Palestine Polytechnic University

College of IT and Computer Engineering

Computer System Engineering Department

Graduation Project Name

WATER QUALITY MONITORING SYSTEM

Project Team

Islam Shweiky **Shereen Abu Yousef** **Hazem Salah**

Project Supervisor

ENG. Elayan Abu Gharbyeh

June 2021

Acknowledgement

First of all, praise be to God "Allah" Almighty and thank him for his blessings throughout our work on our project to give us strength and knowledge and help us succeed in this project.

We are extremely grateful to those whom the pleasure of Allah is in the pleasure of them our parents for their love, prayers, caring and sacrifices for educating and preparing us for our future. We know the thanks are not enough and there are not enough words to describe how thankful we are to them.

We would like to express our deepest gratitude to our graduation project supervisor Dr. Elayan Abu-Gharbyeh for his patience and guidance and help in the project.

Moreover, it is our duty to thank Dr. Wael Takrouri to help us choose a project idea AND his response and cooperation.

We would like to thank ENG. Iyad Shweiky for all the useful information and the beneficial learning that he gave us.

Our thanks to Water Quality Unit at Palestine Polytechnic University for all the useful information they presented to us.

Special thanks to all the family members of the university staff and the educational unit for the efforts they made to provide us with all useful information and the interesting lectures they presented which had great benefit for all of us.

At last, but not the least, we would like to thank all the people who helped, supported and encouraged us to successfully finish the graduation project whether they were in the university or in the life.

Dedication

We dedicate this project to God Almighty, our Creator, our strong pillar, our source of inspiration, wisdom, knowledge and understanding.

Our great teacher and messenger Muhammad (may God bless him and grant him peace) who taught us the purpose of life.

Our homeland, Palestine, is the warmest mother. Great martyrs and prisoners are a symbol of sacrifice.

To our great parents, who never stop giving themselves in countless numbers and gave us strength when we thought of giving up, who are constantly providing their moral, spiritual, emotional, and inner support.

To Rouh Issa Al-Julani, the grandfather of our colleague Islam Shweiky, may God have mercy on him, and Rouh Izdehar, the aunt of our colleague Hazem Salah, may God have mercy on him.

Our brothers and sisters and our beloved friends who encourage and support us, each of our family has a symbol of love and giving, whoever touches our hearts in our lives.

We dedicate this project.

Abstract

In today's times, due to urbanization and pollution, it has become necessary to monitor and evaluate the quality of water reaching our homes and life. Water pollution is one of the biggest fears for the green globalization. In order to ensure the safe supply of the drinking water the quality needs to be monitor in real time. The conventional method of water quality assurance that involve collecting water samples manually and send it to laboratory for testing and analysis, which are not only costly but also time consuming and lack speedy distribution of information to the consumer or the concerned party for making timely and informed decisions, and is prone to human errors.

Through this project, we aim to solve these problems to ensure water quality by building a Water Quality Monitoring system (WQM) that helps in measurement of water status based on four water quality parameters (Temperature, PH, Total Dissolved Solids (TDS) and Turbidity) have been selected to monitor the water quality. Four sensors are connected to the Raspberry pi in a separate way to detect water parameters [1].

The system can transfer the data extracted from the sensors to the mobile app and web page using Raspberry pi Wi-fi. Based on the measured result, the (WQM) system can successfully analyze water parameters to classify whether or not the test water sample is potable, and issue warnings in cases that may not be potable.

This project combined embedded systems with the Internet of Things (IoT) as Industrial Internet of Things (IIoT) applications in real-time operation.

Table of Contents

List of Tables:

List of Figure:

Chapter 1: Introduction

1.1. Overview of The Project

Environment around us consists of five key elements. These are soil, water, climate, natural vegetation and landforms. Among this water is the most essential element for humans to live.

Water is also important for the survival of other living habitants. Whether it is used for drinking, domestic use, and food production or recreational purposes, safe and readily available water is a must for public health.

God Almighty says in the Qur'an ﴿وَجَعَلْنَا مِنَ الْمَاءِ كُلَّ شَيْءٍ حَيٍّ﴾ سورة الأنبياء الآية: 30

So, it is highly imperative for us to maintain water quality balance. Otherwise, it would severely damage the health of the humans and at the same time affect the ecological balance among other species.

Currently, water facilities and resources face new problems in the real world. Due to limited drinking water resources, intense financial requirements, a growing population, urban change in rural areas, and excessive use of marine resources for salt extraction, the quality of the water available to the people has greatly deteriorated. That makes us think deeply to solve all these problems using a Water Quality Monitoring System.

1.2. Motivation

Water is the most important element of life, so making sure of its quality is our top priority. The first thing that must be done is to monitor the quality of the water to protect it from contamination and make sure that it is clean, especially the water used for drinking and food. And our motives went to this idea:

1) Water pollution poses many risks to living organisms, including humans and animals. As it is responsible for 40% of the death rate around the world, it also affects the economy in terms of the costs of treating diseases resulting from pollution. As for soils, which work on deficiency of nutrients in the soil and increase its salinity and thus impede the growth of plants, and to detect pollutants and monitor water.

2) Water should be stored in a very efficient and high-quality manner in order to be clean and away of pollutants.
3) In cases where the water is polluted and cannot be used, the user must know the cause of the pollution and solve it, so finding several ways to keep the water quality, and without wasting can be one of the most important social tasks.

1.3. Project Importance

What makes our project important to the public sector and society is the following:

1) It is extremely important to ensure that the water is clean and in good condition.
2) Our system provides real-time monitoring of water quality. To prevent all the wasted time that can result from laboratory tests or delays and a lack of knowledge of the unfit for use of water. This can help in preventing diseases caused due to polluted water and presence of metals in it.
3) Our system provides a valuable database, which can be compared to a variety of processes due to its resolution and is extremely useful for understanding the drinking water supply system. The possibility of early warnings when changes or contaminations occur. And ensures a reliable water supply and quick actions can be taken to curb extreme levels of pollution.

1.4. Objectives

Our aim through this project is to ensure:

1) The ability to monitor water quality using a low-cost, wireless water quality monitoring system that aids in continuous measurements of water conditions.
2) Provide a detail of recent work carried out in the water quality monitoring in terms of application, communication technology used, types of sensors employed etc.
3) Implement an alert system and notifications to announce the water status based on readings from sensors.
4) Help the users and the communities reducing the problem of water pollution.
5) Attempt to mitigate the damage of water pollution and its exacerbation due to the speed in dealing with the damage.
6) Ensuring that the System works completely without issues.

1.5. Problem Statement

1.5.1 Definition

Water pollution affects human health by causing waterborne diseases, death and dehydration. Especially since the water may be exposed to many causes of pollution at any time continuously, and in the case of water pollution it will be difficult to know without examining it, and examining the water may take a long time.

1.5.2 Problem Analysis

The safety of water quality in homes, factories, offices and buildings is important to human life. Therefore, water pollution causes risks and disasters to human life and activities because many of the water sources used by humans contain vectors and factors that lead to long-term diseases or health problems if they do not meet the drinking water quality standards. In addition to the large economic losses, it causes.

Natural or chemical water pollution, or other factors that cannot be detected by human sensitivity, and water pollution often occurs without our awareness of it, and there is a need to solve this problem.

1.5.3 List of Requirements

The system should have the following requirements:

1) Some sensors may need special circuits to work like (Temperature Sensor need resistor) (pH, TDS and Turbidity Sensors need modules).
2) The system must show warning notifications if the water is experiencing unsuitable conditions.
3) Real time database and alerting notification can be viewed through mobile phones.
4) Mobile application should meet the following needs:
    - Show the user a menu to choose the parameter name of testing.
    - Show the user water quality assessment and result of testing.

1.5.4 Expected Results

After development of the system, the following results are expected to be achieved:

1) The water quality monitoring system detect changes in water characteristics and show a notification to the concerned user about the type of problem to ensure an effective and timely response.
2) Reducing time and reducing costs in all its forms, from the costs of wasting polluted water, the costs of water damage and the costs of conducting water checks.
3) Measurement data is collected periodically, location coordinates are obtained, and packet data is uploaded to a database.
4) After sensing the data from different sensor devices, which are placed in particular area of interest or in water sample. The sensed data will be shown on the screen of lcd display and automatically sent to the web server, when a proper connection is established with the server device. Where users can go to the website or using application to view the result.

1.6. Description of The System

In this project, the main idea is to build a development system for monitoring the water quality in IOT (Internet of Things), by installing an electronic monitoring device that keeps a close eye on water quality. This will keep the stored water or sample of water monitored in case contamination occurs. In case there is anything polluting the water or out of the recommended range, a warning notification with description of the problem will be shown to the user. Another role of the system is to record the sensor readings and export daily and monthly records to a database that projects it on application, users can go to the website to view the result or by application presents a Google Map, with markers shown in locations which have measurements data. The model is based on the idea that the water quality can be a very important parameter when it is used.

1.7. Overview of The Rest of Report

The outline of the report consists of six chapters; the following is a brief description of the topics that are covered in each other next chapters:

Chapter 2: “Background”, contains the theoretical background and Literature review, options (design options for hardware components and design options for software components) and design constraints.

Chapter 3: “Design”, includes a detailed conceptual description of the system (HW and SW), detailed design, schematic diagrams, block diagrams, structural diagrams, and any necessary information about the design.

Chapter 4: “Software”, This chapter describes the implementations of the software that are used in this project, application and webpage.

Chapter 5: “Validation and Discussion”, This chapter describes the implementation, implementation issues, and implementation challenges. Also include description of the method used to validate the system, validation results, analysis and discussion about the results, and recommendations.

Chapter 6: “Conclusion”, In this chapter gives information about conclusion and provides for more feature that can be done in the future.

Key Words:

pH: potential of hydrogen, is a measure of how acidic/basic water is, the allowed limit (6.5-8.5) [1].

TDS: Total Dissolved Solids comprise inorganic salts (principally calcium, magnesium, potassium, sodium, bicarbonates, chlorides, and sulfates) and some small amounts of organic matter that are dissolved in water, the maximum allowed (1000 ML /L) [1].

Turbidity Water: Turbidity is the measure of relative clarity of a liquid it is measured in NTU: Nephelometric Turbidity Units, the maximum allowed (1 NTU) [1].

Chapter 2: Background

2.1. Overview:

This chapter introduces a theoretical background of the project, short description of design options that will be used in the system, discussion of the specification and design constraints and some additional information about the system.

2.2. Theoretical background:

INTERNET OF THINGS

The internet has changed all the human lives in past deader. The Internet of Things, or "IoT" for short, is the extension of Internet connectivity beyond computers and smartphones to a whole range of other things, processes and environments such as physical devices and everyday objects. Embedded with electronics, Internet connectivity and other forms of hardware such as sensors, these devices can communicate and interact with others over the Internet, and can be remotely monitored & controlled [2].

IoT has its applications in a large number of areas such as consumer applications like smart homes, wearable technology and connected vehicles, commercial applications like medical and healthcare and industrial applications like manufacturing, agriculture, energy management and environmental monitoring.

In our system, we employ cloud computing technique for monitoring sensor values on the Internet. Cloud computing Provides the access to applications as utilities, over the Internet.

Cloud computing is a large-scale, low-cost processing unit, which is based on the IP a connection for calculation and storage. The IoT application areas include home automation, water environment monitoring, and water quality monitoring etc. the water quality monitoring application involves large distributed array of monitoring sensor and a large distribution network. In our proposed method, we introduced a cloud computing technique for viewing sensor values on the internet [3].

2.3. Literature review:

This section reviews some of the previously proposed solutions.

- A scientific paper published in 2015 1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India. Nikhil Kedia entitled "Water Quality Monitoring for Rural Areas-A Sensor Cloud Based Economical Project." This paper highlights the entire 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. 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 [4].

  In our project, we will follow the same methodology but we will use raspberry pi 4 to implement the design instead of Arduino UNO and ESP8266.

- A scientific paper published in 2012, Sokratis Kartakis, Weiren Yu, Reza Akhavan, and Julie A. McCann entitled "Adaptive Edge Analytics for Distributed Networked Control of Water Systems" This paper presents the burst detection and localization scheme that combines lightweight compression and anomaly detection with graph topology analytics for water distribution networks. They show that our approach not only significantly reduces the number of communications between sensor devices and the back-end servers, but also can effectively localize water burst events by using the difference in the arrival times of the vibration variations detected at sensor locations. There results can save up to 90% communications compared with traditional periodical reporting situations [5].

  However, our system is based onto the design of system that capable to ensure the safe supply of water. The design of water quality monitoring system that monitor the quality of water in real time consists some sensors which measure the water quality parameter such as pH, turbidity, temperature. The measured values from the sensors are processed by microcontroller and these processed values are transmitted remotely to the core controller that is raspberry pi and if their problem the system will show a notification specifying the problem. Finally, sensors data can view on internet website and application using cloud computing.

2.4. Design Options and Specification

This section describes all of the hardware used in our project, it presents a figure for each one with a short description of its work principle and why it is used in the system.

2.4.1.Hardware components

Microcontrollers:

There are many microcontrollers and one microcomputer can be used in our project.

- First Design Option: Arduino:

  Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. Arduino makes several different boards, each with different capabilities (Arduino Uno, Arduino Mega2560, Arduino Leonardo) and others [6].

- Second Design Option: Raspberry pi:

  Is a small, single-board computer, has a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware video decodes at up to 4Kp60, 2GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on). The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time [7].

Chosen Design Option:

We will choose a Raspberry pi, as shown in Figure 2.1, It includes an Ethernet port, a Wi-Fi antenna which you can connect to the network ,2 USB-2 ports and 2 USB-3 ports, A 40-pin GPIO header. It includes everything required to hold up the system as presented in Table 2.1 and previously available with us. So, it's great for hardware projects in which simply want things to respond to various sensor readings and manual input.

Power Draw/Voltage	**2.5A at 5V**
RAM	**2GB**
USB 2.0 ports	**2**
USB 3.0 ports	**2**
Ethernet Port	**Yes**
Wireless LAN	**802.11ac**
Bluetooth	**Bluetooth 5.0**
GPIO	**40 pins**

Table 2.1: Raspberry pi 4 model B Specification

Figure 2.1: Raspberry pi 4 model B [6]

Display Screen:

There are two display Screens can be used in our project.

