Palestine Polytechnic University

College of Information Technology and Computer Engineering

Department of Computer System Engineering

"Smart Water Tank"

Team members:

Hamza Yousef Tarada

Osama Mazen Ayyad

Supervisors:

Eng. Khalid Daghameen

Dr. Radwan Tahboub

Palestine
2021

Acknowledgment

First of all, all the praises be to Allah Almighty and thank him for his blessings throughout our work on our project. He gave us strength and knowledge, and he helped us succeed in this project.

We would like to express our sincere gratitude to our supervisors Eng. Khalid Daghameen and Dr. Radwan Tahboub, for guidance, direction, encouragement, and absolute patience throughout the project period; transferring their knowledge and experiences to us and urging us to make the project distinct.

Moreover, we would like to thank Eng. Wael Takrouri for his support and assistance to complete this project.

We would like to thank all the 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.

Abstract

With the large increase in the population that led to the presence of taller buildings, and with the lack of water from its source and its frequent interruptions that cause a great shortage of water for the population, water often runs out from tanks belonging to homes without prior knowledge. So monitoring of water tanks has become one of the daily routine tasks that may consume time and effort and maybe dangerous at times in tall buildings. The need for an automated system arose to monitor the water tank and the ability to know data remotely through the application.

Our project seeks to provide a system that performs several operations, including checking the quality of water. Also, this system checks the temperature of the water in the tank to avoid some problems that may occur, especially when the water freezes. Also, the system 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. This system also deals with cases that may occur such as the poor quality of the water. Therefore, this system provides the inhabitant with all the information that he may need from anywhere he is located through a mobile application without time or effort and without the need to manually check the tank.

The system has been built and verified to meet the requirements of this project, and it was found that it performs as it should and meets the needs of the project.

Table of Content

List of Tables

List of Figures

Chapter one

Chapter one

Introduction

1.1 Overview

The smart water tank is an automated system designed to automatically control the water tank, especially in homes. This system enables the user to view some data that takes in our consideration such as the amount of water, running out case, water temperature, and water turbidity ratio through a mobile application, it can also close and open the tank according to the situation, such as closing the tank when the water is close to running out or when the rate of water turbidity is high, and also this system can measure the water temperature, especially in the winter, When the temperature is low, i.e. close to zero or less, the system leaks water from the tank to avoid freezing of water inside the pipes, which may lead to its damage.

1.2 Motivation and importance

The need for this system was due to the large number of problems that occur in the water tanks, most of which may happen without the knowledge of homeowners before the problem occurs or the moment it occurs, as they most often know after a problem has occurred, which may cause material damage such as valves or float damage. Also, there are water tanks in tall buildings and this may pose a risk to people who check their tanks periodically and also monitor the quality of water that may affect the health of people who drink from this water, especially the elderly and those suffering from diseases.

1.3 System objectives

The main objectives of this system are:

- The system will be able to check the turbidity of the water to a certain threshold.
- The system will be able to check the water temperature of the water to a certain threshold.
- The system will be able to check the amount of water in the tank.
- Perform tank closure when the water quality is poor or the water is close to running out, In addition to sending a notification to the mobile application of the status.

- The system will be able to check the rate of water consumption periodically for every hour or daily.

1.4 Problem statement

In this section, we're going to talk about problem analysis, a list of requirements, expected results, and definitions.

1.4.1 Problem analysis

Many people suffer from several problems in their water tanks, such as the problem of high water turbidity, which may affect those who suffer from diseases or the amount of water they have depleted without prior warning, which may cause problems with water traps, and sometimes the water may freeze inside the pipes in a separate winter. Therefore, we need a system that performs several operations to help us solve these problems, record the events that occur and solve them in the best way, and inform the user about the situation that happened.

1.4.2 List of requirements

The system requirements can be summarized as:

1. Collect all data from the tank such as water temperature.
2. The system must always update the status of the tank periodically and send the data to the mobile application.
3. The system should give an alarm if a problem occurs, or if the water is close to running out, or if there is significant water consumption, or if the water turbidity is high.
4. In cases where the system stops working, it should give a warning signal.
5. Build a smart algorithm for anticipating the amount of water.

