etc. The data collector collects data from edge node devices (this happens with the NodeMCU/ESP8266) and also allows the data to be modified for historical data analysis in a software environment. First, the user must log in with details on his/her server. The channel containing data fields and a status field is the primary component of ThingSpeak activity. After a ThingSpeak channel has been developed, data is modified, processed, and interpreted with MATLAB code.

3.6 Algorithm of the proposed system

The proposed system's entire algorithm is shown below. Initially, the serial monitor of Arduino is initialized with 115200 baud rate. Later the ESP Wi-Fi module and the Thing Speak Server is also initialized. The four sensors are being connected and the values are read into the sensors. The algorithm flow of the ultrasonic and DHT 11 sensor flow is explained. The Ultrasonic sensor reads the digital value directly so it is considered as the duration of time in seconds. With the help of the duration, distance is calculated using equations. The DHT 11 Sensor reads the analog values of temperature and humidity. Later the same values are sent into the Thing Speak server and the same values are updated in the Serial monitor.

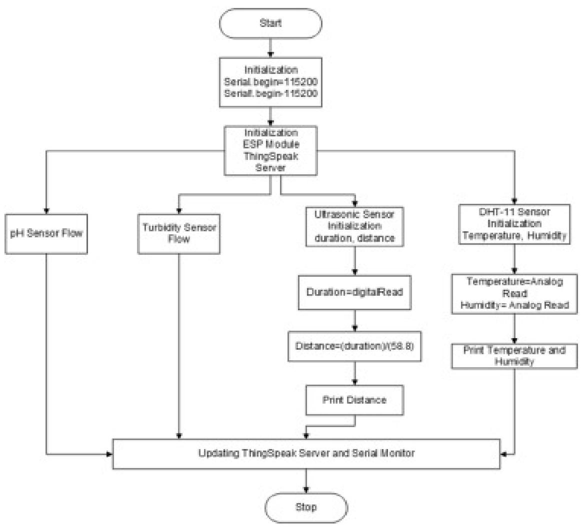

Figure 3.8: Overall Algorithm for the proposed system.

Chapter 4

Results and Discussions

4.1 Result

The experimental setup consists of an MCU with a sensor network that takes samples for every 10s from the water storage tank and the parameters are of water like temperature, pH level, water level,humidity ,multiple different device nodes. displayed on the Arduino IDE serial display. For the real-time monitoring, a Wi-Fi module used which will be updating the ThingSpeak server forever 20s with different parameters. The water sample from Hyderabad Metropolitan water supply and sewerage board and groundwater tested. The entire hardware setup of the Water Quality Monitoring system

4.2 Discussion

The study aimed to design an Arduino-based water quality monitoring system forsmart agriculture. All the modules were created, and all the components were assembled. The device aimed to monitor water quality primarily for smart agriculture by designing,developing, and testing the Arduino-based Water Quality Monitoring Device. Several processes were used to accomplished this goal.

Chapter 5

Conclusions

The system proposed in this Project is an efficient, inexpensive IoT solution for real-time water quality monitoring.The developed system having Arduino and ESP-32 target boards are interfaced with several sensors successfully.The measured pH value ranges from 6.5 to 7.5 of Water. The measured value of turbidity,Humidity and also measure Temperature.

A web-based application ThingSpeak is used to monitor the parameters such as pH value, the turbidity of the water, level of water in the tank, temperature and humidity of the surrounding atmosphere through the webserver. and a Mobile Application Thingview show the graph Measured value.The final test was done was the usability test to measures the ease of using the device was ready to be implemented and ready for use.

References

[1] Sciencedirect. https://www.sciencedirect.com/science/article/pii/S2405844020309403 . Smart water quality monitoring system with cost-effective using IoT

[2] IEEE. https://ieeexplore.ieee.org/document/8480963 .Water Quality Monitoring System Using IOT

[3] Arvin Anthony S. Araneta .International Journal of Computer Science and Mobile Computing, Vol.11 Issue.3

[4] Academia. https://www.academia.edu/76512172/DesignofanArduinoBasedWaterQuality MonitoringSystem