- First Design Option: TFT: The TFT screen is a backlit LCD screen with headers, store bitmap images for the screen to display. The screen is 1.77" diagonal, with 160 x 128-pixel resolution. The screen runs on +5 VDC the LED backlight is dimmable by PWM.
- Second Design Option: LCD (20 x 4): A liquid-crystal display (LCD) is a flat panel display, electronic visual display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly. 20x4 means that 20 characters can be displayed in each of the 4 rows of the 20x4, thus a total of 80 characters can be displayed at any instance of time.

Chosen Design Option:

We will choose an LCD (20 x 4), as shown in Figure 2.2. Because its display format: 20 characters in 4-line, lowest cost, easiest to use and the most suitable size and specifications as presented in Table 2.2 [8].

Power Supply	**+5V**
Controller	**HD44780 or equivalent**
LCD Size	**98 x 60 mm**
LCD Thickness	**8.8 mm**
Viewing Area	**20 Characters in 4 Lines**
Duty Circle	**1/16**

Table 0.2: LCD (20 x 4) Specifications

Figure 2.2: LCD (20 x 4) [8]

TDS Sensor:

The Grove - TDS Sensor as shown in Figure 2.3 and specifications as presented in Table 2.3, detects the Total Dissolved Solids (TDS) levels in the water which can be used to indicate the water quality. Total Dissolved Solids, is a measure of the dissolved combined content of all inorganic and organic substances present in water. Typically, the higher the TDS value, the more substances dissolved in water. Hence, higher levels of Total Dissolved Solids (TDS) can indicate that water has more contaminants that can pose health risks [9].

Input Voltage	**3.3 ~ 5.5V**
Output Voltage	**0 ~ 2.3V**
TDS Measurement Range	**0 ~ 1000ppm**
TDS Measurement Accuracy	**± 10% F.S. (25 ℃)**
Module Interface	**PH2.0-3P**
Electrode Interface	**XH2.54-2P**

Table 2.3: TDS Sensor Specification

Figure 2.3: TDS Sensor [9]

Temperature Sensor:

The DS18B20 digital thermometer as shown in Figure 2.4, provides 9-bit to 12-bit Celsius temperature measurements indicates how water is hot or cold. The range of DS18B20 temperature sensor is -55 to +125 °C. This temperature sensor is digital type which gives accurate reading, and has an alarm function with non-volatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line, eliminating the need for an external power supply3 and the specifications as presented in Table 2.4 [10].

Power Supply	**3.3 ~ 5.5V**
Communication	**1-Wire interface**
Usable temperature range	**-55 to 125°C**
Accuracy	**±0.5°C from -10°C to +85°C**
Query time	**Less than 750ms**

Table 2.4: Temperature Sensor DS18B20 Specification

Figure 2.4: Temperature Sensor DS18B20 [10]

Turbidity Sensor:

The turbidity sensor as shown in Figure 2.53 and specifications as presented in Table 2.5, detects water quality by measuring the level of turbidity. It is able to detect suspended particles in water by measuring the light transmittance and scattering rate which changes with the amount of total suspended solids (TSS) in water. As the TTS increases, the liquid turbidity level increases.

This turbidity sensor has both analog and digital signal output modes. You can select the mode according to the MCU as the threshold is adjustable in digital signal mode [11].

Operating Voltage	**+5V**
Response Time	**<500ms**
Output Method	• **Analog output: 0-4.5V** • **Digital Output: High/Low level signal**
Operating Temperature	**5℃~90℃**
Operating Current	**40mA (MAX)**

Table 2.5: Turbidity Sensor Specification

Figure 2.5: Turbidity Sensor [11]

PH Sensor:

The PH sensor as shown in Figure 2.6 and specifications as presented in Table 2.6, used for the measurement of the hydrogen potential or pH is the electrode of glass, which is made up of an ion-selective membrane, made of glass, permeable to hydrogen. The electrochemical process generated inside the sensor allows to generate a potential difference according to the Nernst equation and that is related to the potential of hydrogen. The PH electrode has a single cylinder that allows direct connection to the input terminal of a PH, controller, or any PH device which has a BNC input terminal [12].

Operating Voltage	**+5V**
Response Time	**≤ 5 S**
The detection concentration range	**PH0-14**
Operating Temperature	**-10℃~50℃**
Output	**Analog voltage Signal**
Settling time	**≤ 60 S**

Table 2.6: PH Sensor Specification

Figure 2.6: PH sensor [12]

GPS Module:

The NEO-6MV2, as shown in Figure 2.7, is a GPS (Global Positioning System) module and is used for navigation. The module simply checks its location on earth and provides output data which is longitude and latitude of its position. The compact architecture, power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints. Its Innovative design gives NEO-6MV2 excellent navigation performance even in the most challenging environments and specifications as presented in Table 2.7 [13].

Power Supply	**3.6V**
Receiver type	**50 Channels**
Operating Temperature	**-40ºC TO 85°C**
Default baud rate	**9600bps**
Sensitivity	**-160dBm**

Table 2.7: GPS Module NEO-6MV2

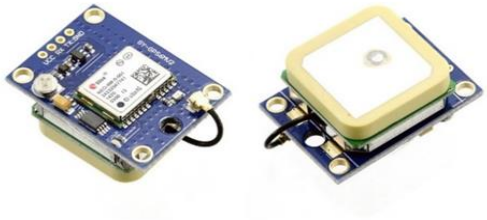


Figure 2.7: GPS Module NEO-6MV2 [13]

Analog to Digital Converter:

The Raspberry Pi computer does not have a way to read analog inputs. It's a digital-only computer. In our system we have 3 sensors with analog outputs, so we need a way to make the Pi analog-friendly.

- First Design Option: MCP3008:

  The MCP3008 is a low cost 8-channel 10-bit acts like a "bridge" between digital and analog and the Pi can query it using 4 digital pins. That makes it a perfect addition to the Pi for integrating simple sensors [14].

- Second Design Option: ADS1115:

  The ADS1115 provides 4 16-bit ADCs, 15 for the measurement and one last for the sign of the analog signal. The ADS1115 is connected by I2C. It has 4 addresses, which is chosen by connecting the ADDRESS pin. It has a PGA that can adjust the voltage gain from 6.144v to 0.256v [15].

Chosen Design Option:

We will choose ADS1115, as shown in Figure 2.8, because it's easy to use with the Raspberry Pi using its I2C communication bus. It is a 4-channel analog-to-digital converter and we only need three.

Figure 2.8: 16-Bit ADC GY-ADS1115 [15]

Power Source and Cooler:

We chose to connect the device to an adapter as shown in Figure 2.9.

Figure 2.9: Raspberry Pi 4 Adapter

To keep the Raspberry Pi 4 running cool, we use the Aluminum Metal Heatsink Raspberry Pi 4 Case with Dual Fans as shown in Figure 2.10.

Figure 2.10: Aluminum Heatsink with Dual Fans

2.4.2.Software Options:

This section provides some information about the main programs and software technologies used in our system.

In this project we used a raspberry pi as a main processor, the Raspberry Pi itself doesn't come with an operating system. For that, we have to install an operating system, which depends on the project idea .in our project we installed a Raspbian operating system.

Raspbian operating system, as shown in Figure 2.11, which is an operating system with the set of programs and utilities that makes the Raspberry Pi run. It is a free OS based on Debian which uses the command line to manage the operation and is optimized to be compatible with the Raspberry pi hardware [16].

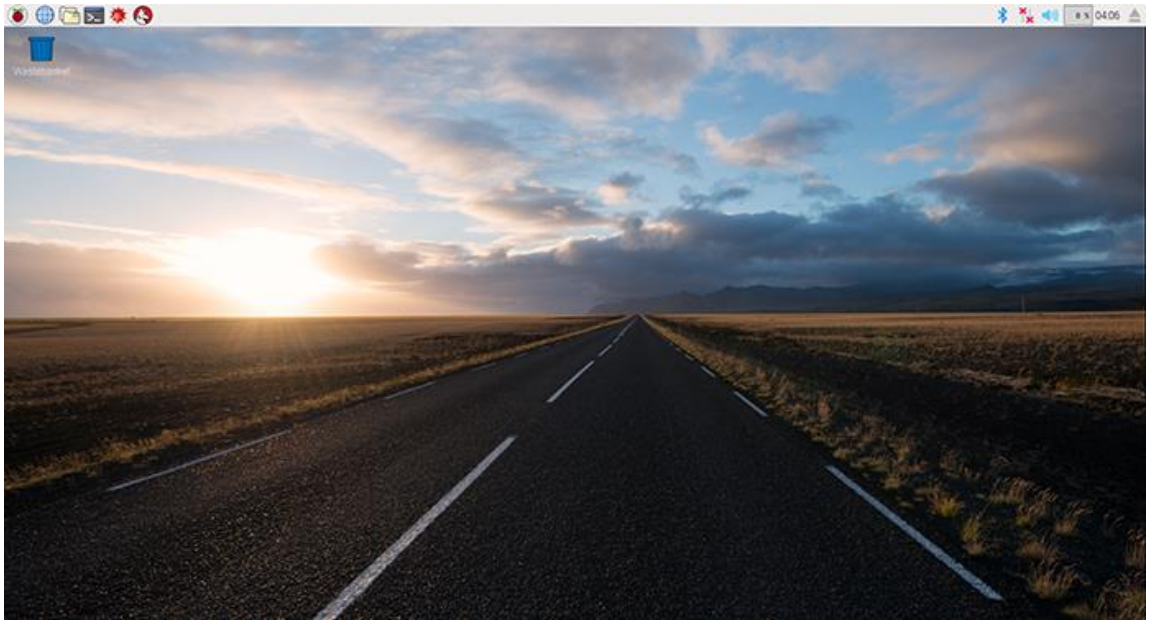


Figure 2.11: Raspbian Operating System Interface

Software used for hardware and sensors:

Python: The best programming language for Raspberry pi, Python is a wonderful and powerful programming language that's easy to use (easy to read and write). Python syntax is very clean, with an emphasis on readability and uses standard English keywords.

Software used for Webpage and Application:

The software that can be used in the Webpage and Application are PHP, MySQL, CSS, Flutter, Dart programming language, API and HTML.

HTML makes up the layout and structure of website. This language is dynamic and allows you to create a beautiful website using less code. HTML is used to create a starting point for the website and is what most of static pages start from.

CSS is used to define styles for web page, including the design, layout and variations in display for different devices and screen sizes.

PHP is used for data-heavy website and app development. This is an open-source language that can be easily modified to meet the needs of our system.

Flutter it is a cross platform developed by Google and is updated periodically, as it has become in the latest update that it supports running on many operating systems such as Windows, Linux and the web, and this is a strong point in the flutter.

MySQL database it is an easy-to-use database that supports many features such as Online Connection by uploading the database on a server that can be accessed via the Internet.

Dart programming language it is a language adopted by Google and used in programming the back-end in flutter.

API: we used it to link between the databases on the server and the mobile application, and it was programmed using the php language.

2.5. Design constraints:

- The system can’t send data or notification without Wi-Fi connection.
- The system must not disconnect from the power during the monitoring water.
- That must to be ensure that all sensors are immersed in water prior to testing.

Chapter 3: Design

3.1. Overview

This chapter discusses the conceptual design of the system. It shows the system requirement analysis, block diagram of the system, flow chart, detailed design and schematic diagrams.

3.2. Detailed Design

Figure 3.2.1 shows the functional block diagram of the whole system and how the system's components work with each other to achieve the needed requirements of our project.

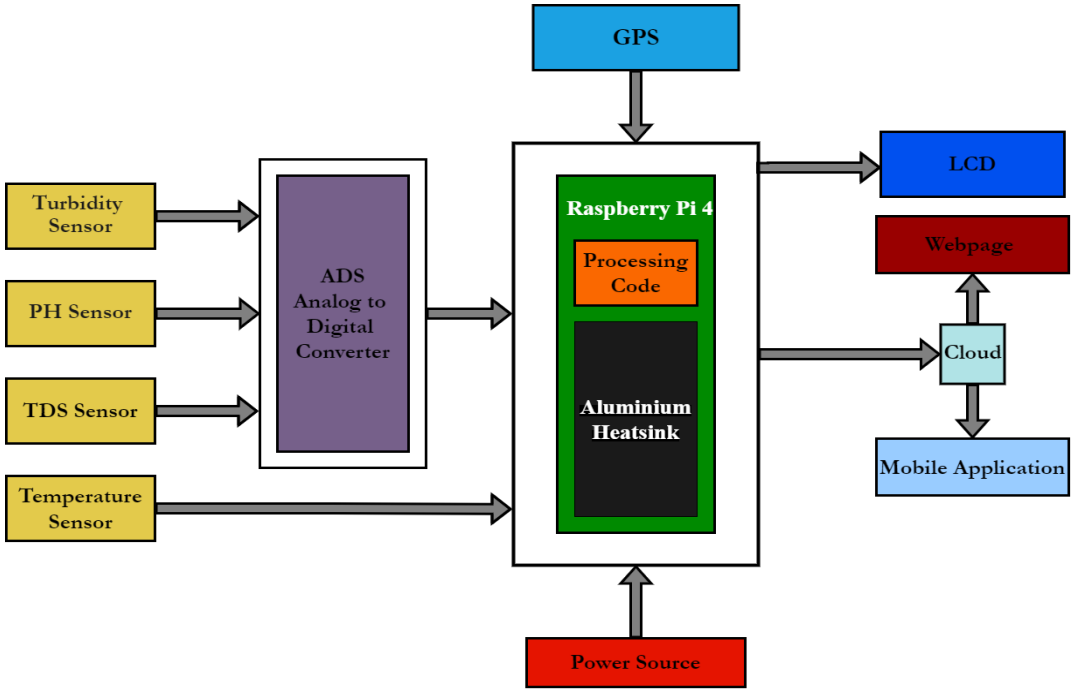


Figure 3.2.1: System Block Diagram

The Raspberry pi connect to the power source, and four types of sensors have been used to define the specific parameters that illustrate the water quality: PH sensor, Turbidity sensor, TDS sensor (Total Dissolved Solids) which are connected to ADS converter to convert their readings into digital. Temperature sensor connect directly to Raspberry pi. The Raspberry pi accesses and processes sensor values to transfer the data through internet. The processing data from sensors can be displayed on LCD display, app or Webpage. The system will also be able to Show an alarm to notify about anything strange in water quality with the coordinates of the exact location. An Aluminum Heatsink is required to cool the Raspberry pi.