1.4.3 Expected results

Expect to build an integrated system with the following specifications:

1. The system must be able to check the water quality, the consumption rate, and measure the temperature of the water inside the tank, as well as the water percentage in the tank.
2. The system must send all data related to the tank to the mobile application and display it to the user.

1.5 Short description of the system

This system works as follows:

This system controls the percentage of water inside the tank and checks the rate of water consumption during a certain time and periodically, it also checks the percentage of turbidity and if the turbidity percentage is high, the system closes the tank and sends a notification to the user about the condition, and also if the water is close to running out, the tank is closed. This system also measures the temperature of the water inside the tank as this process is very important in the winter season, especially when the air temperature is very low, i.e. close to zero or less, when the temperature is low, the system leaks water from the tank through the pipes so that it does not freeze water inside the pipes, which may lead to damage to the pipes. The system will use sensors to carry out the previous operations, and these sensors will be connected with the microcontroller, where the data will be downloaded to it and sent via the Internet to a mobile application so that the user can see all the data.

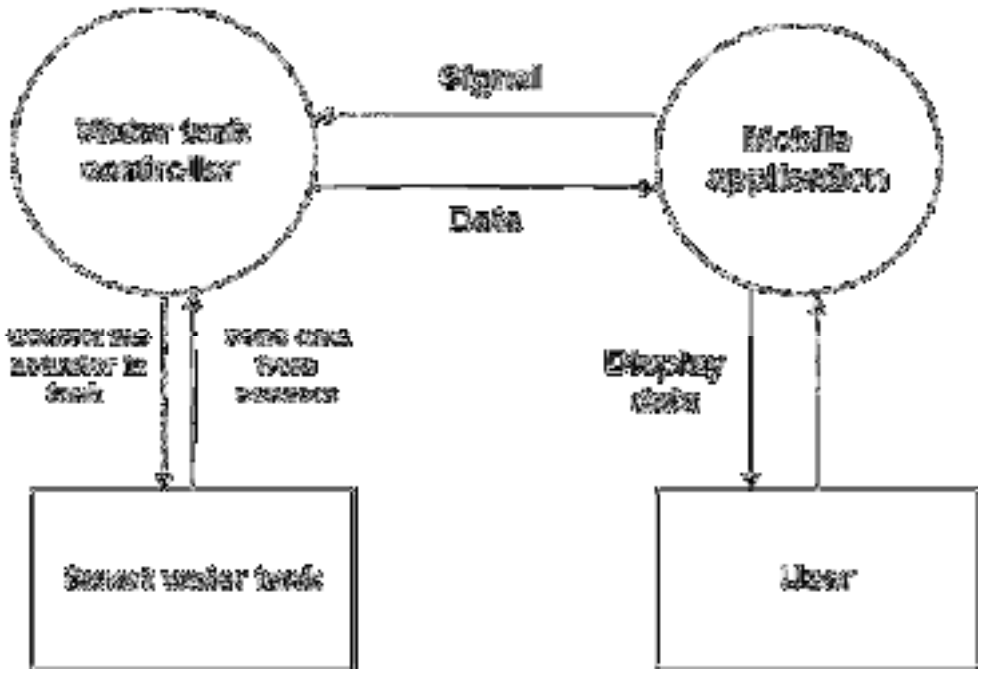

Figure 1.1: Context diagram of smart water tank

1.6 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, a description of the hardware and software components is discussed in addition to the system specification and design constraints. Chapter 3 detailed design, block diagrams, flowcharts. Chapter 4 contains an introduction to the software and the platform used for project programming. Chapter 5 contains the expected results and validation for the project. The last chapter contains a conclusion about our work, then we have a summary, references, appendices.

Chapter two

Chapter two

Background

2.1 Overview

This chapter introduces a theoretical background of the system, some descriptions of the hardware and software components used in the system. Specifications of the design system and constraints are discussed too.

