Palestine Polytechnic University

Collage of Information Technology and Computer Engineering

Department of Computer System Engineering

Smart IOT Based Water Distribution Control System

Team Members

Shaymaa Awlad Mohammed

Areej Al-Battat

Marah Taha

Supervisor

Dr. Amal Al-Dweik

May, 2022

Acknowledgement

We would like to thank all people who helped us, and those who had direct or indirect contributions to our project. Special thanks to our parents who always encourage, support, and care about us. Thanks to Palestine Polytechnic University and its staff for all their teaching and guidance efforts. We are also grateful to our friends in the Computer Systems Engineering program, College of Information Technology and Computer Systems Engineering at Palestine Polytechnic University. Our deepest gratitude goes to our supervisor Dr. Amal Al-Dweik, for the enlightening guidance, support, encouragement, and ultimate patience throughout the entire period of the project.

Abstract

People suffer from several problems in the water distribution process, such as misdistribution of water to neighborhoods, leakage of water, and insufficient water supply for their day's needs. The need to have a fair distribution of clean and enough water to all citizens motivates us to build an automated system that performs all these operations where the system controls the distribution of water, collects data and saves it in the database, then analyzes it and behaves accordingly.

Our project seeks to provide a system that measures the amount of water in the main tanks, amount of water flow, and amount of obtained water. The system controls the process of exiting water in the main tanks and the entry of water into the slave tanks. It, also, deals with cases such as water leakage. Therefore, this system provides the municipality with all information it may need through a mobile application on time and all the time. Finally, the system is expected to enable citizens to add their feedback for water distribution to be sent to the municipality for study. The system succeeded to achieve its objectives such as: measuring the amount of water in tanks and the rate of water in tanks, sending notifications to the authorities, and storing information and analyzing data.

List of Tables

List of Figures

Chapter one

Introduction

1.1 Overview

This chapter introduces the system motivation and its importance. It presents the system requirements and anticipates possible results for existing solutions. A simplified system distribution is introduced at the end of the chapter.

1.2 Motivation and importance

Nowadays, there are many problems in distributing water in a fair manner by the authorities. There is a need for a system that distributes water to neighborhoods to ensure the fair distribution of pure and clean water to the largest possible number of residents and in equal quantities. The system will be developed to address this motivation.

1.3 System aims and objectives

The main aim of the system is to build a system that mimic the municipality system for fair water distribution. To achieve this aim, a set of objectives should be achieved, as:

1. Measure the amount of water in the main tanks.
2. The distribution schedule is according to the amount of water in the main tanks.
3. Check the rate of water consumption periodically.
4. Send notifications to the authority.
5. Store information and analyze data and behave accordingly.
6. Re-compute the amount of water.

1.4 Problem statement

1.4.1 Problem Analysis

People suffer from several problems in water distribution, such as poor distribution of water to neighborhoods, leakage of water, insufficient water supply for their day's needs, and so there is a need to have a fair distribution of clean and enough water to all citizens.

1.4.2 Problem Solution

We need a system that performs several operations to help us solve these problems; operations: monitor and control water distribution, record the system status and update values frequently, inform the Municipal about the current status of the water availability, and suggests solutions to the authority accordingly after analyzing the collected data.

1.4.3 Problem Purpose

The problem purpose is to obtain a fair water distribution according to a pre-prepared and dynamic schedule and detect water leaks (if any).

1.5 Short description of the system

The proposed system controls the distribution of water to all neighborhoods during sufficient periods, by knowing the status of water in the main water tank and update values frequently. The system will be implemented using a number of sensors that will measure the amount of water in the main tanks. There are Valves at the entrance to each tank connected to the main carrier. The sensors and Valves are interfaced to a microcontroller and mobile application. As shown in figure 1.1.

The system will store all values and data in the database using the MQTT protocol. The stored data is analyzed and the system will use it to update the water schedule and periods needed to transfer water to the designated slave tanks, to ensure the fair distribution of water for all participants. Stored information can be: amount of water in main tanks, water period of flow, and amount of obtained water.

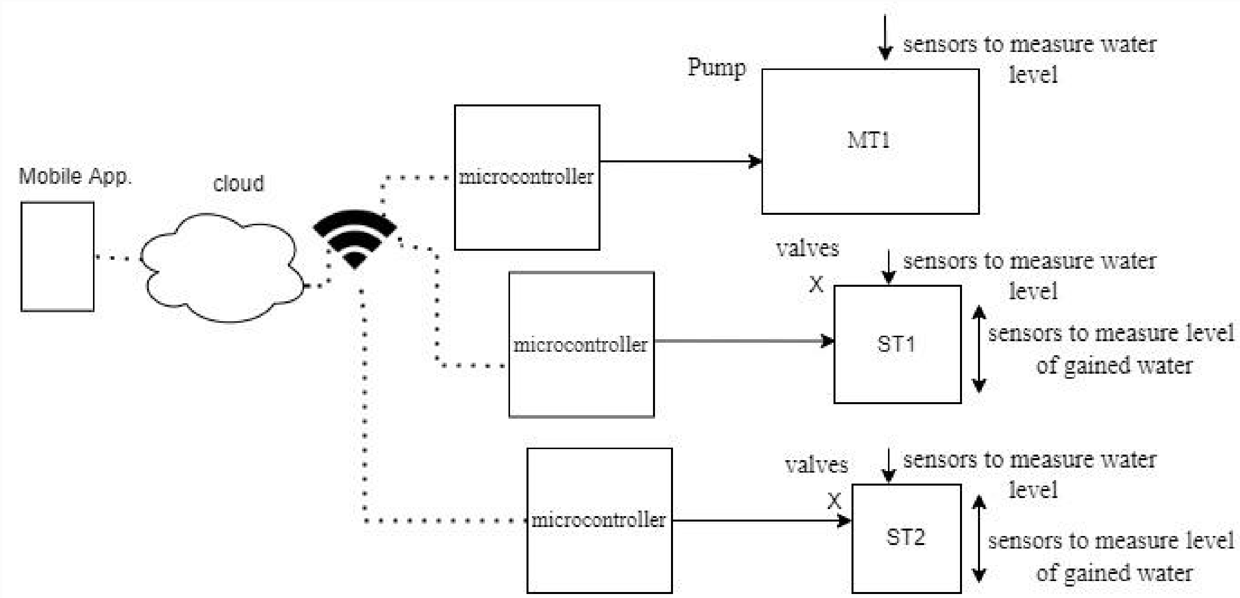


Figure 1.1: Context diagram of system

1.6 List of requirements

The system requirements can be summarized as:

1.6.1 Functional requirements

The system should be able to

- Collect data such as the amount of water, amount of water flow, and amount of obtained water.
- Turn off valves when water leaks.
- Close Valves when one of the main tanks is empty of water.
- Update the status of the tank periodically.

- Analyze obtained data and suggest solutions according to the different situations.

1.6.2 Non-Functional requirements

- Accuracy: Ensuring that the system has high accuracy in detecting, analyzing, and modifying readings.

- Usability: The extent of using the system and the application should be easy and understandable for users, depending on the components of the system and the technologies used in it.

- Scalability: The ability of the system to deal with the increasing number of main or subsidiary tanks without reducing the overall quality of the service provided to any neighborhood.

- Reliability: The system is intended function to do all it's under stated conditions without failure for a given period of time.

- Maintainability: All components are easily serviceable and replaceable.

1.7 Expected results

We expect to build an integrated system with the following specifications:

1. The system must be able to check the amount of obtained water, amount of water in the main tanks.

2. The system must store all values and data in the database, analyze the data to update the water schedule and periods needed.

3. The system must send all data related to the tank to the mobile application.

1.8 Overview of the rest of the report

The rest of the report is organized as follows: Chapter 2 presents a theoretical background of the project. Chapter 3 introduces the system design. The conclusion is presented in chapter 4.

Chapter two

Background

2.1 Overview

In this chapter, we will review the theoretical background and present the previous works and techniques that others used in their projects. In the next sections, the system hardware and software components, options will be discussed. Finally, the design constraints of the system will be mentioned.

2.2 Theoretical background

Billions of people around the world lack adequate access to one of the essential elements of life: clean water. Although, Governments and aid groups have helped many living in water-stressed regions gain access in recent years, the problem is projected to get worse with the harmful effects of global warming and population growth [8].

Palestine is experiencing a severe water crisis caused mainly by the lack of control over the Palestinian water resources. At present, the average per capita water consumption by the Palestinian population is approximately 55 l/c/d, or 55% of the WHO minimum standard of 100 l/c/d. The statements show that the communal water supply for the Palestinian population is substantially inadequate by international standards. The available water resources in the Middle East are scarce, limited, fragile and threatened. They are already exploited, especially in Palestine [6] [7].