3.3. Flow Chart:

The flowchart of the system in figure 3.3.1 shows the steps of the system. Starting with Establish connection between Raspberry Pi and Wi-Fi Network. Then initialize the sensors and getting value from them and send that results to Raspberry Pi to process it.

After processing the result of sensors, the output data shows on LCD, then its upload on the cloud to comparing it with threshold values to determine the quality of water, if anything out of specific range the system will show alarm notification to user with report about problem if not the result will display on application and website with report for the user.

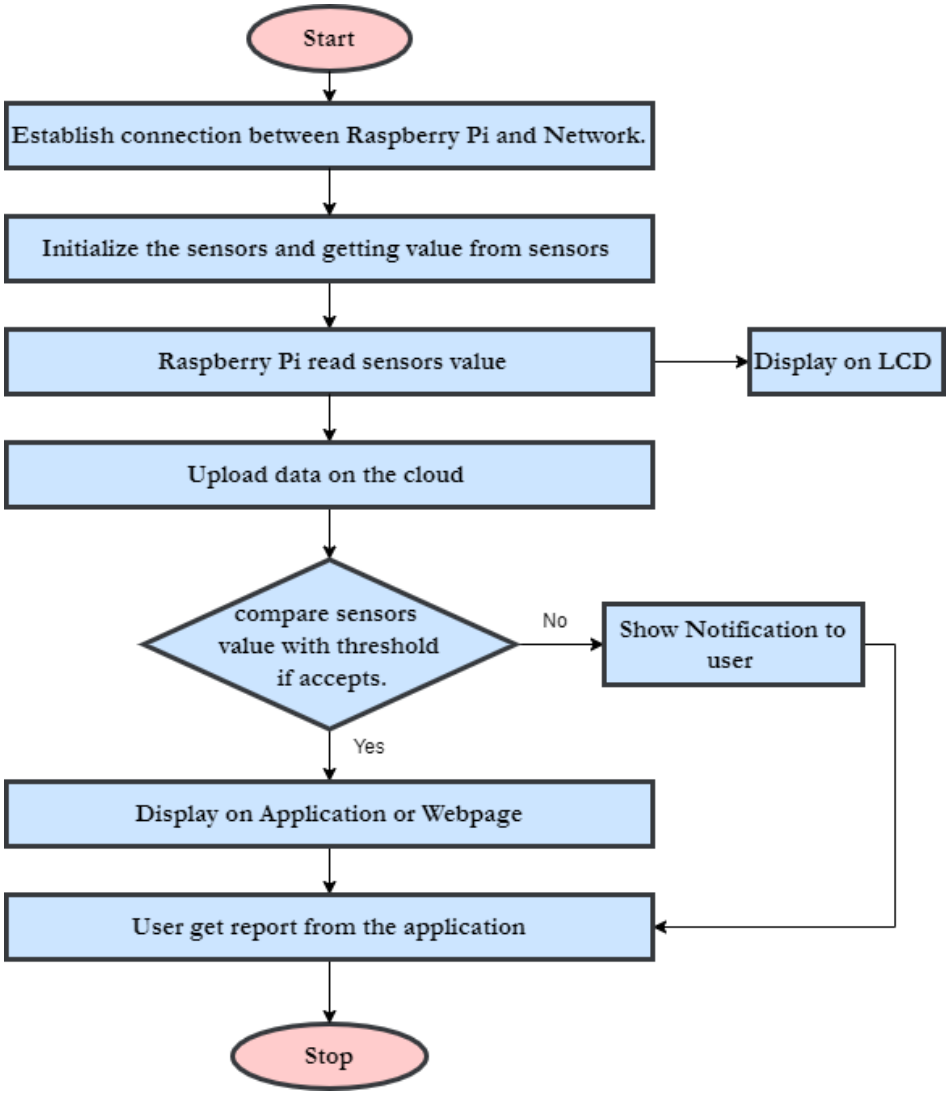


Figure 3.3.1: Flow chart of system activity

3.4. System Schematic Diagrams:

3.4.1. LCD Display

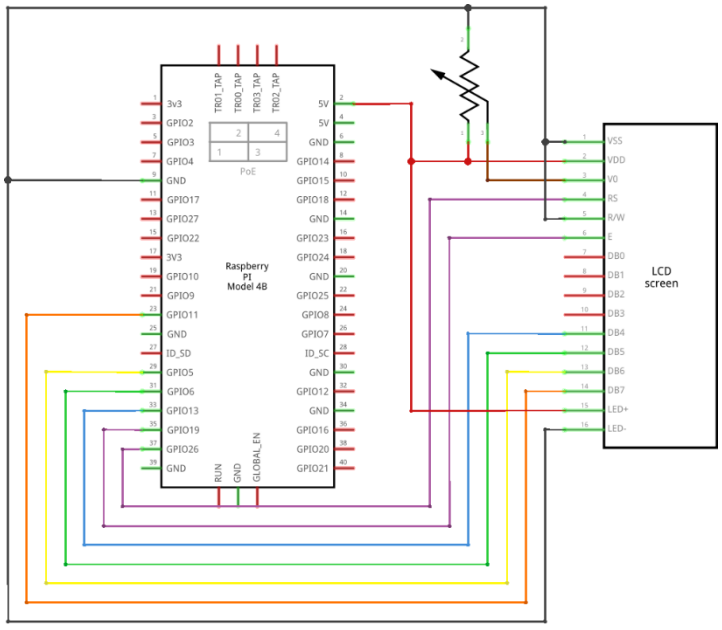


Figure 3.4.1: Schematic diagram of LCD (20 x 4)

Figure 3.4.1 Shows how to connect LCD to Raspberry pi.

3.4.2. Temperature Sensor

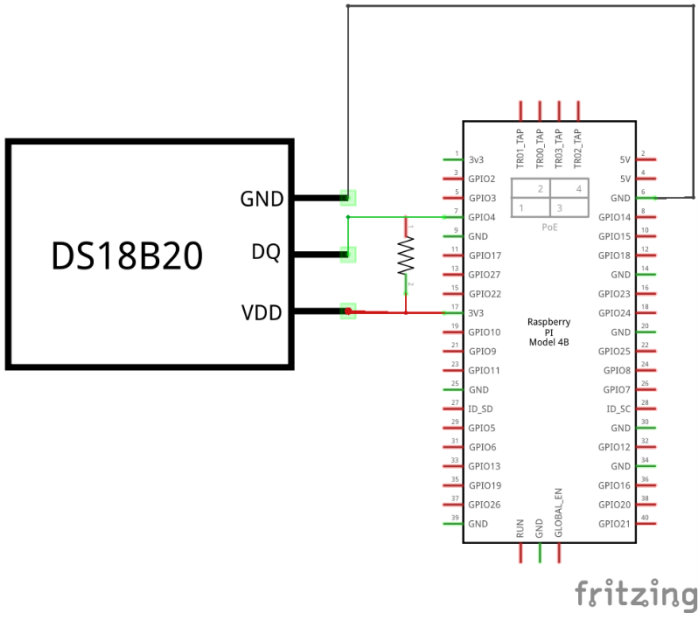


Figure 3.4.2: Schematic diagram of Temperature Sensor DS18B20

Figure 3.4.2 Shows how to connect Temperature Sensor to Raspberry pi.

3.4.3. PH sensor

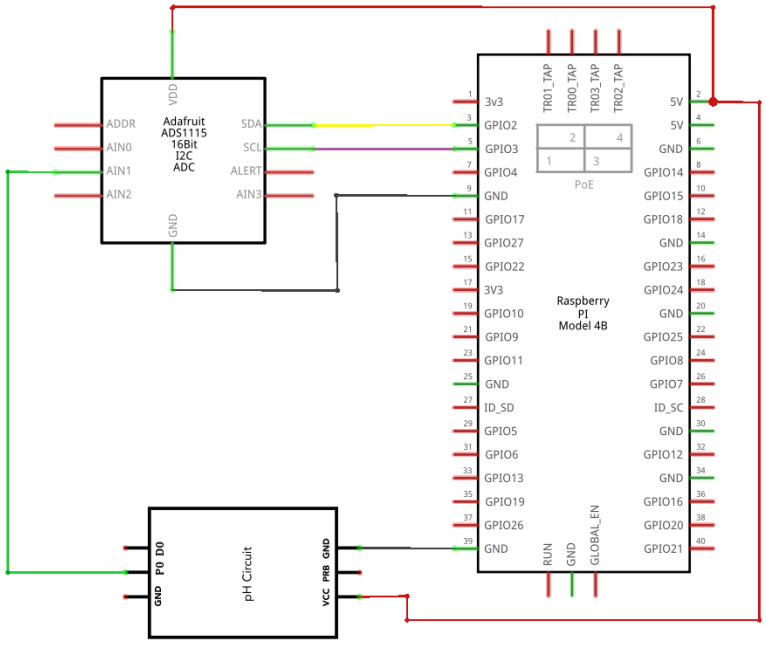


Figure 3.4.3: Schematic diagram of PH sensor

Figure 3.4.3 Shows how to connect PH sensor to Raspberry pi.

3.4.4. Turbidity Sensor

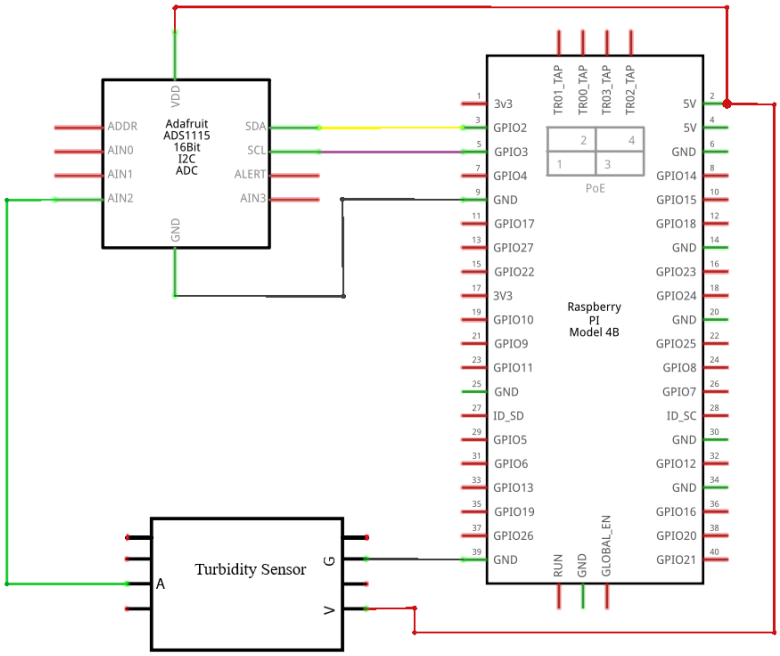


Figure 3.4.4: Schematic diagram of Turbidity Sensor

Figure 3.4.4 Shows how to connect Turbidity Sensor to Raspberry pi.

3.4.5. TDS Sensor

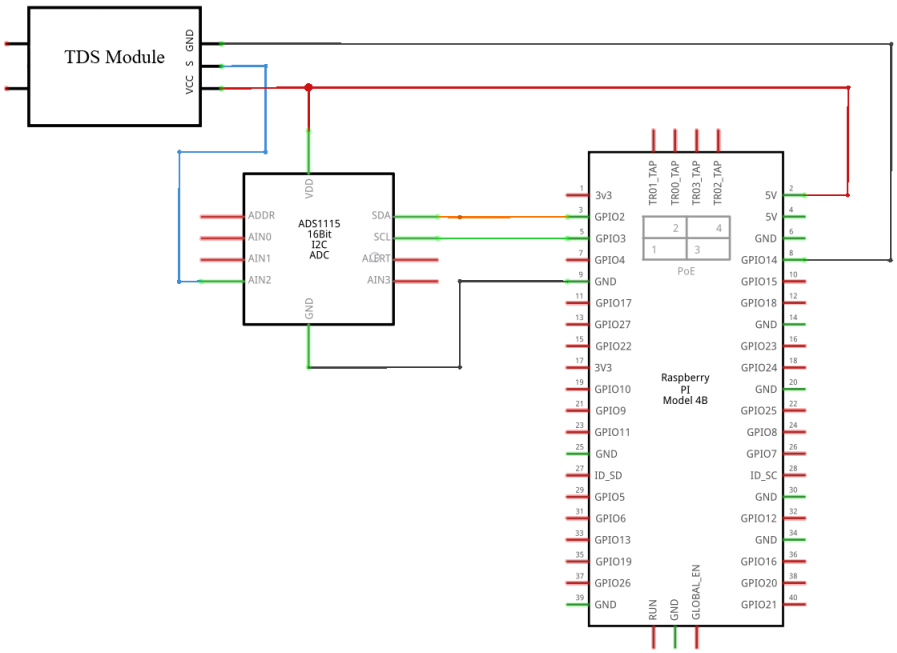


Figure 3.4.5: Schematic diagram of TDS Sensor

Figure 3.4.5 shows how to connect TDS Sensor to Raspberry pi.

3.4.6. GPS Module

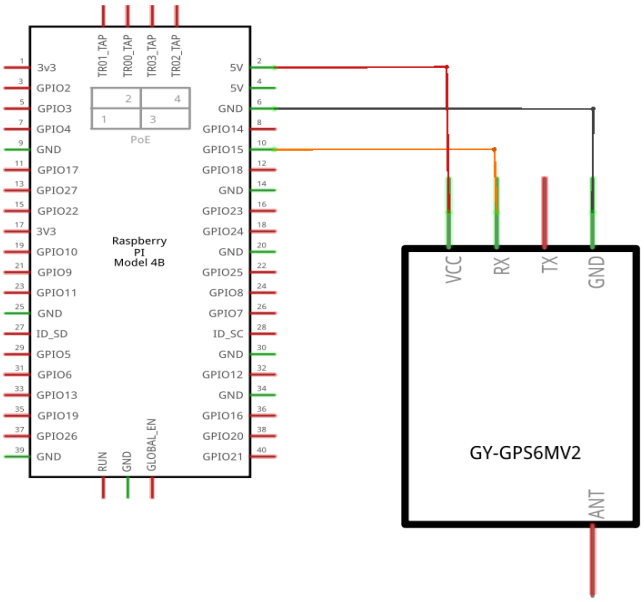


Figure 3.4.6: Schematic diagram of GPS Module NEO-6MV2

Figure 3.4.6 shows how to connect GPS Module to Raspberry pi.