2.2 Theoretical background

At present, most homes depend on water tanks on the roofs of these houses, as it is considered to be the main place to store the water that’s coming from the source to be used when

needed, which can be used for multiple purposes, so the presence of tanks is an essential thing, especially since there may not always be places to establish water wells to store the water inside it.

The project's main idea is to design and implement a smart water tank using an ESP32. Sensors are used to collect data from the tank then send it to ESP32which controls the valves of the tank, after that the ESP32 will send it to the application which will show the data and notifications, it may also allow users to control the valves of the tank.

To achieve the goals of our project, we need several components. To achieve the calculation of the amount of water in the tank, we need the ultrasonic sensor and two flow sensors. A flow sensor is a device to monitor the amount of water being supplied and used, the rate of flow of water has to be measured. This sensor has a plastic valve from which water can pass. A water rotor along with a hall effect sensor has presented the sense and measured the water flow. When water flows through the valve it rotates the rotor. By this, the change can be observed in the speed of the motor. This change is calculated as output as a pulse signal by the hall effect sensor. Thus, the rate of flow of water can be measured.[19]

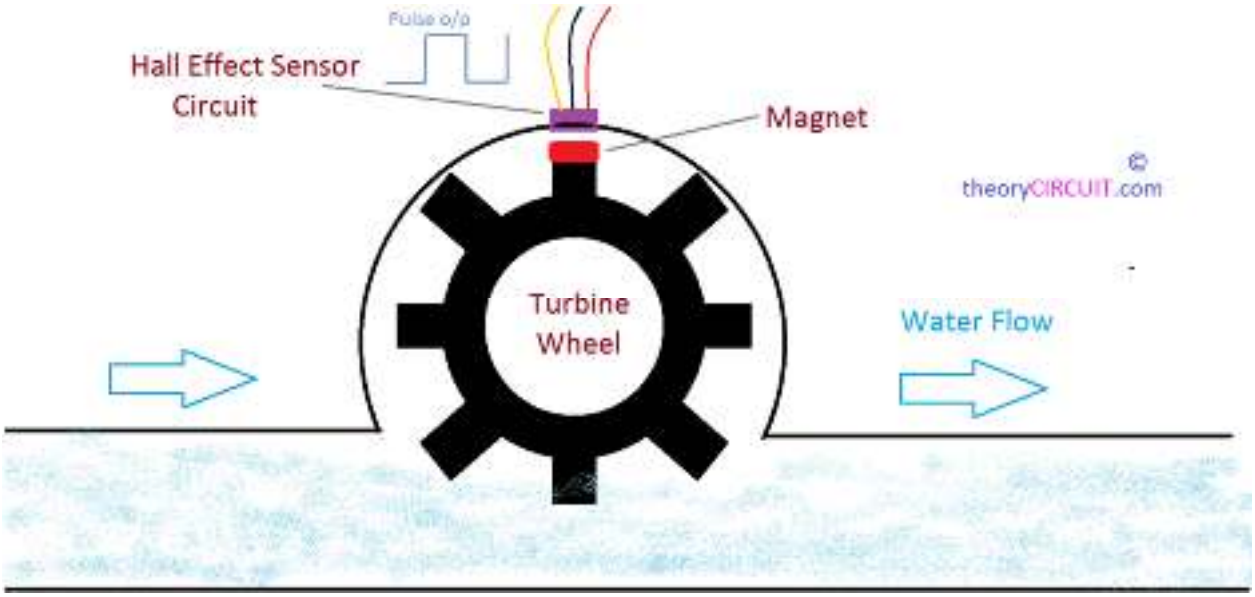

Figure 2.1: Flow sensor operation.[19]

An ultrasonic sensor is a device that can measure the distance to an object by using sound waves as shown in Figure 2.1. It can measure from 2 cm to 400 with a ranging accuracy that can reach up to 3mm. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being

generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.[15]