Appendix

Source Code

```
#include <WiFi.h>
#include "ThingSpeak.h"
#include "DHT.h"
#include <NewPing.h>

#define TRIGGER_PIN 18 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN 19 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at
     400-500cm


#define DHTPIN 5
#define DHTTYPE DHT11

unsigned long int avgValue; //Store the average value of the sensor feedback
float b;
int buf[10], temp;

float phValue;

DHT dht(DHTPIN, DHTTYPE);
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.

const char* ssid = "Jiopp112"; // your network SSID (name)
const char* password = "sus4gnripg"; // your network password

WiFiClient client;

unsigned long myChannelNumber = 1742469;
const char * myWriteAPIKey = "5XWKBFHXE72DXBTO";

// Timer variables
unsigned long lastTime = 0;
unsigned long timerDelay = 30000;


void phSens() {
  for (int i = 0; i < 10; i++) //Get 10 sample value from the sensor for smooth the value
  {
    buf[i] = analogRead(34);
    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;
      }
    }
  }
  avgValue = 0;
  for (int i = 2; i < 8; i++)     //take the average value of 6 center sample
    avgValue += buf[i];
  phValue = (float)avgValue * 5.0 / 6286 / 6; //convert the analog into millivolt
  phValue = 3.5 * phValue;        //convert the millivolt into pH value
  Serial.print(" pH:");
  Serial.print(phValue, 2);
  Serial.println(" ");

}

void setup() {
  Serial.begin(115200); //Initialize serial
  WiFi.mode(WIFI_STA);

  ThingSpeak.begin(client); // Initialize ThingSpeak
  dht.begin();
}

void loop() {
  if ((millis() - lastTime) > timerDelay) {

    // Connect or reconnect to WiFi
    if (WiFi.status() != WL_CONNECTED) {
      Serial.print("Attempting to connect");
      while (WiFi.status() != WL_CONNECTED) {
        WiFi.begin(ssid, password);
        delay(5000);
      }
      Serial.println("\nConnected.");
    }



    //   ThingSpeak.setField(3, number3);
    //   ThingSpeak.setField(4, number4);
    float humidity = dht.readHumidity();
    float temperature = dht.readTemperature();
    delay(100);
    int turbidityvalue = analogRead(35);
    int turbidity = map(turbidityvalue, 0, 6286, 0, 100);
    delay(100);
    int distance = sonar.ping_cm();
    int waterlevel = map(distance, 0, 100, 100, 0);
    delay(100);
    phSens();



    Serial.print(F("Humidity: "));
    Serial.print(humidity);
    Serial.print(F("% Temperature: "));
    Serial.print(temperature);
    Serial.println(F("C "));
```

```
    Serial.print("Turbidity: ");
    Serial.print(turbidity);
    Serial.print("% ph Value: ");
    Serial.println(phValue);
    Serial.print("Water Level: ");
    Serial.println(waterlevel);



    ThingSpeak.setField(1, temperature);
    ThingSpeak.setField(2, humidity);
    ThingSpeak.setField(3, phValue);
    ThingSpeak.setField(4, turbidity);
    ThingSpeak.setField(5, waterlevel);
    // Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different
    // pieces of information in a channel. Here, we write to field 1.
    int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

    //uncomment if you want to get temperature in Fahrenheit
    //int x = ThingSpeak.writeField(myChannelNumber, 1, temperatureF, myWriteAPIKey);

    if (x == 200) {
     Serial.println("Channel update successful.");
    }
    else {
     Serial.println("Problem updating channel. HTTP error code " + String(x));
    }
    lastTime = millis();
  }
}
```

Project Plan

User story ID	Task name	Start date	End date	Days	Status
			Sprint1		
1	Planning	20/04/2022	02/05/2022	3	Completed
2	Hardware Purchasing	03/05/2022	09/05/2022	2	Completed
3	Hardware Testing	10/05/2022	15/05/2022	5	Completed
			Sprint2		
4	Circuit designing	15/05/2022	18/05/2022	2	Completed
5	Coding	19/05/2022	21/05/2022	1	Completed
6	PCB Designing	22/05/2022	29/05/2022	6	Completed

User story ID	Task name	Start date	End date	Days	Status
			Sprint3		
7	PCB designing	01/06/2022	05/06/2022	2	Completed
8	Coding	28/05/2022	20/06/2022	1	Completed
9	Soldering	21/06/2022	24/06/2022	9	Completed
10	Testing& Output generation	25/06/2022	30/06/2022	1	Completed