3.5. Design Description

The whole system design as shown in figure 3.5.1, is fundamentally based on IOT which is a newly introduced concept in the world of development. There are basically two parts involved, the first is the hardware and the second is the software. The hardware part contains sensors that help measure real-time values and GPS to get the location of the test waters. The LCD shows the monitors output from the sensors, another one is raspberry pi to process the readings of the sensors, but it can't convert analog values from analog sensors to digital, so we use the analog to digital converter to do that, and it has a Wi-Fi inside those gives the communication between hardware and software. The application and web page are designed to see the output and monitor the water. When the system is started, a constant current is given to the system and raspberry pi. The water parameters are tested and its results are presented to the LCD screen with a notification if there is any error in the water.

The application and web page are provided with a connection that gives the exact value as in the LCD displays on the system. Thus, when the system is in any specific water body and WIFI is provided, we can observe its value in real time on our phones or browsers anywhere, anytime.

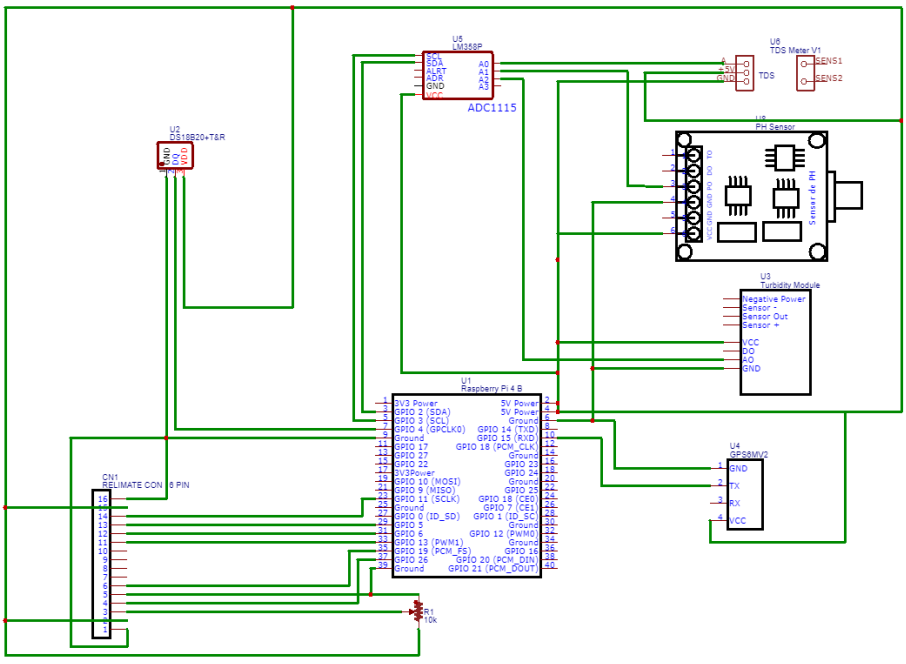


Figure 3.5.1: PCB Design of the system Hardware

Chapter 4: Software:

4.1. Overview

This chapter describes the software that are used in this project, Application and Webpage and all necessary information about the software developed to drive the system.

4.2. Software requirement specifications

4.2.1. Microcomputer Software

For the microcomputer software, there are needs; the following paragraphs describe these.

- Operating system

  Raspbian operating system is widely used with a raspberry pi as we mention in section 2.4.2, so we have decided to install it on the raspberry pi. It's easy to use and install libraries we need.

- Compiler

  The Hardware components codes were written in Python programming language according to its specifications which can be found in Python Libraries provided by Adafruit. The compiler we used is Thonny IDE version 3.3.10.

4.2.2. Server's Software

At the server side we need a program that receives data from WQMS through the internet, process it, and store it in the database according to these data. Also, the server accepts commands from user and send it to the BS, this server work as a middleware between webpage and application and hardware connection. The server Implemented using the Apache local host web server and webhost for hosting database for application.

Also, we need a database to store the received data from sensors and store it into the appropriate format in tables according to the arrival time of these data. the database is built using MySQL, which is responsible for reading, editing and deleting from the database.

4.2.3. Webpage Software

The webpage needs to deals with the information in the database and shown it to the user. To design the pages, we use CSS to define styles for the web page, html for the interface and the structure of the pages and we use php to communicate between the data and the webpage.

4.2.4. Application Software

We use a flutter for the application because is a great tool for implementing mobile programs, whether for Android or iOS, we also designed the API technology to link MySQL databases with the application.

When the user opens the application, he logs in, and if he does not have an account on the application, he makes an account from the sign up page and then logs in, each user has his own user ID so that the application can display the data of this user only, as each user has private data With it, after registration, the program directs the user to the home page, which contains data on water quality and shows the values of all parameters such as pH, purity, temperature, etc…. , And show the water quality with the information of the values coming from the data base, whether the quality is (good, medium, poor), there is also a side page or what is known as a drawer to move between the application pages, when the user opens the side menu, he will first find the home page, which is the page The automatic system on which the application opens, and then finds a page Total dissolved solids, and there is a report on it that is a graph to follow up on water measurements during the month. It also shows the highest value recorded by the sensors, the lowest value and the average value, and shows the measurement status if it is good, poor or average based on the accepted standards, as well as Then he finds the Temperature page, which has a graph as on the Total dissolved Solids page, and also the same reports are found in the rest of the pages (Ph, Turbidity)

There is also the location page, which is a page with a specific map on which the location of the sensors is specified, and the last button is the logout button.

4.3. Flowcharts

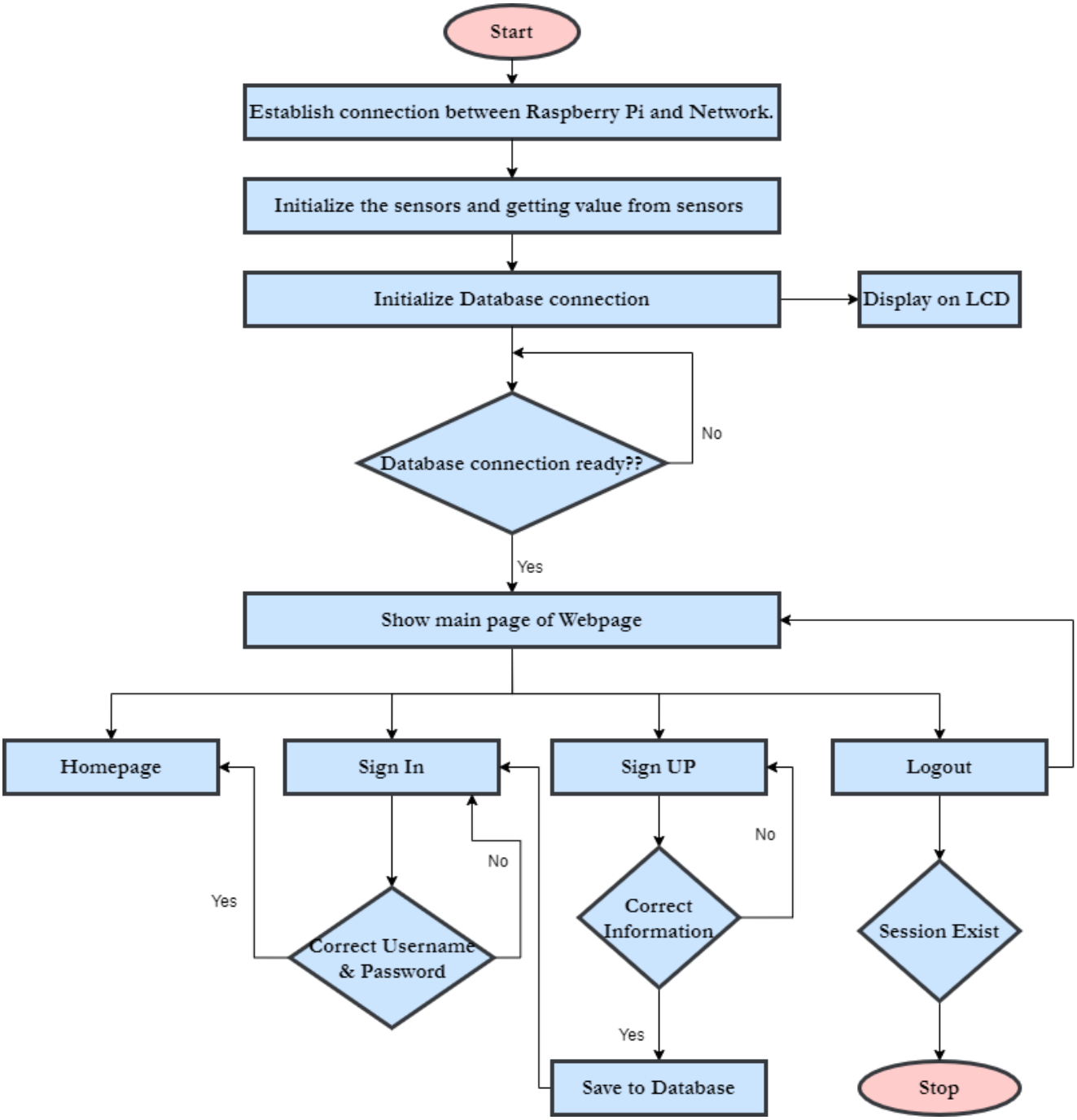
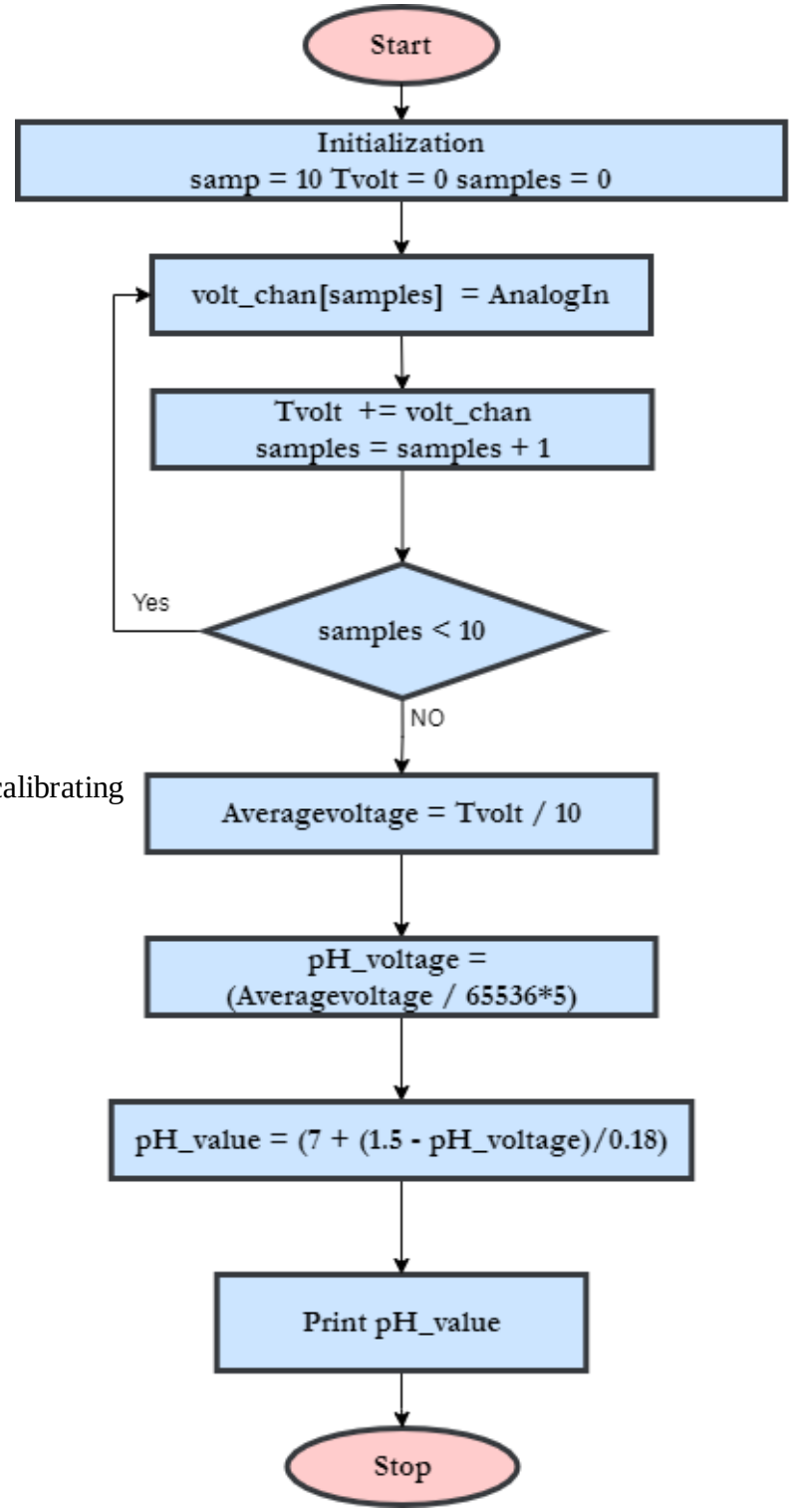