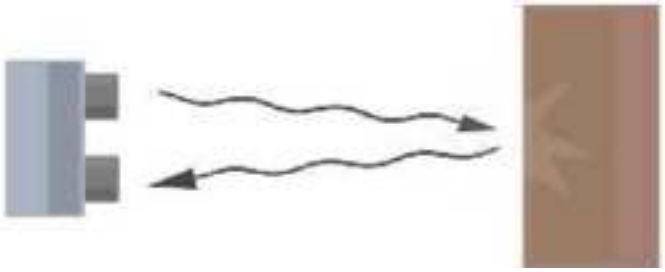

Figure 2.2: Ultrasonic sensor operation.[15]

To find the total round-trip distance of the sound wave. Round-trip means that the sound wave traveled 2 times the distance to the object before it was detected by the sensor (it includes the trip from the ultrasonic sensor to the object and vice versa).[15]

Formula used :

► **D = (S * T) / 2**.

Where D is distance, S is the speed of waves, T is the time taken by waves to fall back.

The idea of using two flow sensors is that needed to put one of them at the place where the water enters the tank, assuming that the name is **F1**, and needed to put the second at the place where the water exits from the tank, assuming that name is **F2** and put the ultrasonic sensor at the top of the tank and assume that its name is **U**. To calculate the water consumption rate, take the value of sensor F2, and to find out the amount of water that arrived from the source take the value of sensor F1. To calculate the amount of water available in the tank, obtain this value by subtracting the value of sensor F2 from the value of sensor F1.

► **amount of water consumption = F2**

► **amount of water that arrived from the source = F1**

► **amount of water available in the tank = F1 - F2**

But to make this system better in dealing with these readings, we want to store the values that we get in an array. We want this system to store the amount of water in the tank at every point in this tank. That is, for example, we store in the array, at a distance of 50 cm from the ultrasonic sensor, 100 liters of water are available. This process may take a long time, but it is better than taking data from sensors each time, as this system initially takes the value of the ultrasonic sensor and searches for it in the field. If it is present, it takes the value of the amount of water stored in the array, and if it is not present, it takes the data from the sensors and then stores it in the array. In this way, and after a period of time,

that is, when storing the amount of water available at each point in the tank, there is no need to take values from the flow sensors and do arithmetic equations.

If we assume that the water tank is a plastic model with a height of 1 meter, then we will place the ultrasonic sensor at the top of the tank, and there will be a valve for the exit of water from the tank at a height of 5 cm from the bottom of the tank. There will be a stopcock for entering the water into the tank 10 cm from the top of the tank, meaning it is 10 cm away from the ultrasonic sensor. If the remaining water in the tank is 7 cm from the height of the tank, meaning that it is 93 cm away from the sensor, the controller will send a signal to close the valve that allows water to exit the tank, and if the water is 12 cm away from the sensor, the controller will close the valve that allows water to enter for the tank.

Water freezes at 0°C, so when the water temperature approaches freezing, that is, when it is close to zero, the controller must give an order to allow water to leak from the pipes by using an external valve, as a result, we need DS18B20 waterproof temperature sensor which is a digital sensor which can reach the digital data resolution up to 12 bits and has ±0.5°C accuracy from -10°C to +85°C. It includes an analog-to-digital converter to convert the analog signal to the digital output with a resolution of up to 12 bits. The communication of this sensor can be done through a one-wire bus protocol which uses one data line to communicate with an inner microprocessor.[16]

To check whether this water is safe to drink or not we need to measure the turbidity of water. Turbidity is the degree or level of cloudiness or haziness of a liquid. This happens due to the presence of large numbers of invisible particles with the naked eye similar to white smoke in the air. When light passes through liquids, light waves get scattered Due to the presence of these tiny particles. The turbidity of a liquid is directly proportional to the free suspended particles, that is if the number of particles increases, turbidity will also increase. Also, it is usually measured in nephelometric turbidity units (NTU). So to measure the turbidity of water in the tank we need a water purity sensor that comes with 3 parts. A waterproof lead, a driver circuit, and a connecting wire. Its detection range is (0-4550NTU), with an error range of ±05%.[17]

The plan is to display the turbidity values from 0 to 4550. That is the meter should display 0 for pure liquid and 4550 for highly turbid ones. So if the turbidity is less than 500, we conclude that the water is clear, and if it is greater than 500 and less than 3000, we conclude that it is cloudy, and if it is greater than that, then the water is turbid.