Our project focused on the monitoring and controlling of water distribution using IOT based model. It aims to design and develop a low cost reliable and efficient technique to improve water distribution in the community. Technologies to be used in the project would be as follows:

1. IoT

The Internet of Things (IoT) describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet [5]. In our project, we used the sensors in water tanks to describe the amount of water in each tank. The amount of water coming out of the tank is then sent via the internet, to be later stored and analyzed in a database.

2. MQTT: The Standard for IoT Messaging

MQTT (MQ Telemetry Transport) is a lightweight, publish-subscribe network protocol that transports messages between devices. The protocol usually runs over TCP/IP, however, any network protocol that provides ordered, lossless, bi-directional connections can support MQTT. It is designed for connections with remote locations where resource constraints exist or the network bandwidth is limited [18].

3. Mobile Application

A mobile application is a type of application software designed to run on a mobile device. Mobile applications frequently serve to provide users with services [3]. We use a mobile application to display the data that comes from the microcontroller to the user, and this app allows the user to control valves by sending a signal from the app to the microcontroller.

2.3 Literature review

In this section we will talk about some projects and ideas related to our project idea.

2.3.1 Smart Water Distribution Network

The backbone of the smart water network is the water distribution network with electrically actuated valves (butterfly valves) that controls the flow of water to the households, area-wise and street-wise. The water level sensors will be designed to monitor the availability of water in the ground level storage points. The system identifies the leaks based on pressure information or with other simpler techniques and informs the authority the position of leak accurately so that the water gets conserved and also helps the authority to send the labor force to the right place of leak in a right time. Water quality parameters like pH, Turbidity and free residual chlorine are monitored and informed to the authority and also the consumers. Other water quality parameters like, temperature, conductivity, dissolved oxygen would also be considered in the system design [1].

2.3.2 Smart Water Tank

This work was done by Hamza Yousef Tarada and Osama Mazen Ayyad. The main idea of this project is to provide high quality water and prevents freezing of water in tanks. Their system checks the temperature of the water in the tank, and measures the percentage of consumed water as well as knowing the amount of water available in the tank and controlling the process of entering and leaving the water from the tank [16].

2.3.3 Water level monitoring

This work was done by Areen Nadi Shalaldeh, The main idea of this project is to monitor the level of water using an ultrasonic sensor connected with a microcontroller (raspberry pi3), then this information could be accessed from the user's phone. This project has been implemented based on a certain software that is loaded on the raspberry pi. On the other side, the raspberry pi is connected to the internet (LAN), and the user connected to the same LAN, so he could access the result in an easy way. This project is able to detect the level of water in a certain water tank, or other water storage system that is similar to it. The system senses the amount of water in the tank, and the user have the result while the tank is filling with water using his mobile phone as a practical, and low cost solution [2].

Table 2.1 shows a comparison between our system and other built systems.

Table 2.1: comparison between previous projects based projects

	Smart Water Distribution Network	Smart Water Tank	Water level monitoring
Goals	Water quality parameters like pH, Turbidity and free residual chlorine are monitored and informed to the authority and also the consumers.	Measuring the amount of available water, water turbidity, temperature, consumption rate, and also controlling valves based on emerging cases	Monitor the level of water
Succeeded in	Control the flow of water into homes, at the district and street level.	Reducing wasted water through water monitoring at the level of residential building tanks	Creating a low-cost, easy-to-handle water level monitoring system.
Design components	Arduino Wireless data transmission using GSM , Bluetooth, sensors to measure (pH, Free Residual Chlorine, Turbidity, Dissolved Oxygen, Pressure) water flow control valve.	NodeMCU ESP32, ultrasonic sensor, solenoid valve, water temperature sensor, water flow sensor, water purity sensor.	Raspberry pi3, water flow sensor, water Pump, water level sensor, and sensor interface.
View data remotely	Yes	Yes	Yes

Differences from Our project	We will distribute water to neighborhoods. While they distributed lines with pressure/flow sensors and electrically actuated valves.	We will create a fair water distribution system that regulates the flow of water to every neighborhood of the city. While they did not distributed water.	We will calculate the water consumption rate, the amount of water that entered the tank, control the entry and exit of water from the tank using valves. While they monitor the level of water.

2.4 Description of system

The project's main idea is to design and implement a smart IoT-based water distribution system. Sensors are used to collect data from the tank then send it to the microcontroller which controls the valves of the tank, after that the microcontroller will send it to the application which will show the data and notifications. To achieve the goals of our project, we need several components. Sensors are needed to measure the distance for the amount of water in the tanks, to measure the water flow, and control the process of entering and exiting water coming from the main sources to the main tanks and then to the Slave tank through the valves.

To achieve this, we want to store values obtained from the system in the database. The obtained data is analyzed to make the proper decisions accordingly. To achieve the calculations of the amount of water in the tank, we need the ultrasonic sensor and two flow sensors. To measure the rate of flow of water. To find the total round-trip distance of the sound wave. We can use the following formula:

$$D = (S * T) / 2.$$

Where D is the distance, S is the speed of waves, T is time taken by waves to fall back. The idea of using flow sensors, is that we want to put one of them at the place where the water enters the main tank, Let's name it as Flow1, and we want to put the second at the place where the water exits from the main tank, and let's name it as Flow2, and we put the ultrasonic sensor at the top of the tank, name it as U. Accordingly, we have:

water consumption rate = Flow2

amount of water that arrived from the source = Flow1

amount of water available in the tank = Flow1 - Flow2

If we assume that the water main tank is a plastic model, then we will place the ultrasonic sensor at the top of the main tank, and there will be a valve for the exit of water from the main tank at a height specified from the bottom of the main tank. If the remaining water in the tank is not enough, the controller will send a signal to close the valve that allows water to exit the main tank.

The pump will It was used to pump water in case the water direction was upward, and if the available amount of water was little, it needed to be pushed to complete the water distribution.

2.5 Design options and Specification:

This part describes the hardware and software components and their design options.

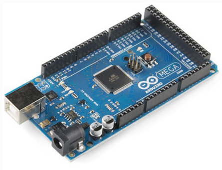
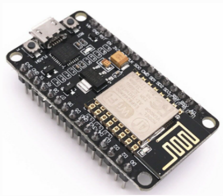
2.5.1 Hardware components:

2.5.1.1 Microcontroller

We need a microcontroller, to deal with the data received from the connected sensors, analyze it and schedule distribution schedules based on it, and send it to the municipality using the application.

The options for the microcontroller used are listed in Table 2.2.

Table 2.2: List of Microcontroller options [10]

Microcontroller		
Type	Image	Specifications
Raspberry pi		1. High cost. 2. Small size. 3. Suitable to install. 4. non-built in WiFi. 5. non-built in bluetooth.
Arduino mega		1. Low cost. 2. Small size. 3. Suitable to install. 4. Functionally the same as their larger counterparts. 5. fast . 6. Limited on board memory can make complex programs difficult.
ESP8266		1. Low cost. 2. built-in WiFi. 3. built-in bluetooth. 4. small size. 5. suitable to install.

We chose ESP8266 because it is easy to deal with, contains wifi and Bluetooth, Clock frequency 160kHz / 240kHz, contains Internal Flash Memory size 4 GB,16 Pulse-width modulations, 34 General-purpose input/output, low cost, high performance, low power consumption, easy to program, wireless SoC(System-On-Chip).

2.5.1.2 Sensors

1- We need a sensor that measures the amount of water in the main tanks via placing it at a point at the top of the tanks and it will measure the distance between it and the level of water in the tank. This sensor sends the data obtained to the microcontroller, where it will be connected to it.

The different options are listed in Table 2.3.

Table 2.3: Ultrasonic sensor versus Sharp distance sensor [11]

Sensor measures the amount of water in the main tanks		
Type	**Image**	Specifications
Sharp distance sensor		1. Small size. 2. High resolution. 3. Fast. 4. Detection range low. 5. High cost.
Ultrasonic sensor		1. Low cost. 2. Small size. 3. Higher resolution. 4. Detection range high. 5. Fast . 6. Middle cost. 7. High power usage.

We chose Ultrasonic sensor because it is easy to deal with, the cost is middle, the resolution is higher, the operating current is lower, and the detection range is higher.

2- We need valves to control the falling of water in tubes from the source to the main tanks. and from the main tanks to the slave tanks.