Figure 4.3.1: Webpage Flowchart



10: Number of Samples for calibrating

65536: 16 bits addressing

5: Input DC voltage

7: normal pH Value

1.5: losing voltage

Figure 4.3.2: pH Sensor Algorithm Flowchart

800: Number of Samples for calibrating

65536: 16 bits addressing

5: Input DC voltage

2.1: Half input read voltage

3000: Maximum NTU value

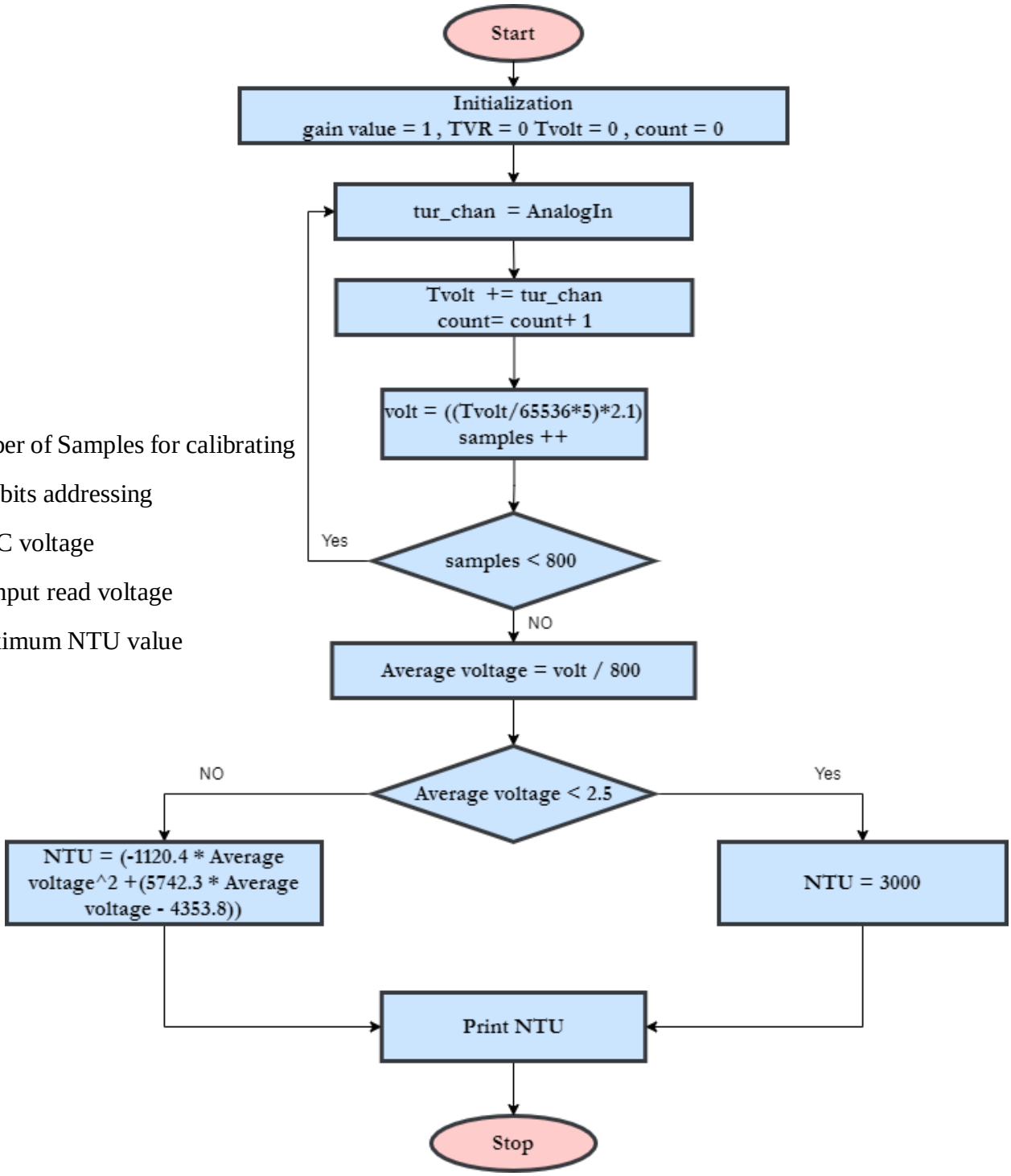


Figure 4.3.3: Turbidity Sensor Algorithm Flowchart

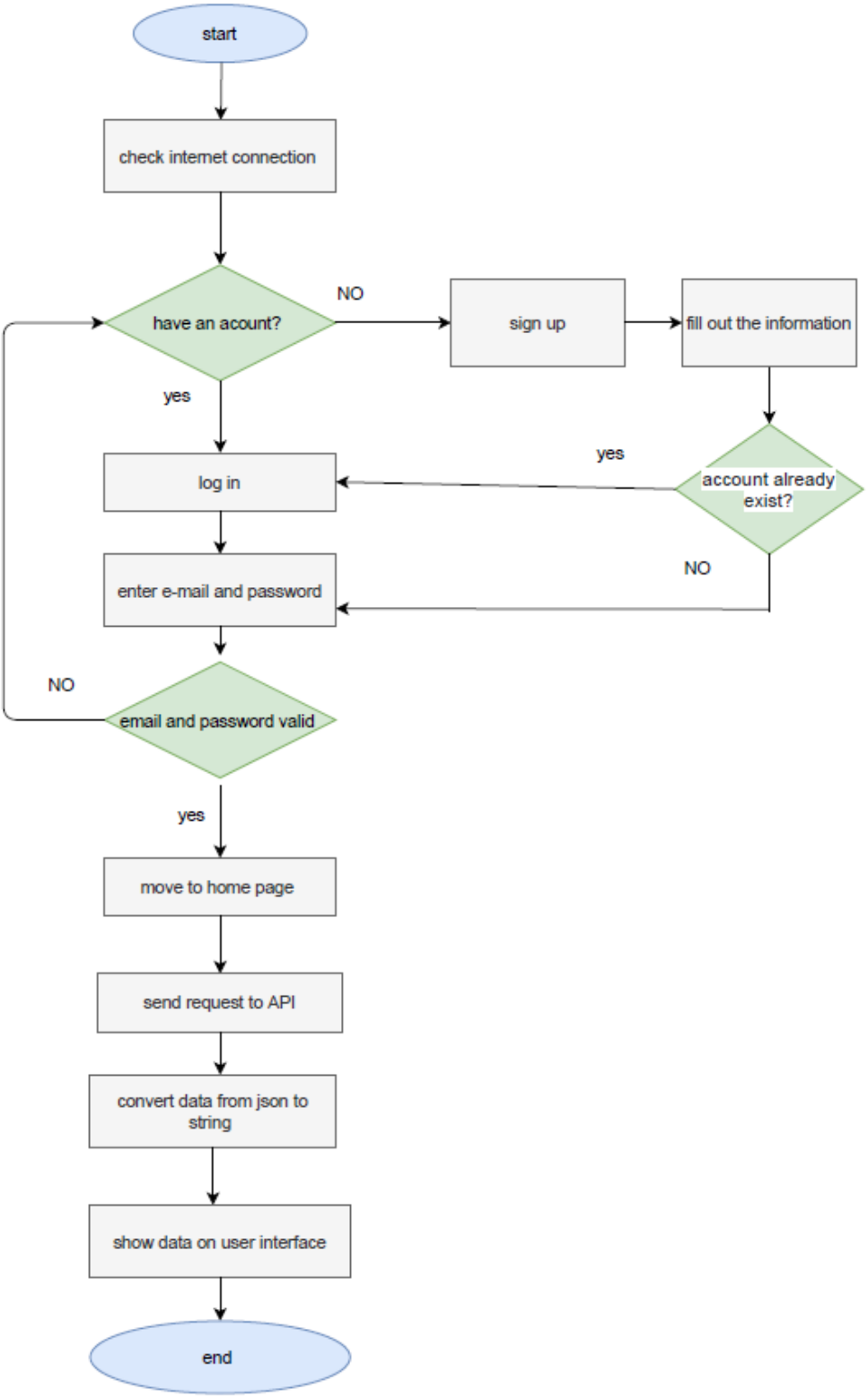


Figure 4.3.4: Application Flowchart

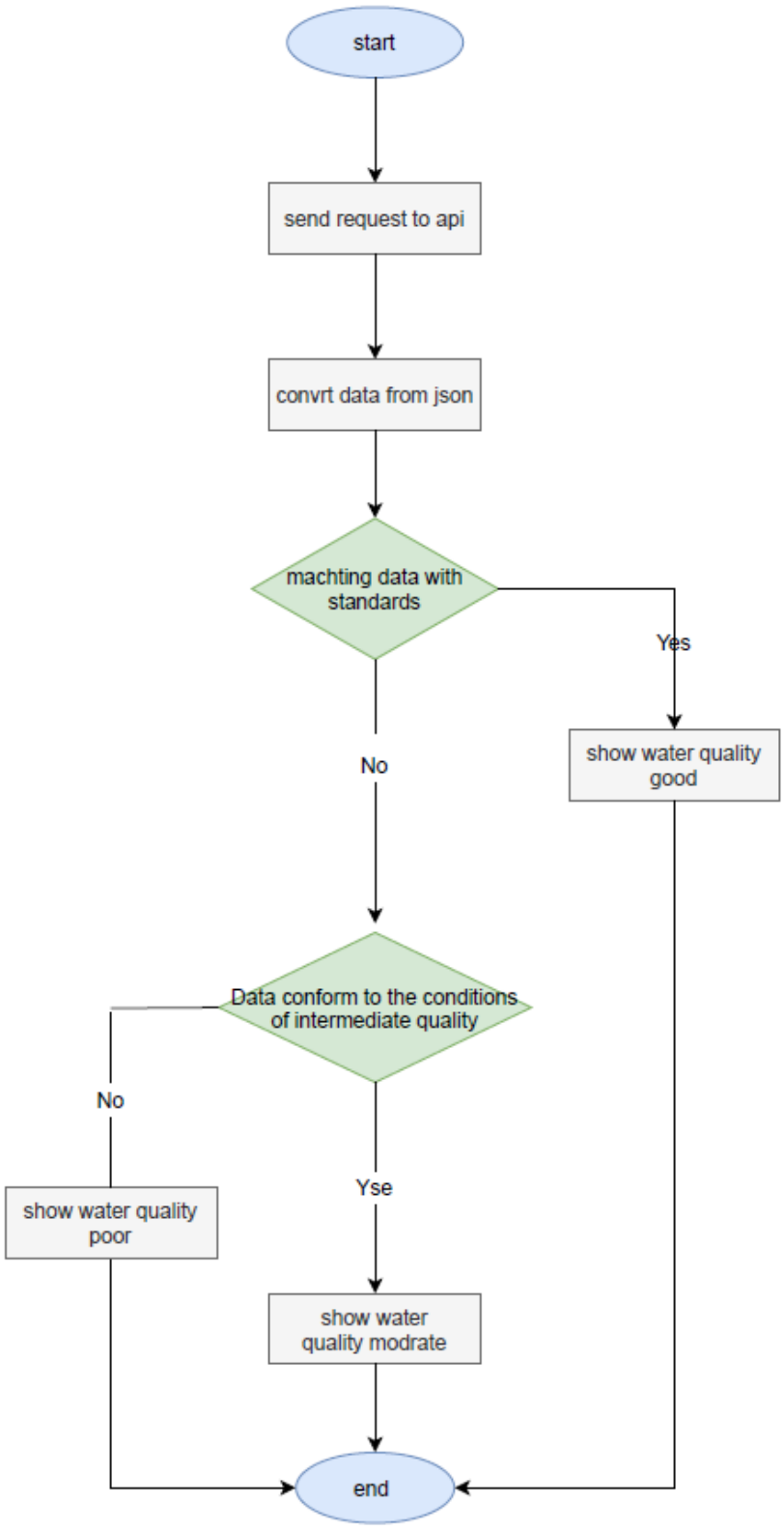


Figure 4.3.5: Water Quality Status Flowchart

Chapter 5: Validation and Discussion

5.1. Overview

This chapter describes the implementation, implementation issues, and implementation challenges. Also include description of the method used to validate the system, validation results, analysis and discussion about the results, and recommendations.

5.2. Hardware Testing

This section discusses independent testing of all hardware devices to ensure that the component achieves its specified function. This test method simplifies the detection of faults.

Turbidity Sensor Testing

Turbidity Sensor can work as analog or digital sensor. Here we use them as analog. So, as the pH sensor we need ADC to interface with Raspberry pi. As shown earlier in Figure 3.4.4, we Connect the VCC of the Turbidity Sensor with Raspberry pi 5V, GND to GND & Analog Output to A2 of the ADC. We wrote a test code on the Thonny IDE as shown in figure 5.2.1.1, to upload to the Raspberry Pi, and the result was that the Turbidity sensor gives reading. This way we have made sure that the Turbidity sensor is working properly.

```
#****** Turbidity Sensor ******
ads.gain = 1
totalVoltageReading = 0
tur_channel = AnalogIn(ads, ADS.P2)
Tvolt = tur_channel.value
for count in range(800):
    currentVoltageReading = float((Tvolt /65536* 5)*2.1)
    totalVoltageReading = totalVoltageReading + currentVoltageReading

averageVoltageReading = totalVoltageReading / 800
volt = round(averageVoltageReading,2)

if (volt < 2.5):
    ntu = 3000
    tur_value = ntu
else:
    ntu = round(float(-1120.4 * volt * volt + (5742.3 * volt - 4353.8)),2)
    tur_value = ntu

print("Voltage value is" ,volt ,"V" , ';', "Turbidity value is" ,tur_value ,"NTU")
```

Figure 5.2.1.1: Turbidity Sensor Test Code

Temperature Sensor Testing

The Temperature sensor can be powered with a 3V to 5.5V power supply and consumes only 1mA during active temperature conversions. As shown earlier in Figure 3.4.2, we connect VCC to 5V DC on the Raspberry pi, GND to ground of the Raspberry pi and data that goes to GPIO4 pin 7 on Raspberry pi. We wrote a test code on the Thonny IDE as shown in figure 5.2.1.2, to upload to the Raspberry Pi, and the result was that the Temperature Sensor determine the Temperature in Celsius and Fahrenheit. This way we have made sure that the Temperature Sensor is working properly.

```
#****** Temperature Sensor ******
def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
 lines = read_temp_raw()
 try:
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = round(float(temp_string) / 1000.0,1)
        temp_f = math.ceil(temp_c * 9.0 / 5.0 + 32.0)
        return temp_c
 except:
    temp_c = -9999
    temp_f = -1
    return temp_c

while 1:
    temperature = read_temp()
    if(temperature!=-9999):
       print("Temperature value = " ,temperature)
    time.sleep(1)
GPIO.cleanup()
```

Figure 5.2.1.2: Temperature Sensor Test Code

PH Sensor Testing

The PH sensor is analog and the Raspberry Pi has no analog inputs so we use ADC to interface the pH with Raspberry pi. However, As shown earlier in Figure 3.4.3 for connection, we Connect the GND pin of the pH circuit to the ground pin of RPi and VCC to 5V DC on the Raspberry pi and we Connect the Po (analog output) pin to A1 on the ADC. We wrote a test code on the Thonny IDE as shown in figure 5.2.1.3, to upload to the Raspberry Pi, and the result was that the PH sensor gives reading. This way we have made sure that the pH sensor is working properly.

```
#****** pH Sensor ******
    samp = 10
    Tvolt = 0
    for samples in range(10):
        ph_channel = AnalogIn(ads, ADS.P1)
        volt_chan = ph_channel.value
        Tvolt = Tvolt + volt_chan
    AverageVoltageRead = Tvolt / 10
    ph_voltage = (AverageVoltageRead/65536* 5)
    ph_value = round(7+((1.5-ph_voltage)/0.18) ,2) #calabration
    print("pH value is" , ph_value)
```