2.3 Literature review

In this section we will talk about some projects similar to the idea of our project:

- **Water level monitoring**

  This work was done by Areen Nadi Shalaldeh, in "2018". "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 can be used in buildings with multiple floors, houses, and in all the institutions that use the water inlet sources and reservoirs".[10]
  This project only measures the percentage of water in the water tank using the ultrasonic sensor and sends the data to a mobile application, but in our project, we not only measure the amount of water in the tank, but also calculate the water consumption rate, the amount of water that entered the tank, and the water temperature, in addition to measuring water turbidity and controlling the entry and exit of water from the tank using valves.

- **Water level controller**

  The level of the water in a field is controlled using three main components which are the PIC microcontroller, a motor to control the water levels, and a sensor (to sense the level of water when it reaches it). The sensor senses the intensity of water and indicates a time to the microcontroller. The microcontroller produces the control signals to drive the motor. If there is no water then the microcontroller gives a control signal to start the motor and if there is sufficient water in the field then the microcontroller gives a control signal to stop the motor.[18]
  This project controls the water level in the field as if the water runs out from the field, the system sends a signal to the motor to fill the field, and when the field is full, the system sends a signal to the motor to stop. But in our project, we control the entry and exit of water from the tank through the use of valves that allow the entry and exit of water based on the signals it receives from the controller or the user.

Table 1-2.1: previous study VS Our project

	Water level monitoring	**Water level controller**	**Smart water tank (our project)**

Idea	Monitor the level of water	Automatically fill the field with water	Measuring the amount of available water, water turbidity, temperature, consumption rate, and also controlling valves based on emerging cases
Microcontroller	Raspberry pi3	PIC	NodeMCU ESP32
View data remotely	Yes	No	Yes

2.4 Technologies to be used in the project

This subsection illustrates the main technologies to be used in this project and what tasks they do.

- **Arduino programming language**

  Is an open-source computer programming language based on the wiring development platform, the Arduino IDE is based upon the Processing IDE, and it is available in several operating systems, which give us a programing editor with integrated libraries support and a way to easily compile and load our Arduino programs to a board connected to the computer. This language is a framework built on top of C++, the main difference from c/c++ is that you wrap all your code into two main functions, any Arduino program must provide at least two main functions.[1][2]

- **Flutter**

  Flutter is Google's portable UI toolkit for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.[20] We use this technology to build a mobile application to display the data that comes from the ESP32 microcontroller to the user, and this app allows the user to control valves by sending a signal from the app to the microcontroller.

2.5 Hardware components of the system

This part describes the components and devices used in the system.

1. **NodeMCU** **ESP32**

The ESP32 is a dual-core 160MHz to 240MHz CPU, whereas the ESP8266 is a single-core processor that runs at 80MHz. These modules come with GPIOs that support various protocols like SPI, I2C, UART, ADC, DAC, and PWM. The best part is that these boards come with wireless networking included, which makes them apart from other microcontrollers like the Arduino. This means that you can easily control and monitor devices remotely via Wi-Fi or Bluetooth for a very low price.[12]

We will program this controller to deal with the data coming from the connected sensors and send it to the application for display to the user.