The different options are listed in Table 2.4.4

Table 2.4: Solenoid valves versus Ball valves [12]

Valves		
Type	Image	Specifications
Solenoid valves		1. Small size. 2. Low power. 3. Fast response time. 4. Remote operation. 5. Sensitive to voltage.
Ball valves		1. High cost. 2. Middle size. 3. Quick to open and close. 4. Connecting two pipes going in the same direction.

We chose Solenoid valves because it is easy to deal with, the cost is lower, and use ball valves as an inline valve for connecting two pipes going in the same direction. As for the Solenoid valves it is used in all cases.

3- We need a sensor that measures water flow to calculate the rate of water consumption in a certain period of time by calculating the amount of water leaving the tank and the amount of water entering it. The different options are listed in Table 2.5.

Table 2.5: Water Flow Sensor versus Flow Meters [13]

Sensor measures water flow		
Type	Image	Specifications

Water Flow Sensor		1. Low cost. 2. Small size. 3. Operates in both flow directions (forward and reverse). 4. Fast response time. 5. Low maintenance 6. high pressure drops
Flow Meters Sensor		1. Small size. 2. High reliability. 3. Durability. 4. Low maintenance. 5. High cost.

We chose Water Flow Sensor because it is easy to deal with, the cost is lower, small in size, operates in both flow directions (forward and reverse), fast response time, and less need for maintenance.

2.5.1.3 Pump

We need a Pump that is used to pump water in case the water direction was upward, and if the available amount of water was little, it needed to be pushed to complete the water distribution. 2

Table 2.5: Water Pump versus Motor Water Pump [16][17]

Pump		
Type	Image	Specifications
Water Pump		1. Middle cost. 2. Middle size. 3. Operates in both directions (push and pull). 4. Fast response time. 5. Low maintenance 6. Rated flow 30 L/H.
Motor Water Pump		1. Middle size. 2. High cost.. 3. Need a motor 4. Rated flow 650 L/H.

We chose Water Pump because it Operates in both directions (push and pull), Middle cost, Middle size and Low maintenance.

2.5.2 Software components:

2.5.2.1 Arduino IDE

The Arduino Integrated Development Environment or Arduino Software (IDE) li is open-source, contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino and Genuine hardware to upload programs and communicate with them. It runs on Windows, Mac OS X, and Linux, written in the Java programming language.

2.5.2.2 Types of Software

Table 2.7 a comparison of techniques to choose one to be used in the project.

Table 2.7: Mobile Application versus Web Apps [9]

Development Options:	Specifications
Mobile Application	1. Improved performance and user experience. 2. Full access to functionalities of underlying mobile OS and device specific capabilities. 3. Highly complex as cross-compilers need to be kept consistent and up-to-date with mobile platforms.
Web Apps	1. Multiple and NodeJS 2. Limited access to OS API'S

We chose to develop the Mobile App due to its look and Feel and Performance in both platforms (Android & IOS).

2.6 Design constraints.

1. Permanent need for the internet.
2. This system works for the main reservoirs responsible for distributing water to the slaves that represent a group of people's subscriptions.
3. Height of the tanks is not more than 4 meters.

(The used sensors can measure up to 4 meters. However, for simplicity, we will use plastic tanks with short heights.)

Chapter Three

System Design

3.1 Overview

In this chapter, we will explain the abstract block diagram of the system. Next, the detailed conceptual description of the system is introduced. The design diagrams will be explained in detail, Too.

3.2 Detailed conceptual description of the system

Our system will be designed to establish a fair distribution of water to all neighborhoods during sufficient periods. When the system is activated, The ultrasonic sensor will measure the water level, and then it will send it to the ESP8266. If the tank is empty, the ESP8266 will close the valve of the water tank. There are three flow sensors that will calculate the rate of water consumption and the amount of water available in the tank. These sensors send the data to the ESP8266 to process it. The data is then sent to the database via WiFi using the MQTT protocol. The system will analyze the data such as water level, rate of water consumption, and water available in the tank. The municipality is informed all the time with the tank's status and the system suggests solutions based on the analysis of the data. The system is described in figure 3.1 below.

The model will be designed using a main tank with a height of 12cm and two slave tanks with a height of 12cm. The connection between the main tank and the slave's tanks will be done using plastic tubes.

Some data will be processed inside the microcontroller, such as the water level.The valves will be controlled by the microcontroller, and the amount of water distributed, the time of distribution and the time of repetition within the database will be treated.

The fair distribution of water uses a mechanism for equitable distribution of water to neighborhoods all every two weeks and it works as follows:

Statistics are brought from the municipality about the number of subscriptions attached to each tank and the percentage of water consumption for each subscription. The system calculates the amount of water distributed to each neighborhood considering whether there is a hospital or government organization that increases the need for water.

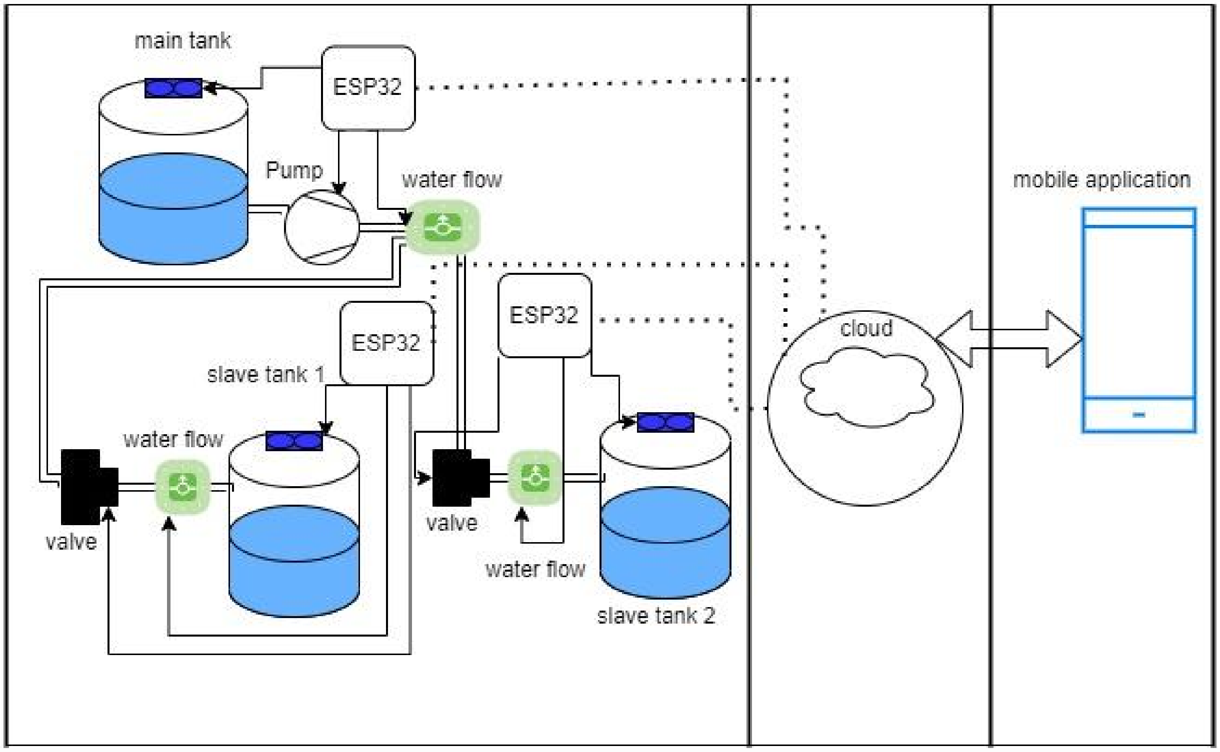


Figure 3.1: General system description

The data will be analyzed on the basis of the share of each of the sub-tanks that were identified according to the number of people, where the water in the main tank is examined if it is sufficient for the needs of the sub-tanks, the distribution will be according to the equation:

needed_to slave tank= specific need slave tank- (slave tank height - slave tank_ultrasonic *10)

If the water in the main tank is less than the need of the sub tanks, the distribution is done according to the following equation:

If ((needed_to slave tank1 + needed_to slave tank2) > water in main tank)

{ (needed_to slave tank1 = needed_to slave tank1 / (needed_to slave tank1 + needed_to slave tank2) * water in main tank

needed_to slave tank2 = needed_to slave tank2 / (needed_to slave tank1 + needed_to slave tank2) * water in main tank }

Leak detection by :

Check if water level = water level before distribution - amount of water distributed

The system allows citizens to add their notes on water distribution to be sent to the municipality for study. We will use XAMPP to associate the database with the web page on which the data is stored.

3.3 System block diagram

Figure 3.3 shows the general block diagram of the project. As illustrated below, when the system is activated, the sensors will collect the necessary data and send it to the ESP8266. The microcontroller will then process it and send it to the database for storage and analysis. The data obtained from the ultrasonic sensor and flow sensors will be manipulated to give the right decisions for the closing/opening of valves and control of pumps.