Figure A.1: Project plan

Product Backlog

User Story ID	Priority <High/Medium/Low >	Size (Hours)	Sprint <#>	Status <Planned/In progress/Completed>	Release Date	Release Goal
1	**Medium**	**2**	**1**	**Completed**	**02/05/20 22**	**Planning**
2	**High**	**3**		**Completed**	**09/05/20 22**	**Hardware Purchasing**
3	**High**	**5**		**Completed**	**08/01/20 22**	**Basic coding**
3	**High**	**5**	**2**	**Completed**	**18/05/20 22**	**Circuit Designing**
4	**Medium**	**5**		**Completed**	**20/05/20 22**	**Preprocessing**
5	**High**	**5**	**3**	**Completed**	**04/06/20 22**	**PCB designing**
6	**medium**	**5**		**Completed**	**05/05/20 22**	**Page designing**
7	**Medium**	**5**	**4**	**Completed**	**24/06/20 22**	**Soldering**
8	**High**	**5**		**Completed**	**30/06/20 22**	**Output generation**

Figure A.2: Product Backlog

Sprint Backlog Plan

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
#1,#2,#3,#4		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Planning	25/04/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Hardware Purchasing	05/05/2022	3	0	0	0	1	1	1	0	0	0	0	0	0	0	0
page design	08/05/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Hardware Testing	13/05/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Circuit designing	17/05/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
PCB Designing	25/05/2022	5	0	0	0	0	0	0	0	1	0	1	1	1	0	1
Soldering	23/06/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Testing& Out Put Generation	28/06/2022	5	0	0	0	0	0	0	0	0	0	2	1	1	1	1
Total		40	4	4	2	4	3	2	0	2	0	5	4	4	3	4

Figure A.3: Sprint Backlog Plan

Sprint Actual

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
#1,#2,#3,#4		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Planning	25/04/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Hardware Purchasing	05/05/2022	3	0	0	0	1	1	1	0	0	0	0	0	0	0	0
page design	08/05/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Hardware Testing																
Circuit designing																
PCB Designing																
Soldering																
Testing& Out Put Generation																
Total		15	1	1	0	1	1	1	0	0	0	1	1	1	1	1

Figure A.4: Sprint Actual 1

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
#1,#2,#3,#4		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Planning	25/04/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Hardware Purchasing	05/05/2022	3	0	0	0	1	1	1	0	0	0	0	0	0	0	0
page design	08/05/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Hardware Testing	13/05/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Circuit designing	17/05/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
PCB Designing																
Soldering																
Testing& Out Put Generation																
Total		25	3	3	1	2	2	1	0	0	0	2	2	2	2	2

Figure A.5: Sprint Actual 2

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
#1,#2,#3,#4		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Planning	25/04/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Hardware Purchasing	05/05/2022	3	0	0	0	1	1	1	0	0	0	0	0	0	0	0
page design	08/05/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Hardware Testing	13/05/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Circuit designing	17/05/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
PCB Designing	25/05/2022	5	0	0	0	0	0	0	0	1	0	1	1	1	0	1
Soldering	23/06/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Testing& Out Put Generation	28/06/2022	5	0	0	0	0	0	0	0	0	0	2	1	1	1	1
Total		40	4	4	2	4	3	2	0	2	0	5	4	4	3	4