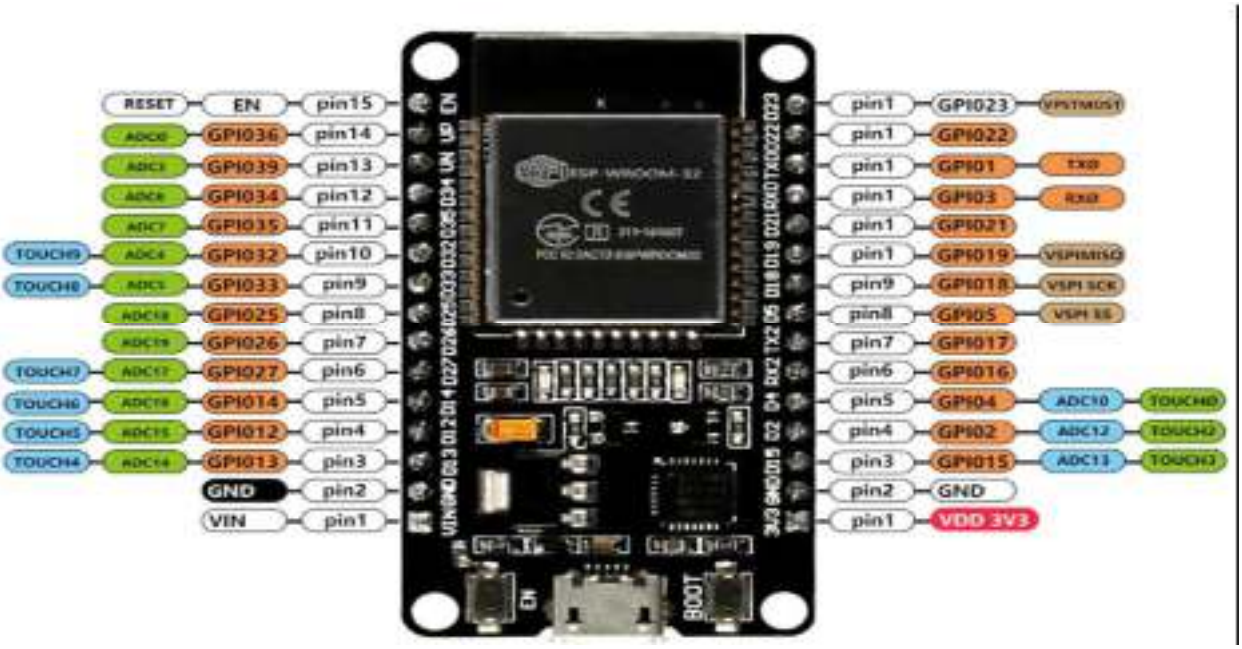

Figure 2.3: NodeMCU ESP32 .[12]

2. **Ultrasonic sensor**

The ultrasonic sensor measures distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected from the target. Ultrasonic sensors measure the distance to the target by measuring the time between the emission and reception.[4]

We will use this sensor in this project to calculate the amount of water in the tank, as we will place it at a point at the top of the tank and it will measure the distance between it and the level of water in the tank. This sensor sends the data obtained to the ESP32, where it will be connected to it.