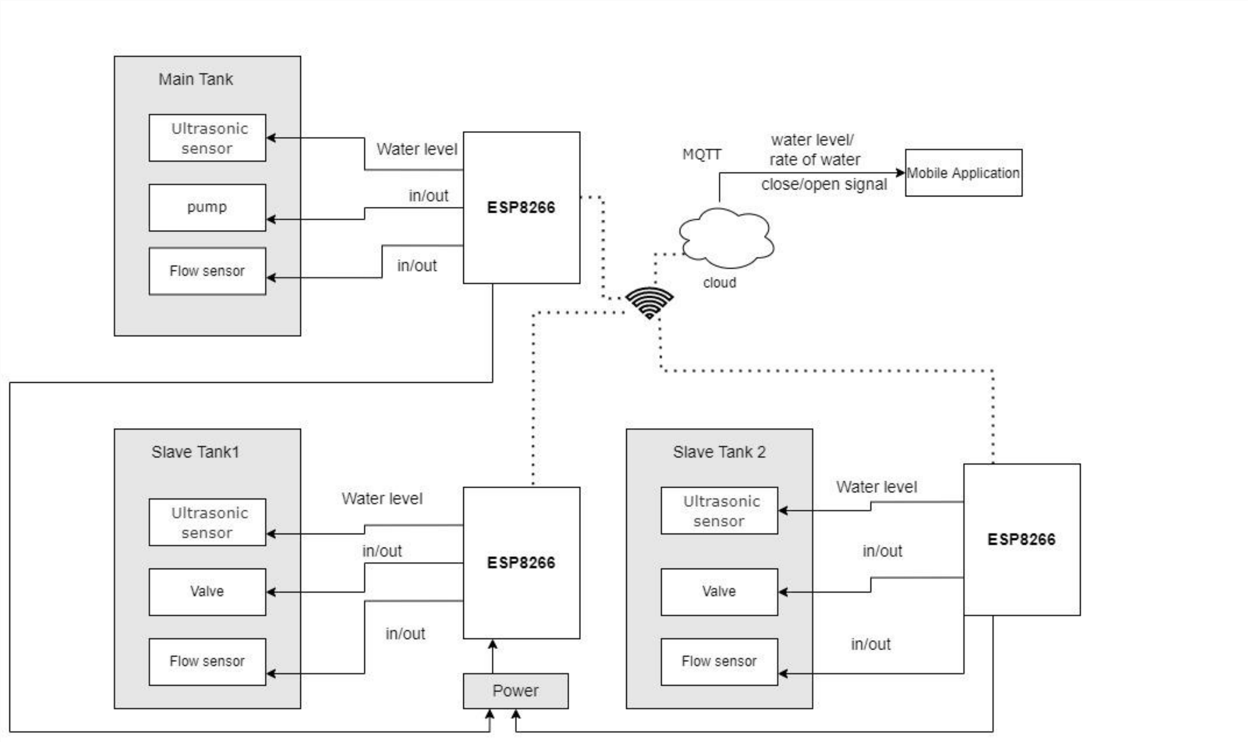


Figure 3.3:General block diagram of the system

3.4 Schematic diagram

Figure (3.4),(3.5) and (3.6) describe the overall system elements and how they are connected. Figure 3.4 shows Schematic diagram of main tank, while figure 3.5 shows Schematic diagram of slave tank 1, while figure 3.6 shows Schematic diagram of slave tank 2.