Figure A.6: Sprint Actual 3

Screen Shots

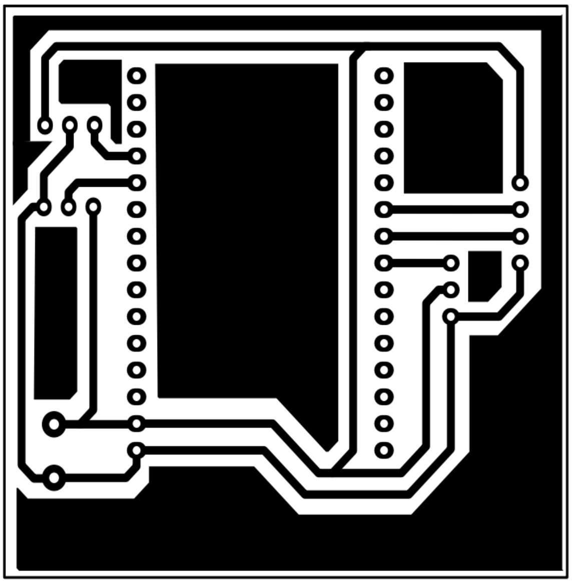

Figure A.7: PCB Design

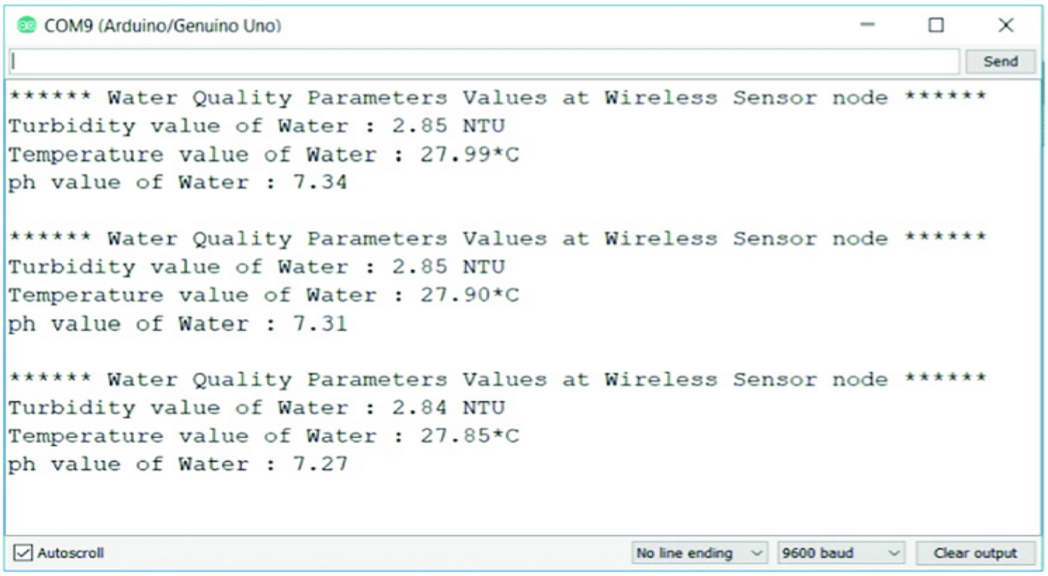

Figure A.8: Serial Monitor

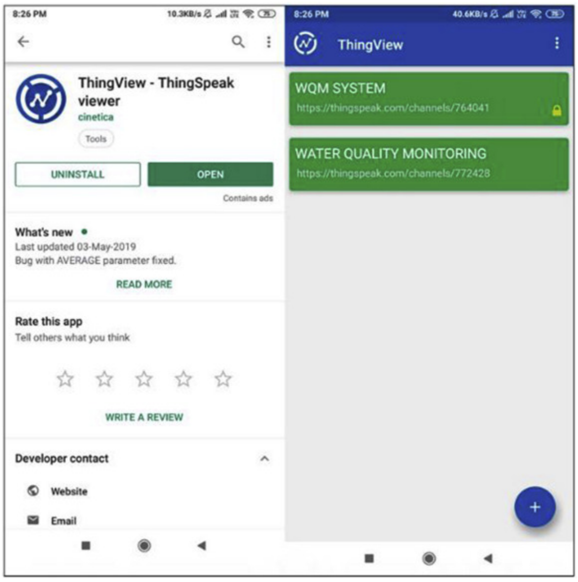

Figure A.9: Thing View 1

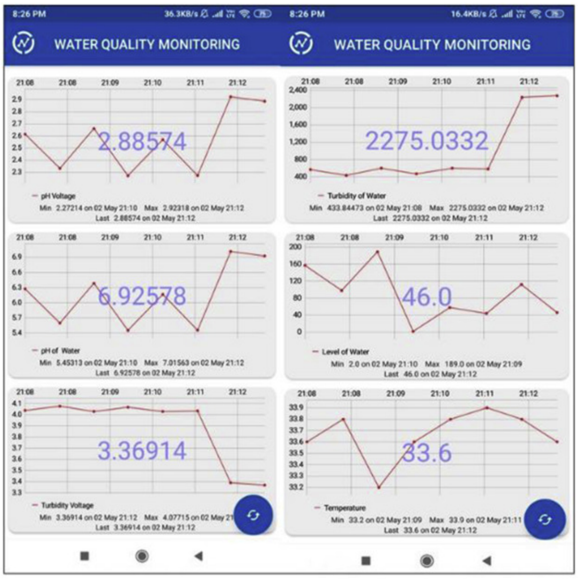

Figure A.10: Thing View 2