Figure 5.2.1.3: PH Sensor Test Code

TDS Sensor Testing

The connection of TDS Sensor with Raspberry pi is like pH and Turbidity sensors. We connect the VCC to Raspberry pi 5V & GND to GND. As shown earlier in Figure 3.4.5, we connect its Analog pin to A0 of the ADC. We wrote a test code on the Thonny IDE as shown in figure 5.2.1.4, to upload to the Raspberry Pi, and the result was that the TDS sensor gives reading. This way we have made sure that the TDS sensor is working properly.

```
#****** TDS Sensor ******
ads.gain = 2/3            # Gain value
tds_channel = AnalogIn(ads, ADS.P0)
tds_value = tds_channel.value
Voltage = (tds_value/65536* 5) # 16 bit  addrissing
tds = round(((133.42*Voltage*Voltage*Voltage- 255.86*Voltage*Voltage+ 857.39*Voltage)/2),2)
print("TDS value is" , tds,"ppm")
time.sleep(5)
```

Figure 5.2.1.4: TDS Sensor Test Code

GPS Module Testing

The NEO-6MV2 GPS module can easily interface with Raspberry pi port. However, As shown earlier in Figure 3.4.6, we connect the VCC pin to 5V DC on the Raspberry pi and the ground pin to a common ground of the Raspberry pi and The TX pin of the module connected to the GPIO15 that works as RX of the Raspberry pi port. We downloaded the necessary libraires for GPS on Raspberry Pi and we wrote a test code on the Thonny Python IDE as shown in figure 5.2.1.5, to upload to the Raspberry Pi, and the result was that the GPS determine the position of the Raspberry. This way we have made sure that the GPS is working properly.

```
#****** GPS Module ******
try:
    port = "/dev/ttyAMA0"
    ser=serial.Serial(port,9600,timeout = 0.5)
    dataout =pynmea2.NMEAStreamReader()
    newdata=ser.readline()
    if newdata[0:6] == "$GPRMC":
        newmsg=pynmea2.parse(newdata)
        lat=newmsg.latitude
        lng=newmsg.longitude
        requests.post(url, data ={'lat':lat,'lon':lng,'user_id':user_id})
except:
    print ("..... Loading")
```

Figure 5.2.1.5: GPS Test Code

LCD Display Testing

As shown earlier in Figure 3.4.1. In connection of LCD to the Raspberry pi in our system it will be enough to connect pins 1–6 and 11–16 of the LCD display to the Raspberry pi. Here the details:

LCD pin 1 has to be connected to the ground of Pi.

LCD pin 2 has to be connected to the 5V pin of Pi.

LCD pin 3 is used for the contrast, in this case is grounded to provide the maximum contrast.

LCD pin 4 is the RS (Register Select) and it will be connected to GPIO 26 of the Raspberry pi.

LCD pin 5 is the R/W (Read/Write state), This pin does not use a GPIO but it is grounded to ensure that the display is in the write mode, which allows the pi to write in the display.

LCD pin 6 is the E(Enable)and it enable the LCD to be able to execute instructions and it will be connected to the GPIO 19.

LCD pin 7–10 (D0, D1, D2, D3) will not be used

LCD pin 11–14 (D4, D5, D6, D7) will be used because we are in 4-bit mode that requires just 4 data lines. They will be connected respectively to GPIO 13, 06, 05, 11.

LCD pin 15 is for the LCD backlight (Anode) and it will be connected to the 5V pin for the full brightness.

LCD pin 16 is for the LCD backlight (Cathode) and it will be connected to the Ground.

We wrote the test code on the Thonny Python IDE to upload to the Raspberry Pi, and the result was that the text in the test file appeared on the screen, so we repeated the process on different scripts and sensor readings. This way we have made sure that the LCD screen is working properly.

```
#********* LCD Test *************
from RPLCD import CharLCD
from time import sleep
#Instantiate lcd and specify pins
lcd =CharLCD(cols=20,rows=4,pin_rs=26,pin_e=19,
             pins_data=[13,6,5,11],charmap='A02',
             numbering_mode=GPIO.BCM)

lcd.cursor_pos = (0,0)
lcd.write_string("WQMS")
lcd.cursor_pos = (1,0)
lcd.write_string("Water Quality Monotir")
GPIO.cleanup()
```

Figure 5.2.1.6: LCD Display Test Code

5.3. Software Testing

The system was fully checked and ensured how it worked and the results of the Testing were successful, and the following tables are a review of some of the tests that we have carried out.

5.3.1. Application

In this section, the communication between the server and the mobile application will be Testing.

#	Case	Expected Output	Obtained Output	Pass/Fail
1	The information entered is correct	Added successfully	Added successfully	Pass
2	The information entered is incorrect	Error message	Error message	Pass
3	User Already Exist	Error message	Error message	Pass
4	If the user left an essential field empty	Error message	Error message	Pass

Table 5.1: Application Sign Up Test

#	Case	Expected output	Obtained output	Pass/Fail
1	The information entered is correct	successfully	successfully	Pass
2	The information entered is incorrect	Error message	Error message	Pass
3	If the user left an essential field empty	Error message	Error message	Pass

Table 5.2: Application Login Test

#	Case	Expected Output	Obtained Output	Pass/Fail
1	API returns data	show data in user interface	show data successfully	Pass
2	API does not return data	show data and set all value = 0	All data are displayed with a value of 0	Pass
3	Values that match the criteria	Water quality is good	displayed Water quality is good	Pass

Table 5.3: Application Home Page Test

#	Case	Expected Output	Obtained Output	Pass/Fail
1	There is data	show data in user interface	show data successfully	Pass
2	API return graph data	Display data on a graph	displayed Display data on a graph	Pass

Table 5.4: Application PH, Temperature, Turbidity, Total Dissolved Solids Page Test

#	Case	Expected Output	Obtained Output	Pass/Fail
1	Connect to google maps API	show Map In the application	displayed map In the application	Pass
2	API return lat & long	show mark in map	mark displayed In map	Pass

Table 5.5: Application location Page Test

#	Case	Expected Output	Obtained Output	Pass/Fail
1	Internet connection	Log out successfully	Log out successfully	Pass

Table 5.6: Application Log Out Test

5.3.2. Webpage

In this section, the communication between the server and the Webpage will be Testing.

#	Case	Expected Output	Obtained Output	Pass/Fail
1	Correct Information	Added successfully	Added successfully	Pass
2	User Already Exist	Error message	Error message	Pass
3	If the user left an essential field empty	Error message	Error message	Pass

Table 5.7: Webpage Sign Up Test

#	Case	Expected output	Obtained output	Pass/Fail
1	Correct Information	Login successfully	Login successfully	Pass
2	Invalid Information	Error message	Error message	Pass
3	If the user left an essential field empty	Error message	Error message	Pass

Table 5.8: Webpage Login Test

#	Case	Expected Output	Obtained Output	Pass/Fail
1	User Name	Show in the Webpage	Shown in the Webpage	Pass
2	System Sensors Data	Show and updated in the Webpage	Error message	Pass
2	System Location	Show the exact location	System exact location	Pass
4	Notification	Alarm Notification	Alarm Notification	Pass

Table 5.9: Webpage Home Page Test

#	Case	Expected Output	Obtained Output	Pass/Fail
1	Correct Information	Added successfully	Added successfully	Pass
2	User Already Exist	Error message	Error message	Pass
3	If the user left an essential field empty	Error message	Error message	Pass

Table 5.10: Webpage Log Out Test

5.3.3. Server

In this section, the communication between the server and the Hardware will be Testing.

An HTTP request from Raspberry pi is sent into the system as a result of sensor readings, and when the server receives this request, it modifies the database by converting sensor reading values into sensor values (see Figure 5.3.3.1).

```php
<?php
    include('./dbconnect.php');
    if(isset($_POST['TDS'])){
        $TDS_VALUE = $_POST["TDS"];
        $USER_ID = $_POST["user_id"];
        $query = "UPDATE sensors_readding SET tds_sensor = $TDS_VALUE WHERE user_id = $USER_ID";
        $result = mysqli_query($conn,$query);
    }
    if(isset($_POST['TUR'])){
        $TDS_VALUE = $_POST["TUR"];
        $USER_ID = $_POST["user_id"];
        $query = "UPDATE sensors_readding SET turbidity_sensor = $TDS_VALUE WHERE user_id = $USER_ID";
        $result = mysqli_query($conn,$query);
    }

    if(isset($_POST['PH'])){
        $TDS_VALUE = $_POST["PH"];
        $USER_ID = $_POST["user_id"];
        $query = "UPDATE sensors_readding SET ph_sensor = $TDS_VALUE WHERE user_id = $USER_ID";
        $result = mysqli_query($conn,$query);
    }
    if(isset($_POST['TEMP'])){
        $TDS_VALUE = $_POST["TEMP"];
        $USER_ID = $_POST["user_id"];
        $query = "UPDATE sensors_readding SET temperature_sensor = $TDS_VALUE WHERE user_id = $USER_ID";
        $result = mysqli_query($conn,$query);
    }
 if(isset($_POST['lat'])){
        $LAT_VALUE = $_POST["lat"];
        $LON_VALUE = $_POST["lon"];
        $USER_ID = $_POST["user_id"];
        $query = "UPDATE sensors_readding SET lat = $LAT_VALUE,lon=$LON_VALUE  WHERE user_id = $USER_ID";
        $result = mysqli_query($conn,$query);
    }
?>
```

Figure 5.3.3.1: Server to Hardware Data Request

5.4. Over All System Implementation and Validation

After ensuring that all the parts (Mobile Application, Webpage and Hardware components) are working well, we started implementation and integrating the parts with each other to make the system ready to work.

5.2.1. Hardware Implementation and Validation

In this section the hardware is built and connected to each other according to the design described in Section 3.5 Design Description in chapter 3.

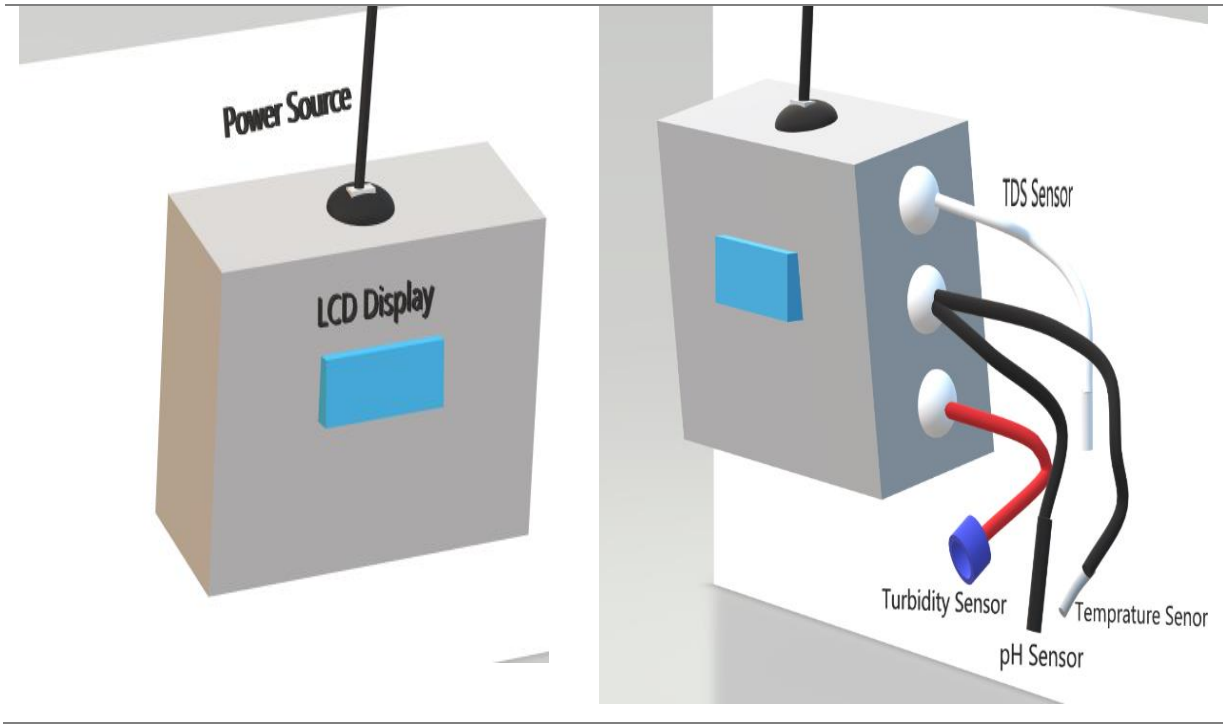
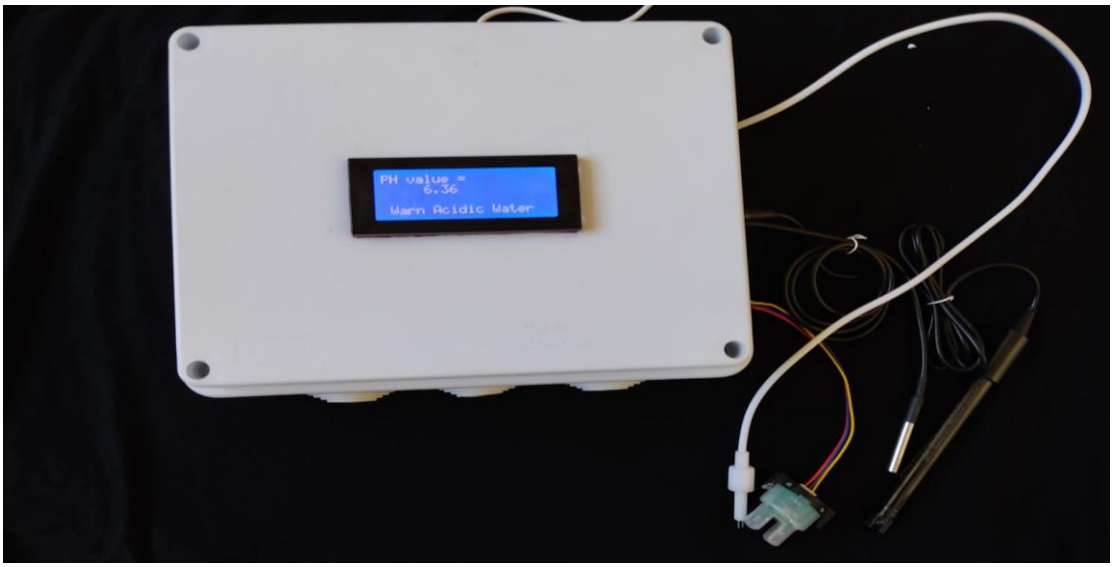