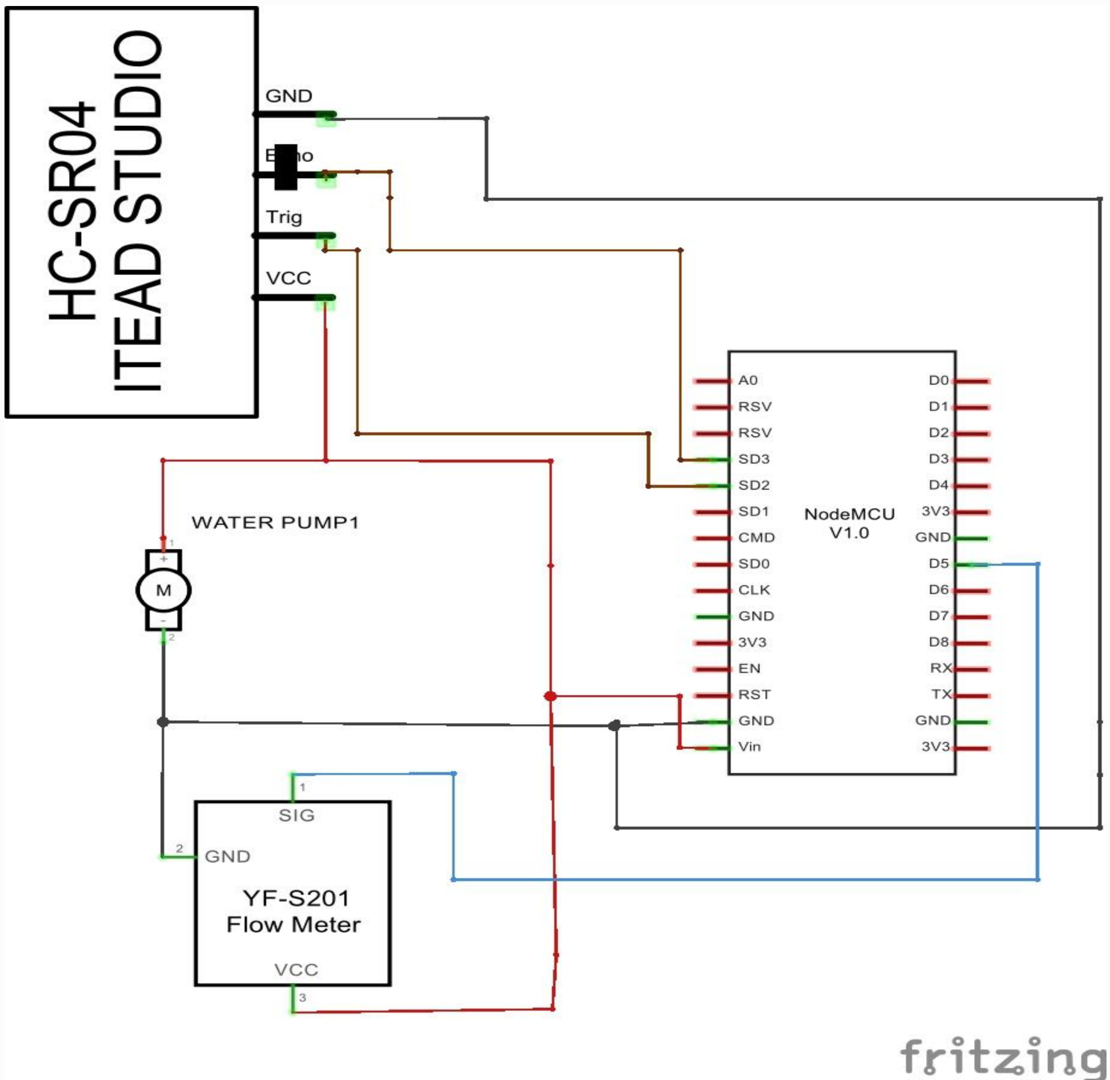


Figure 3.4: Schematic diagram of main tank

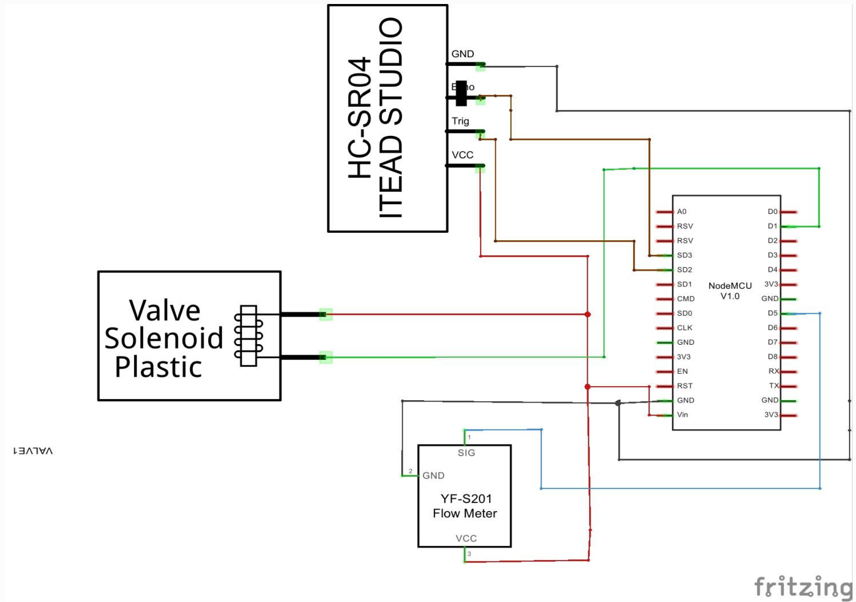


Figure 3.5: Schematic diagram of slave tank 1

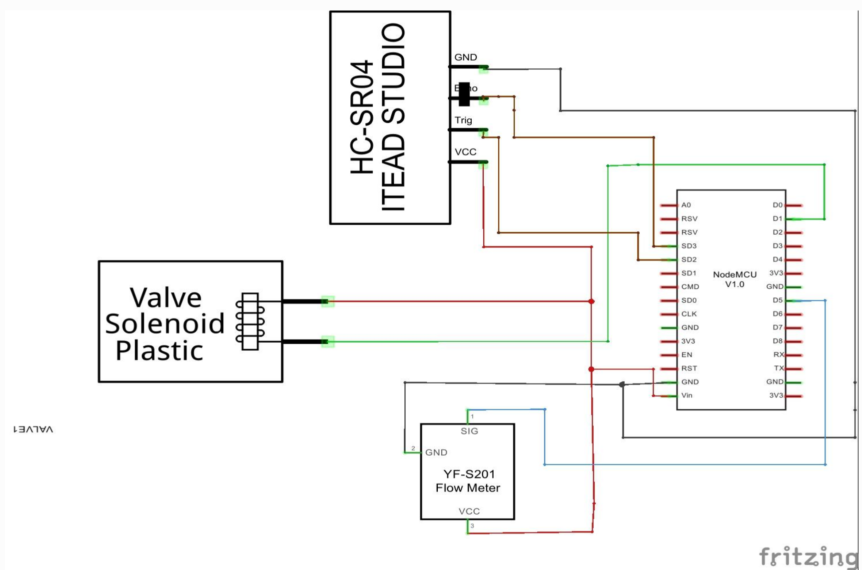


Figure 3.6: Schematic diagram of slave tank 2

The main components appear in the block diagram are as follows:

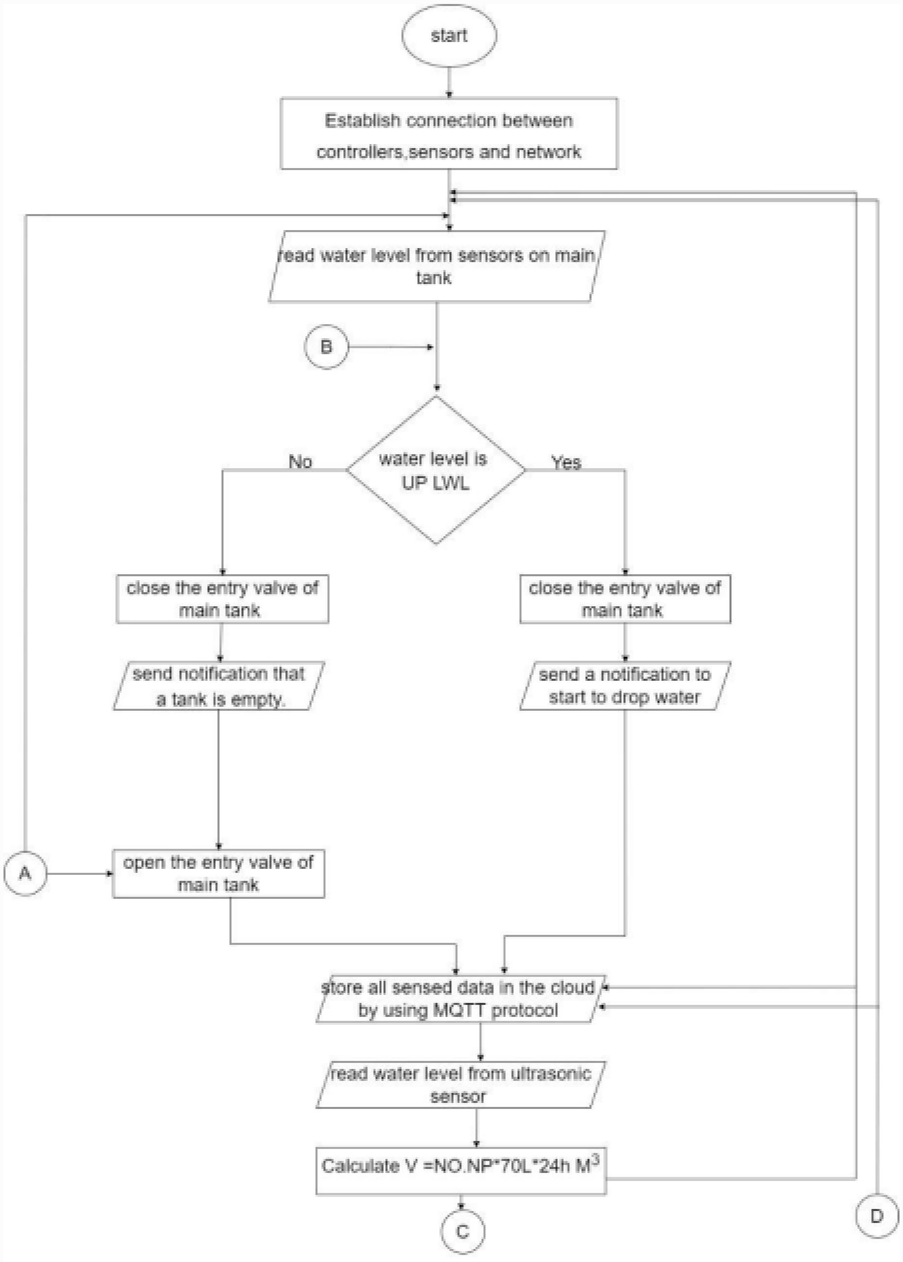
1. ESP8266: The main component of the system, where all components are connected to it. It will control the valves based on the data it receives from the sensors connected to it.

Then, it will send the data accordingly, to the database for storage and analysis and display the proper status on the mobile application.

2. Ultrasonic sensor: It is used to measure the distance between it and the surface of the water. If it is about to running out or to fill, it sends a signal to the ESP8266.
3. Valves: It is used to allow water to enter and leave the main water tanks and slave water tanks based on the signal coming to it from the ESP8266.
4. Flow sensors: It is used to calculate the rate of water consumption, the amount of water in main water tanks that entered the tank, and also the amount of water available in the main tanks.
5. Pump: It was used to pump water in case the water direction was upwards, and if the available amount of water was little, it needed to be pushed to complete the water distribution.

3.5 Flow chart

This diagram shows the flow chart of the system, these are shown in Figures 4.7 Details of the abbreviations used:


start
Establish connection between controllers,sensors and network
read water level from sensors on main tank
B
water level is UP LWL
No
Yes
close the entry valve of main tank
close the entry valve of main tank
send notification that a tank is empty.
send a notification to start to drop water
A
open the entry valve of main tank
store all sensed data in the cloud by using MQTT protocol
read water level from ultrasonic sensor
Calculate V =NO.NP*70L*24h M3
C
D

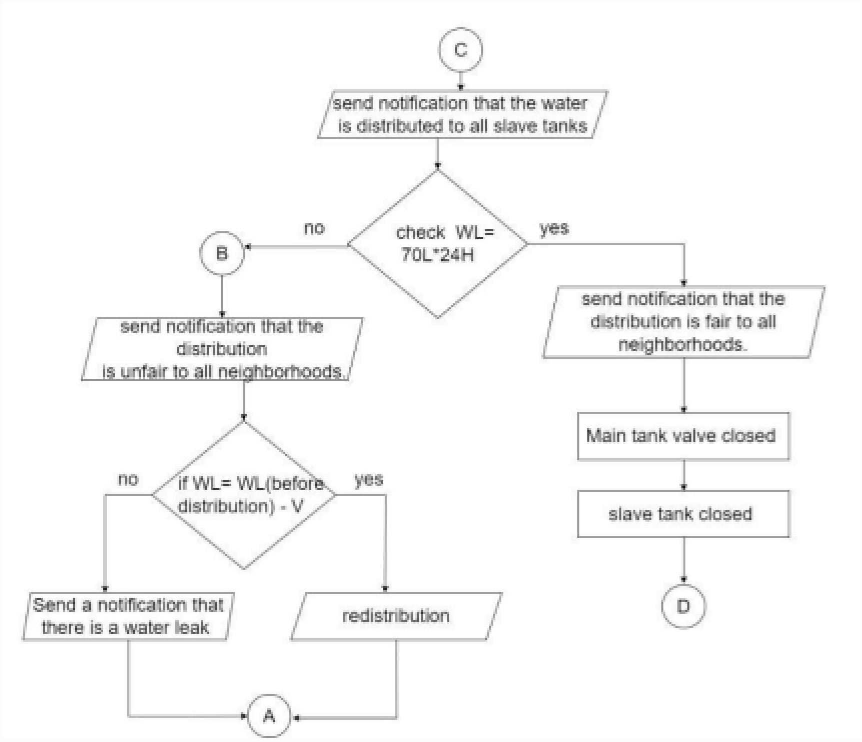


Figure 3.7: Flowchart of system

3.6 Summary

A detailed conceptual description of the system, system block diagrams, a detailed system design, a system flow diagram, and a sequence diagram, are provided with the diagram.

Chapter four

System implementation, testing, and discussion

4.1 Overview

This chapter introduces a description of the implementation, implementation issues, implementation challenges, description of the methods used to validate the system, validation results which include an analysis and discussion about the results and recommendations based on the results.

4.2 Description of the implementation

4.2.1 Software Implementation tools

C code is just used as a partial interface between humans and the Microcontroller to instruct the Microcontroller so as to perform a particular operation. After the code is written, it is compiled. App Inventor code is used for building a mobile application. Then we used XAMPP to connect database with web page on which the data is stored.

4.2.2 Communication

To connect the main tank ESP8266 with The slave tanks' ESP8266 a Wi-Fi connection is needed before we start showing the data of sensors. Implement a Wi-Fi connection between the tanks using the "ESP8266WiFi.h" library.

4.3 Implementation issues and challenges:

We faced some challenges during the system implementation, some of them:

1. Finding the appropriate microcontroller.
2. Storing data in a cloud and analyzing this data.
3. Errors during connection and learning how to connect the ESP8266 to the internet.

The solutions to the previous implementation issues that we encountered:

1. The appropriate microcontroller for the system is ESP8266 supported with enough digital pins, and contains wifi. It also has a suitable size.
2. We use MySql database to store data because the clouds that support the MQTT protocol are expensive. The data analysis was carried out on the microcontroller of the main tank.

3. We made a private WiFi in it with a specific name and a specific password.

4.4 Validation result

4.4.1 Hardware testing

All NodeMCU ESP8266 were tested with all sensors and connected directly to the laptop as well as the 12V power for all tanks (main and slaves).

Unit testing on the main tank:

1. We connected ESP8266 with Wi-Fi, to communicate with the main tank.
2. We connected ESP8266 with ultrasonic sensor, to read distance.
3. We connected ESP8266 with flow meter, to read the amount of water.
4. We connected ESP8266 with water pump, to pull and push the water.

Result, we made sure that the system was working successfully.

Integrating testing on the system:

1. We connected three ESP8266 with Wi-Fi, the communication between the three tanks was done successfully.
2. Test done on the system as a whole containing 3 ESP8266 and confirmed that they work well when combined together.

This image represents the project that was built in the university lab and that we did the tests on.

By the end of the implementation process, the system (shown in figure 3.1) was

constructed successfully.

(Figure 4.1) shows the Hardware Components of the system.

(Figure 4.2) shows the main tank, on which all components are fixed.

(Figure 4.3) shows the slave tank, on which all components are fixed.

Detailed information about the connectivity of these components are presented in

Section 3.2.