Figure 5.2.1.1: 3D design for System Hardware



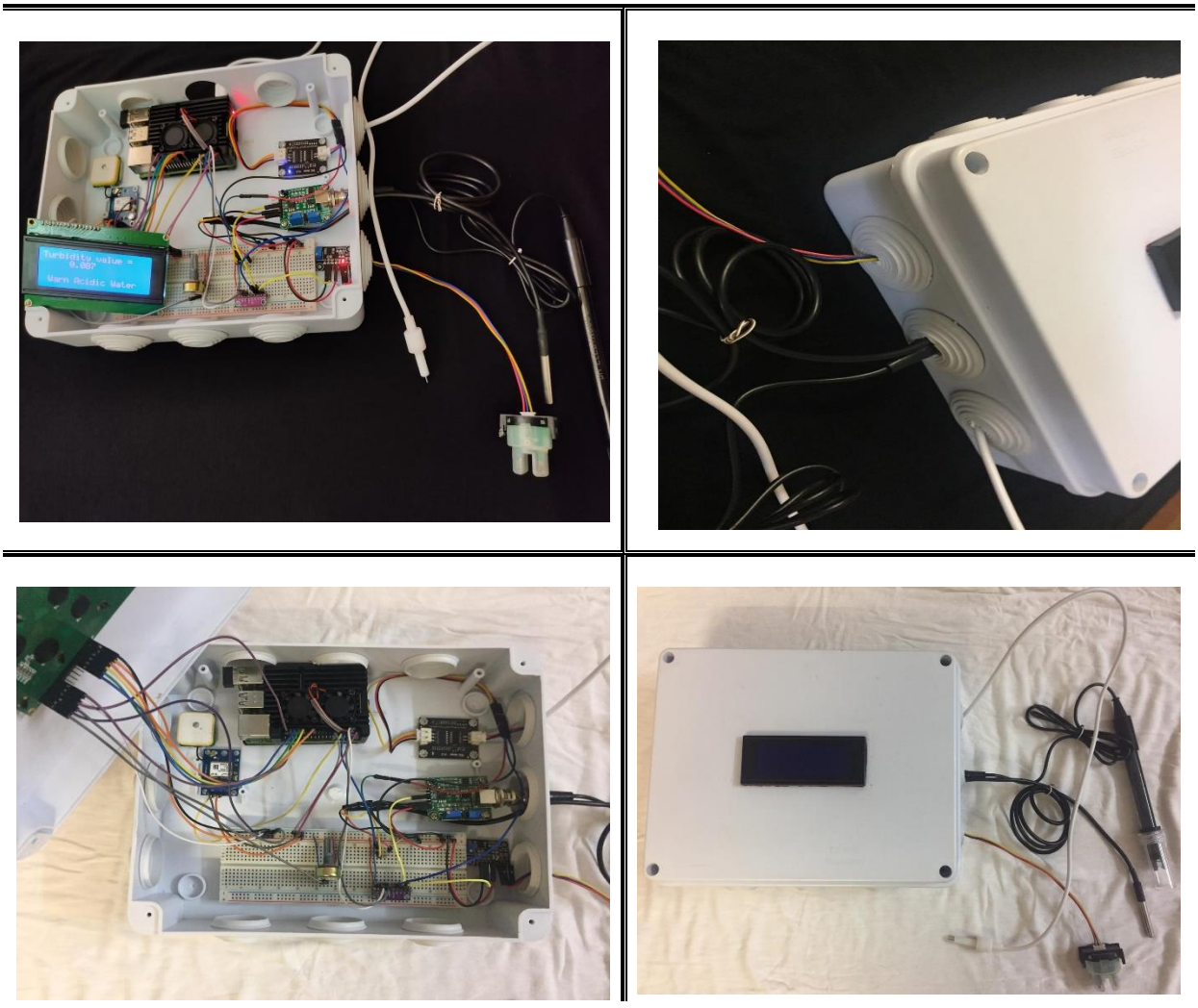


Figure 5.2.1.2: Implementation of System Hardware

5.2.2. Software Implementation and Validation

we implemented the program using Cross Platform and the flutter programming language, as flutter is a great tool for implementing mobile programs, whether for Android or iOS, we also designed the API technology to link MySQL databases with the application.

When the user opens the application, he logs in, and if he does not have an account on the application, he makes an account from the sign up page and then logs in, each user has his own user ID so that the application can display the data of this user only, as each user has private data With it, after registration, the program directs the user to the home page, which contains data on water quality and shows the values of all parameters such as pH, purity, temperature, etc..... , And show the water quality with the information of the values coming from the data base, whether the quality is (good, medium, poor), there is also a side page or what is known as a drawer to move between the application pages, when the user opens the side menu, he will first find the home page, which is the page The automatic system on which the application opens, and then finds a page Total dissolved solids, and there is a report on it that is a graph to follow up on water measurements during the month. It also shows the highest value recorded by the sensors, the lowest value and the average value, and shows the measurement status if it is good, poor or average based on the accepted standards, as well as Then he finds the Temperature page, which has a graph as on the Total dissolved Solids page, and also the same reports are found in the rest of the pages (Ph, Turbidity).

There is also the location page, which is a page with a specific map on which the location of the sensors is specified, and the last button is the logout button.