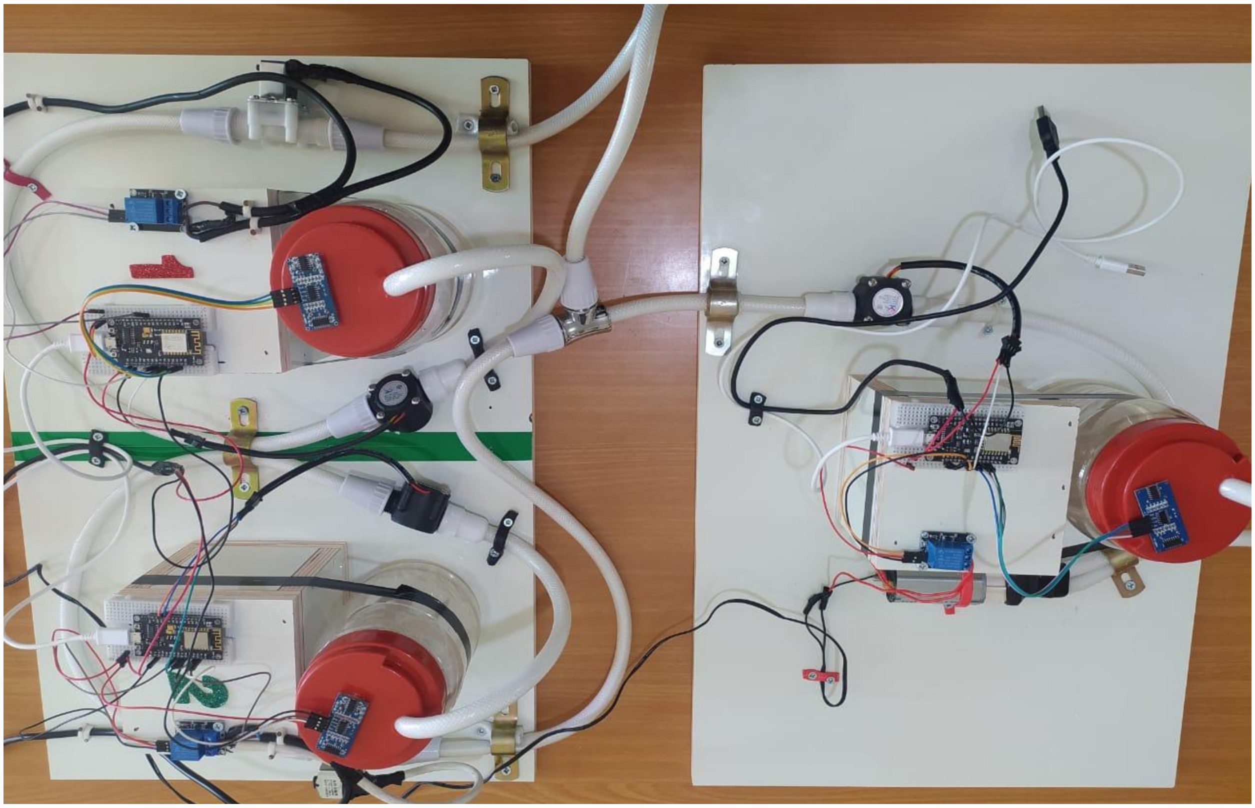


Figure 4.1: Hardware Components

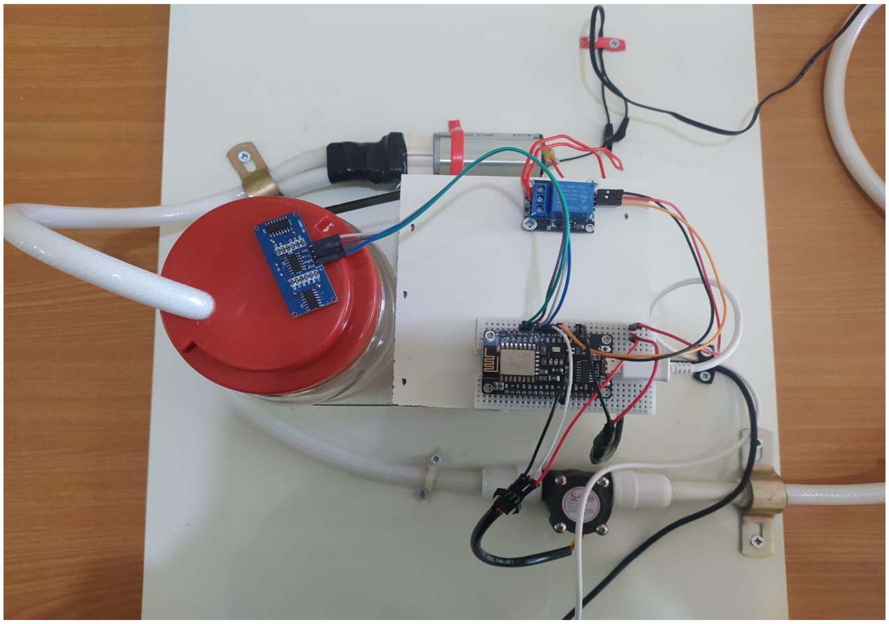


Figure 4.2: Hardware Components of main tank

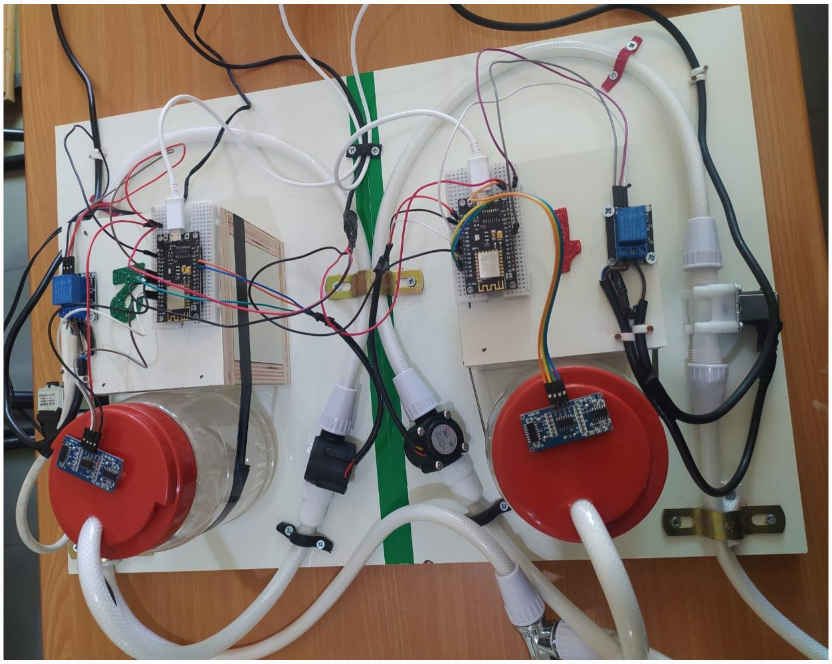


Figure 4.3: Hardware Components of slaves tanks

4.4.2 Software testing

The system was fully checked and ensured how it worked. The developed software was successful. Table 4.1, table 4.2 and table 4.3 show some of the tests that we have carried out.

Table 4.1: Software Testing

Case	Expected Output	Obtained Output	Pass/Fail
Connect to the internet	Connect to the internet	Connect to the internet successfully	Pass
Connect to the MQTT	Connect to the MQTT	Connect to the MQTT successfully	Pass
Get necessary data	Get data from	Get data successfully	Pass

from the database	database		
Get necessary data from sensors	Get data from all sensors	Get data successfully	Pass
Send data to database	Send data to database	Send data successfully	Pass
Water distribution	Water distribution	Water distribution successfully	Pass

Table 4.2: Sensors Testing

Case	Expected Output	Obtained Output	Pass/Fail
Turn on the ultrasonic sensor to calculate the water level inside the tanks	Turn on the ultrasonic sensor	Turn on the ultrasonic sensor successfully	Pass
Open the pump if there is enough water in the main tank	Open the pump if there is enough water in the main tank	Open pump successfully	Pass
Turn on the water flow sensor to calculate the amount of water flowing	Turn on the water flow sensor to	Turn on the water flow sensor successfully	Pass

Open valves when distributing water	Open valves when distributing water	Open valves successfully	Pass

Table 4.3: Mobile Application Testing

Case	Expected Output	Obtained Output	Pass/Fail
Mobile application returns data	Show data in the user interface for the main tank and slaves	Show data successfully	Pass
Show current system status	Show current system status (valves, pump) on or off	Show data successfully	Pass
Show if there is a problem	show text in a user interface	Show data successfully	Pass

- The user is asked to connect to the "w_d_s" network. After the connection, he can enter the mobile application, where he is asked to enter the username and password.

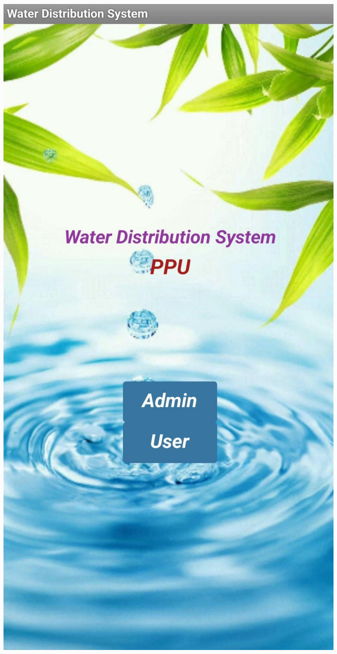


Figure 4.5: mobile application interface

- When the user opens the mobile application, the user goes to the home page which contains a list of the names of the tanks in the system as shown in Figure 4.2.

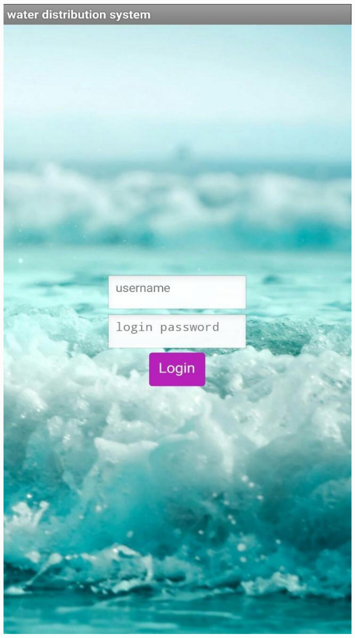


Figure 4.6: interface 2

- Then the user goes to the page belonging to each tank, whether the Main, Slave 1 or Slave 2, which contains data on the pump for the main tank and valves for the slave's tanks which can turns on or off by clicking him and ultrasonic sensor and water flow sensor, as shown in Figure 4.3.

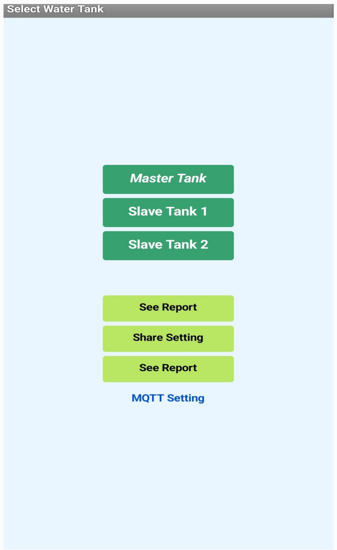


Figure 4.7: interface 3

- Then it shows the water level in the main tank and slaves tanks and the amount of water flowing from the main tank to the slave's tanks, as shown in Figure 5.4.

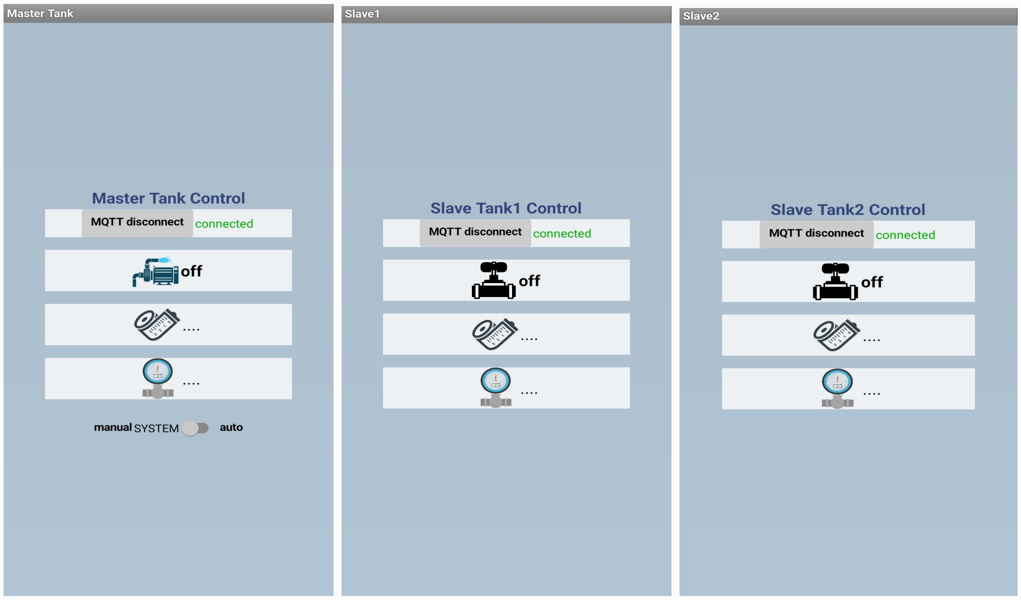


Figure 4.8:interface 4,5 and 6.