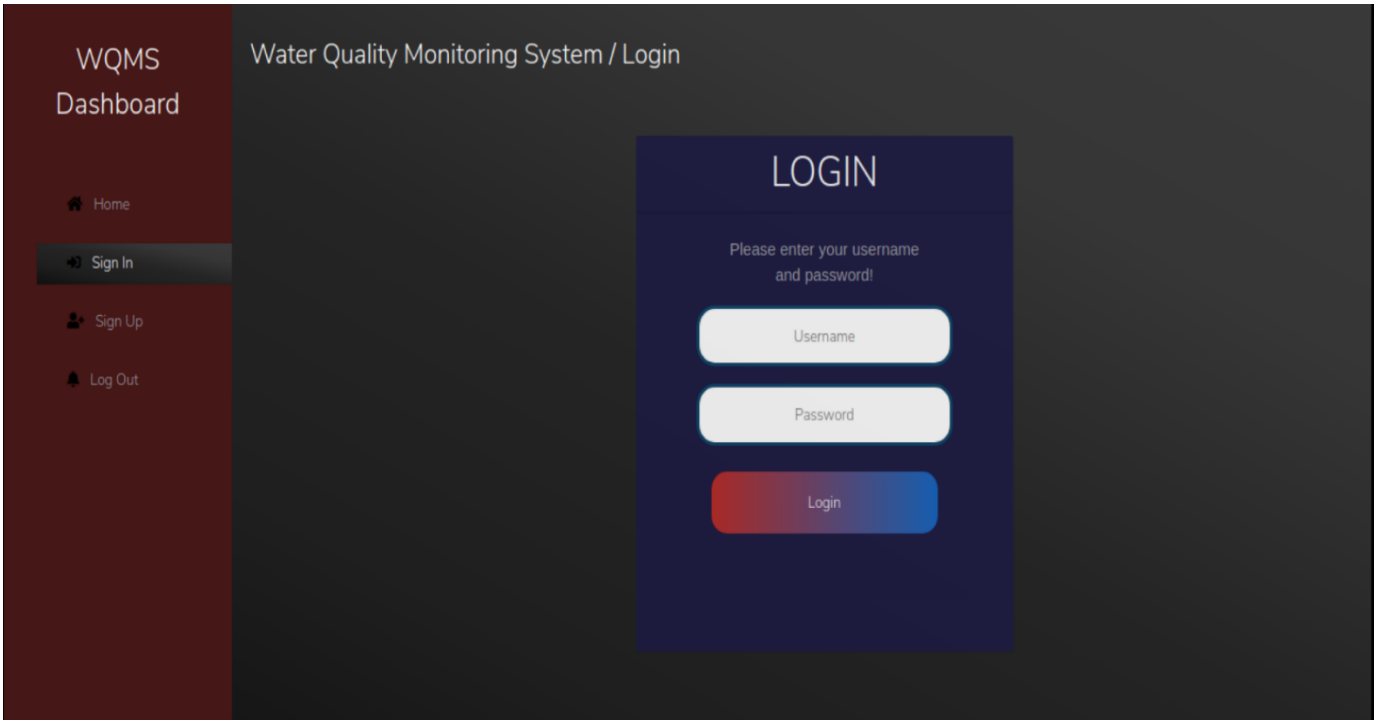


Figure 5.2.2.1: Webpage Sign Up

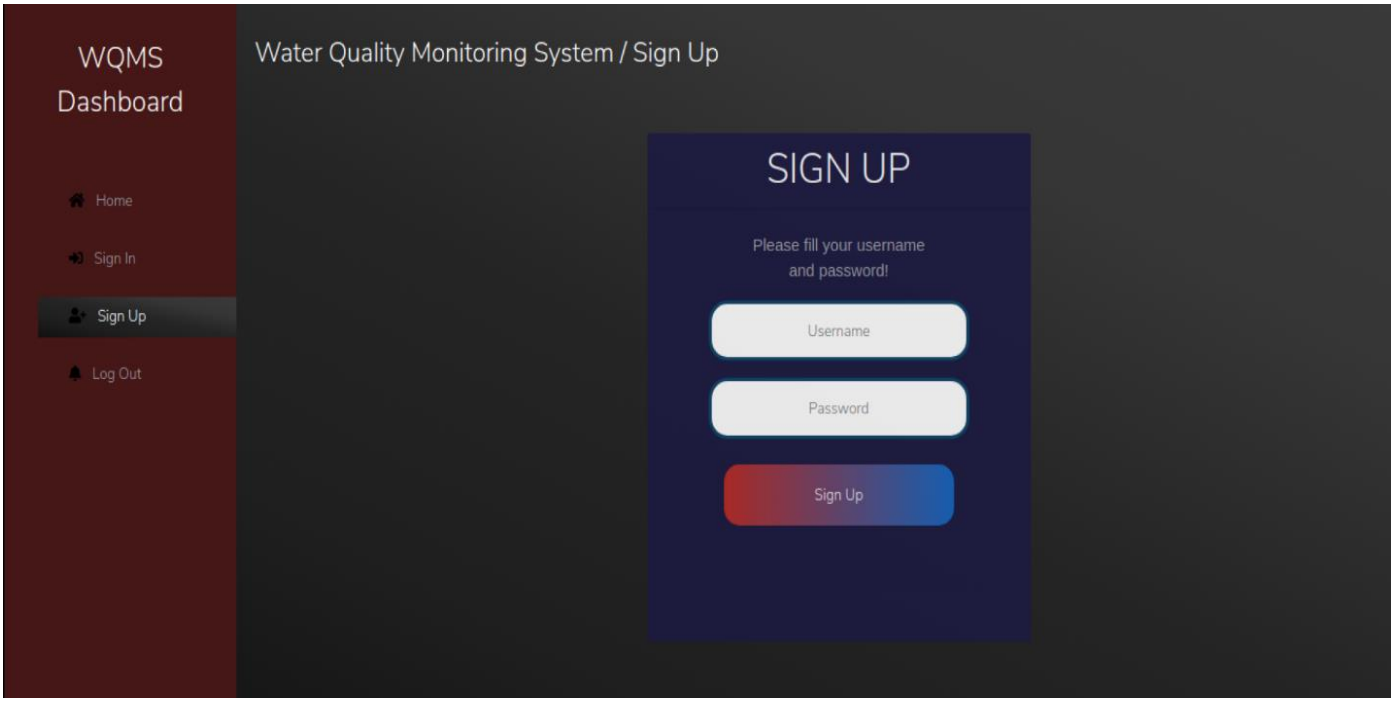


Figure 5.2.2.2: Webpage Login

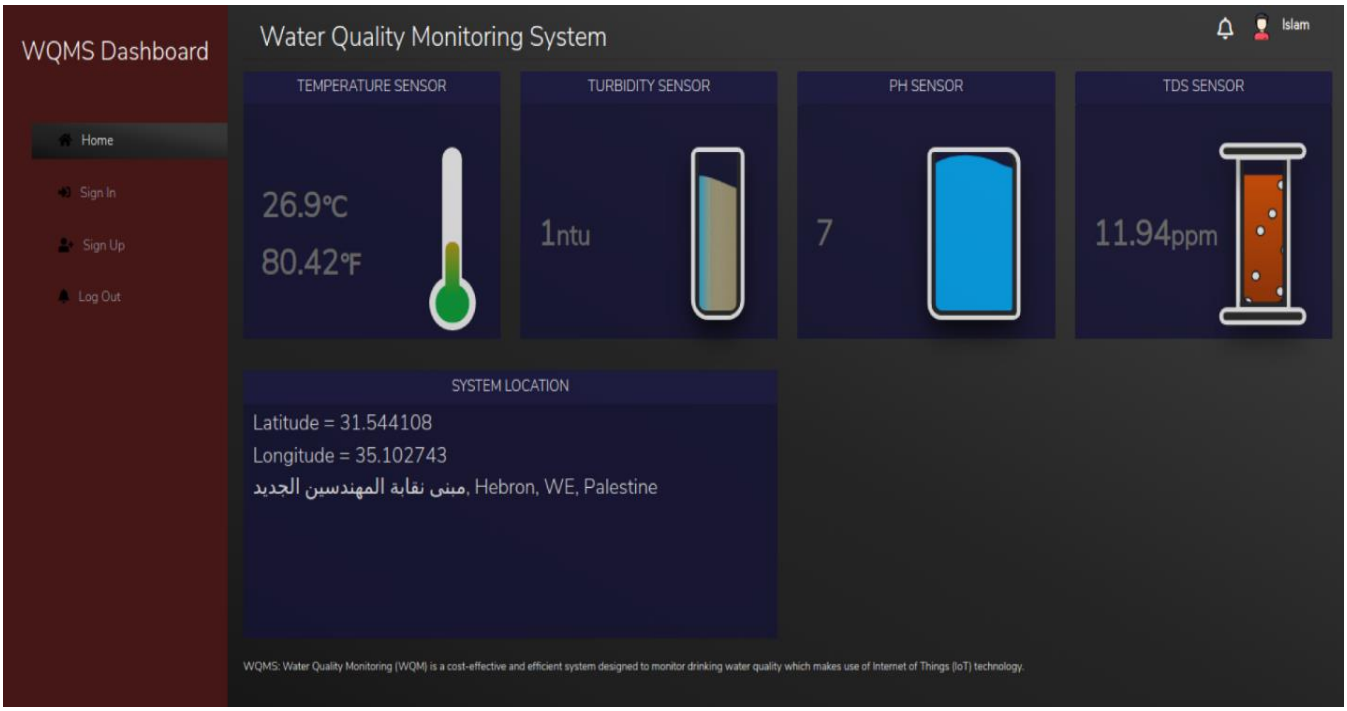


Figure 5.2.2.3: Webpage Homepage

Figure 5.2.2.4: Webpage Notification

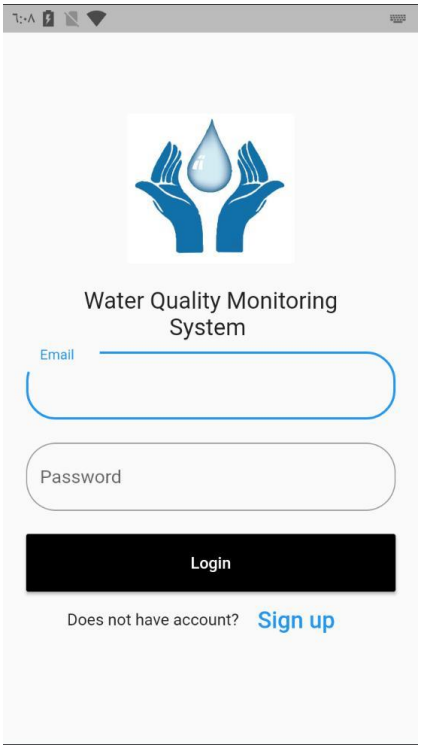


Figure 5.2.2.5: Application Login Screen

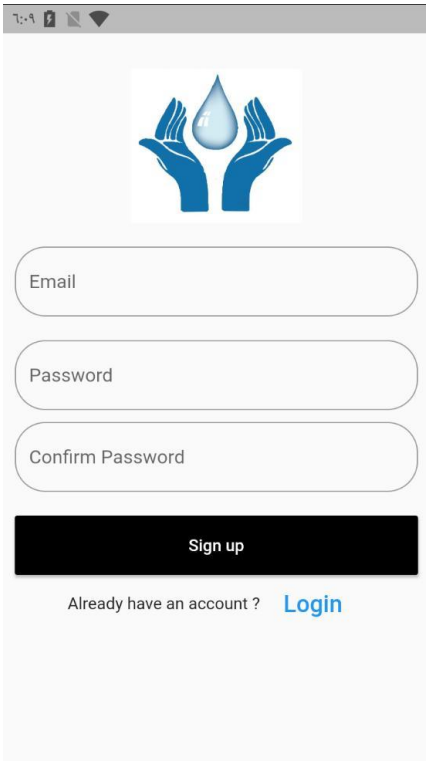


Figure 5.2.2.6: Application Sign Up Screen

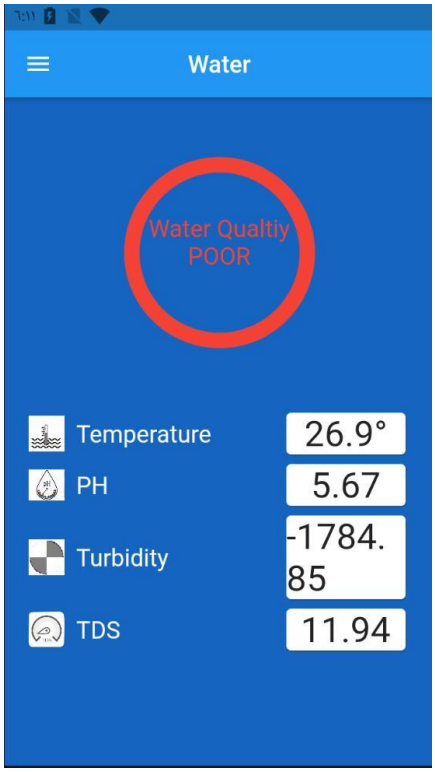


Figure 5.2.2.7: Application Home Screen

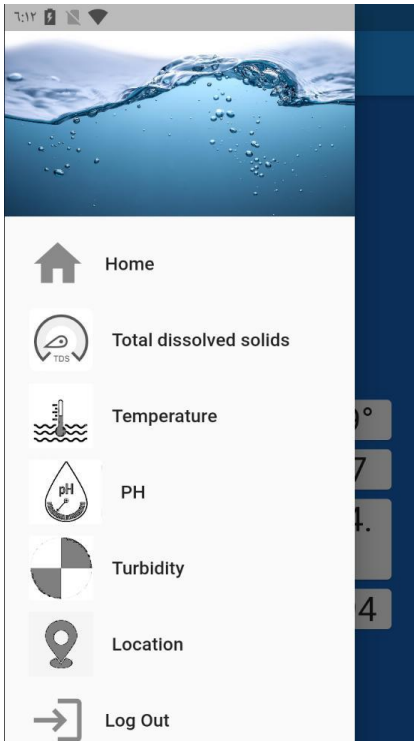


Figure 5.2.2.8: Application Drawer Screen

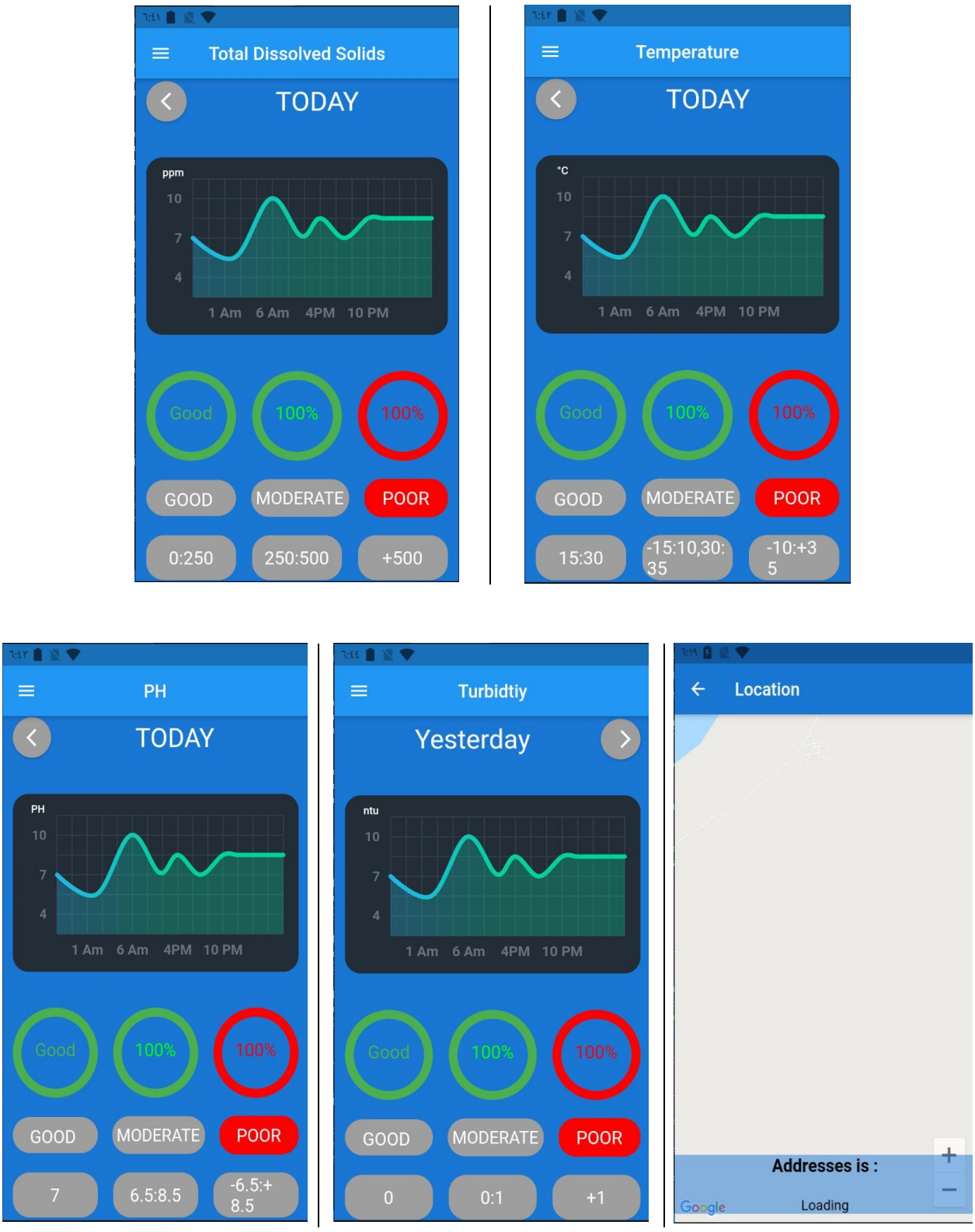


Figure 5.2.2.4: Application Parameters Screens

5.5. Implementation Challenges and Problems

Challenges and issues are normal, especially when one develops their first project, but solving these problems is a success. Below are the main challenges and issues that were raised during the development of the system.

- Some project components are not available at the beginning of the semester, and the procurement process for components took a lot of time.
- Some devices were damaged due to wrong connections, ADC and temperature sensor were damaged so new devices were purchased.
- When analog sensors are connected to the circuit during setup, we don't get values from them, and Raspberry Pi doesn't have analog inputs, so we're thinking of several solutions and we spent a lot of time solving this problem. The solution was to use an analog-to-digital converter (ADC).
- After connecting the pH and turbidity sensors to the ADC with Raspberry Pi, they give unacceptable values (-1000ntu, 19pH), and we spend a lot of time solving this problem, the solution was to adjust the pH and turbidity compensation voltage, the idea behind this step is that the voltage The analog oscillates between positive and negative values. We only want the positive values of the ADC conversion.
- There is a problem with getting System Location, that the API use for it is not free and it cost 200$ each month for usage, the problem solved by use a free API show the location as string location instead of a map.
- The application is unable to access the database. The problem solved by uploaded the database to a host server other than the local server so that the application can access this data.
- We have designed a PCB circuit as shown in figure 3.5.1, to make the connection of the parts appear tidier and more stable, but it will take a long period of time to be ready. We had spoken to the Dr. Alaa AL Tamimi at the university and he told us that it would be ready in the middle of the summer semester, that is, a month and a half after the date of the discussion, so we were satisfied with connecting the system using Breadboard for now.

5.6. Final Result

At the end of this project, we are happy to say that we have achieved all of its goals that were in our plan in advance, as now we have many sensors, each with a specific function, where it is located to measure temperature, pH, turbidity and total dissolved solids , also the LCD screens connected to the Raspberry Pi , the Raspberry Pi able to connect to the Internet to conduct a test and show the result on the LCD screen and show this data on the web page that we designed and also show it on the phone application that we designed, which deals with its accuracy for the user as it shows him the quality and efficiency of the water and the ability to easily know whether it is usable or not .

Chapter 6: Conclusion & Future work:

6.1. Overview

This chapter include summary of the final result and the future work of the project.

6.2. Summary

Nature and society, including residents, factories and companies, face great obstacles and challenges in ensuring the quality of the water we use in our lives. So, in this project we created an idea for an integrated IoT system for water quality monitoring.

This system is designed for two different usage scenarios: the on-demand water quality checker and as a standalone IoT device. In the first scenario, one could use the device to test a specific water sample. The device will check certain parameters, give values about quality parameters. In the second scenario, the device is installed in a shifting location, and it monitors the water quality over a longer period of time. The focus in designing the device's system was that it should operate in the context of factories and companies that are most in need of continuous water monitoring and are most vulnerable to pollution. With this system, water pollution can be easily detected, which helps in controlling it. In addition, the system enables the sensors to provide data online to consumers and prevents all the waste of time and money that can result from lab tests or delays and lack of knowledge of the unfit for use of water.

6.3. Futures Works

In the future work of the project, we plan to further develop the system with the addition of a few sensors, and improve it by integrating algorithms to detect anomalies in water quality. And to be able to connect it with other operating systems such as IOS, and expand it to be a product for sale in IOT e-market.

References:

[1] Water Quality Unit at Palestine Polytechnic University

[2] Calum McClelland. “What Is IoT? – A Simple Explanation of the Internet of Things.” IoT For All, November 16, 2020. Available: www.iotforall.com/what-is-iot-simple-explanation

[3] M. Spinola. (2013). The Five Characteristics of Cloud Manufacturing Things, eBook.

[4] Nikhil Kedia, Water Quality Monitoring for Rural Areas- A Sensor Cloud Based Economical Project. 978-1-4673-6809-4/15/$31.00 ©2015 IEEE.

[5] (SECON), 978-1-4673-1905-8/12/$31.00 ©2012 IEEE

[6] Arduino, URL: https://www.arduino.cc/en/Main/OldSoftwareReleases, [Online; accessed 20-November 2020].

[7] Raspberry Pi, URL: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/?resellerType=home, [Online; accessed 16-November 2020].

[8] Vishay, URL: https://www.vishay.com/docs/37314/lcd020n004l.pdf, [Online; accessed 18-November 2020].

[9] Seeedstudio, URL: https://media.digikey.com/pdf/Data%20Sheets/Seeed%20Technology/101020753_Web.pdf, [Online; accessed 17-November 2020].

[10] DS18B20 Datasheet, URL: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0198_Web.pdf, [Online; accessed 17-November 2020].

[11] Open circuit /Product/Gravity-Analog-Turbidity-Sensor [Online; accessed 17-November 2020].

[12] Okystar, URL: https://www.okystar.com/product-item/liquid-ph-value-detection-detect-sensor-module-monitoring-control-oky3483, [Online; accessed 17-November 2020].

[13] Components101, URL: https://components101.com/modules/neo-6mv2-gps-module , [Online; accessed 17-November 2020].

[14] Learn.Adafruit, URL: https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/mcp3008, 2021, [Online; accessed 11-march, 2021].

[15] Bristolwatch, URL: https://www.bristolwatch.com/rpi/ads1115.html, 2021, [Online; accessed 11-march, 2021].

[16] Raspbian, URL: https://www.raspberrypi.org/documentation/raspbian, 2021, [Online; accessed 11-january, 2021].