● User feedback interface: The interface that enables citizens to add their notes on water distribution.

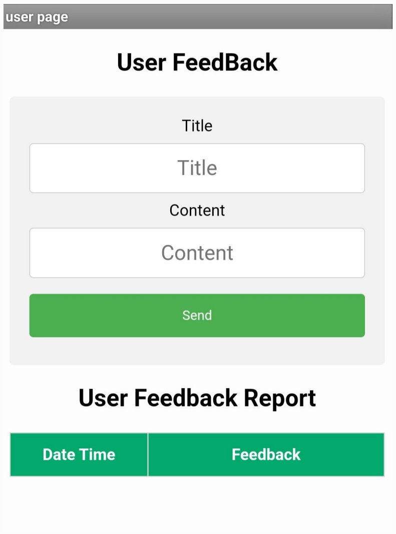


Figure 4.9: interface 7

● MQTT setting page: MQTT settings are set by entering each of the(broker, port, client id, username, password) and saving this data, as shown in Figure 4.6.

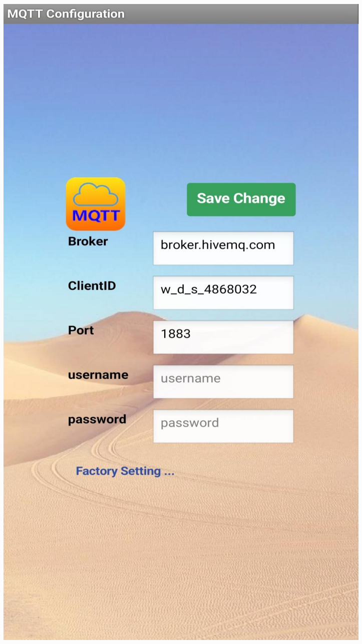


Figure 4.10: interface 8

All data are stored in MySql Database in the fields shown in figure 4.10.

	Table	Action	
	quantity_distribution_history		Browse
	set_tank_share		Browse
	user_feedback		Browse
	3 tables	Sum	

Figure 4.11:MySql Database.

4.5 Discussion

From the first stages of the project implementation we were following the purposed objectives of the system. Components have been connected and programmed to achieve the requirements of the system. After that, we built the system and performed the necessary tests to ensure that it works in the required manner and achieves the goals successfully. The system receives the data from the sensors correctly, the controller processes and analyze the data by calculations done on the obtained data then store the last version of data. The pump and valves were controlled based on the obtained values. The process of sending and receiving data via the MQTT protocol from the database were done correctly. The mobile application displayed the required message and made the control as needed.

Chapter five

Conclusion and Future work

5.1 Overview

The chapter introduces a summary of the project, the future directions, and future work.

5.2 Conclusion

Our project focused on monitoring and controlling water distribution using IoT-based models. It aims to design and develop a low-cost, reliable and efficient technology to improve water distribution in the community. At the end of the project, we were able to measure the water level in the master tanks and slave tanks, measure the amount of water flow to all tanks, and fetch values from the system's mobile application through the application interfaces. The system was able to control the water distribution process according to the amount of water in the main tank and the amount of water needed by each of the slave tanks. The information is stored and analyzed, and then the pump is turned on in the main tank. The valves of the slave tanks were able to be opened and closed on time and on purpose. The results have been overwritten by the built-in mobile application. Finally, citizens were able to write notes to assess the distribution process.

5.3 Future work

In the future, we look forward to adding important features to the system, the most

important can be:

- Developing the system to deal with more than one main tank and more slave tanks.
- Developing and enlarging the pump system inside the tank so that the tank can be emptied.
- Develop the system to check for impure water inside the tanks.
- Connecting the system with the authorities responsible for managing the so that it can be controlled better and faster.

References

[1] Cdac.in. Available from: https://www.cdac.in/index.aspx?id=pe_i.

[2] Palestine Polytechnic University. Available from: Water Level Monitoring (ppu.edu).

[3] Mobile Application. Available from: https://www.techopedia.com.

[4] Oracle.com. Available from: https://www.oracle.com/.

[5] Researchgate. Available from: https://www.researchgate.net/.

[6] Sciencedirect. Available from: https://www.sciencedirect.com/.

[7] World Health Organization. Available from: https://www.who.int/news/.

[8] Appstudio Canada. Available from https://www.unicef.org/press-releases.

[9] en.wikipedia. Available from: https://en.wikipedia.org/wiki/List_of_common_microcontrollers.

[10] Seeed Studio. Available from: https://www.seeedstudio.com/.

[11] Sovna valve. Available from: https://www.covnavalve.com/.

[12] Arad Group. Available from: https://arad.co.il/blog/water-meters-vs-flow-meters/.

[13] amazon. Available from: https://www.amazon.com/Turbidity-Quality-Monitoring-Detection.

[14] Tethys. Available from: http://www.tethys-instruments.com/TurbiditySensor.aspx.

[15] Palestine Polytechnic University. Available from: Smart Water Tank (ppu.edu).

[16] GWS. Available from: https://shop.gwsonline.it/shop/pompa.

[17] amazon.in . Available from: https://www.amazon.in/Atoz-prime.

[18] TechTarget. Available from: https://www.techtarget.com/iotagenda/definition/MQTT.

Appendices

Appendix A Data sheets for needed components:

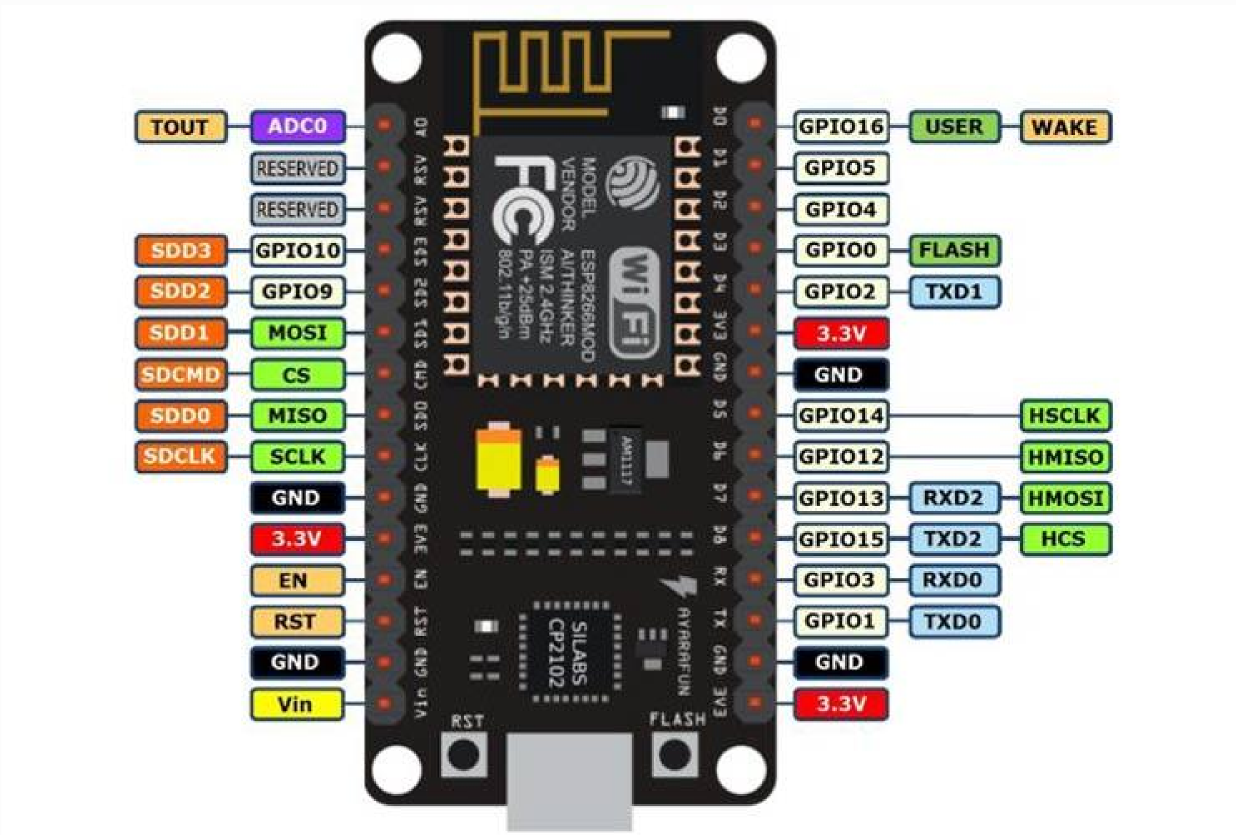


Figure 1: NodeMCU ESP8266 pinout