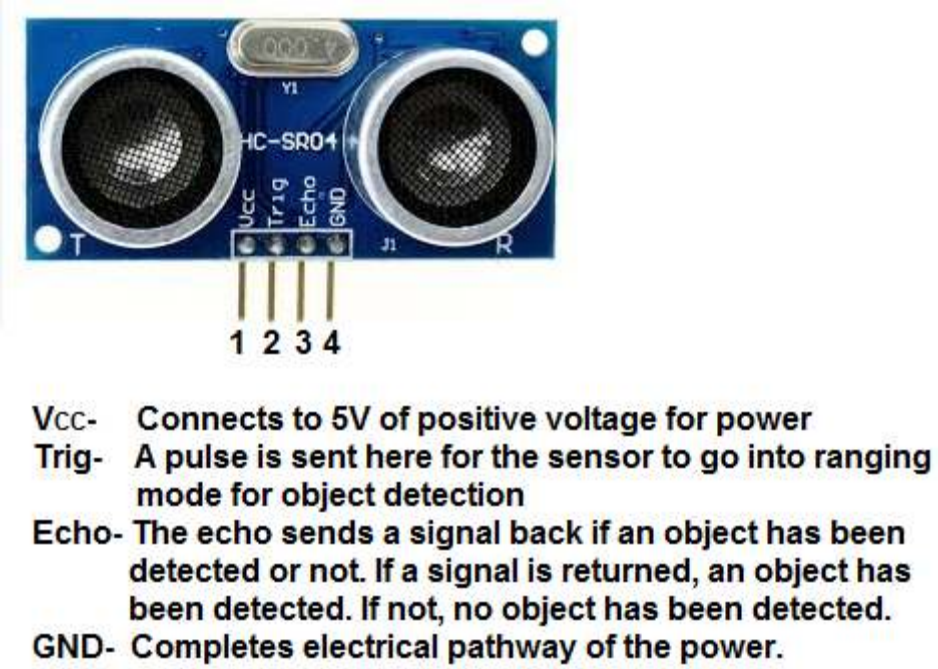

Figure 2.4: Ultrasonic sensor [13]

3. Solenoid valve

Solenoids are very commonly used actuators in many process automation systems. Solenoid valves are used wherever fluid flow has to be controlled automatically. These valves are control units that, when electrically energized or de-energized, either shut off or allow fluid flow. The actuator takes the form of an electromagnet. We use these valves to open or close tank pipelines.[6]

We will put two of these valves in the tank. We will place the first one at the point of entry of the water into the tank, where we will control the process of entering the water coming from the source to the tank through it. We will place the second valve at the point of exit of the water from the tank to control the process of exit of water from the tank to the consumer through it.

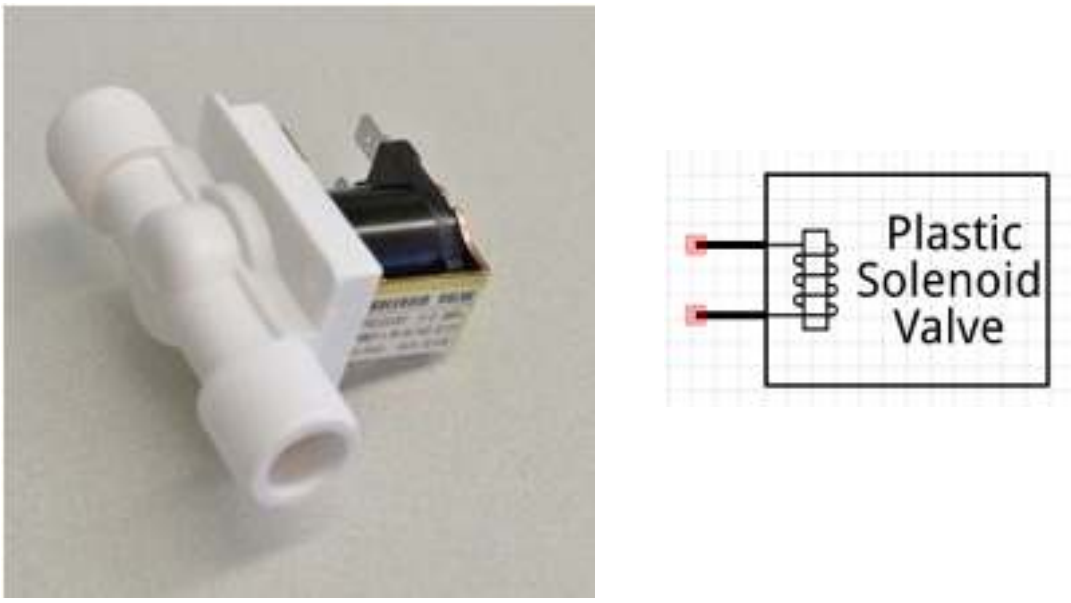

Figure 2.5: Solenoid valve [6]

4. **Water temperature sensor**

This is a highly sensitive water temperature sensor, small temperature delay, waterproof, Designed with a pluggable terminal adapter so that DS18B20 temperature sensor could connect to a pull-up resistor to use, and this temperature range is from -10℃ to +85℃ with error ±0.5℃.[7]

We will use this sensor to measure the temperature of the water so that we can tell the user to make a water leak when the water is nearing freezing so that the water does not freeze inside the water pipes, which may lead to damage.

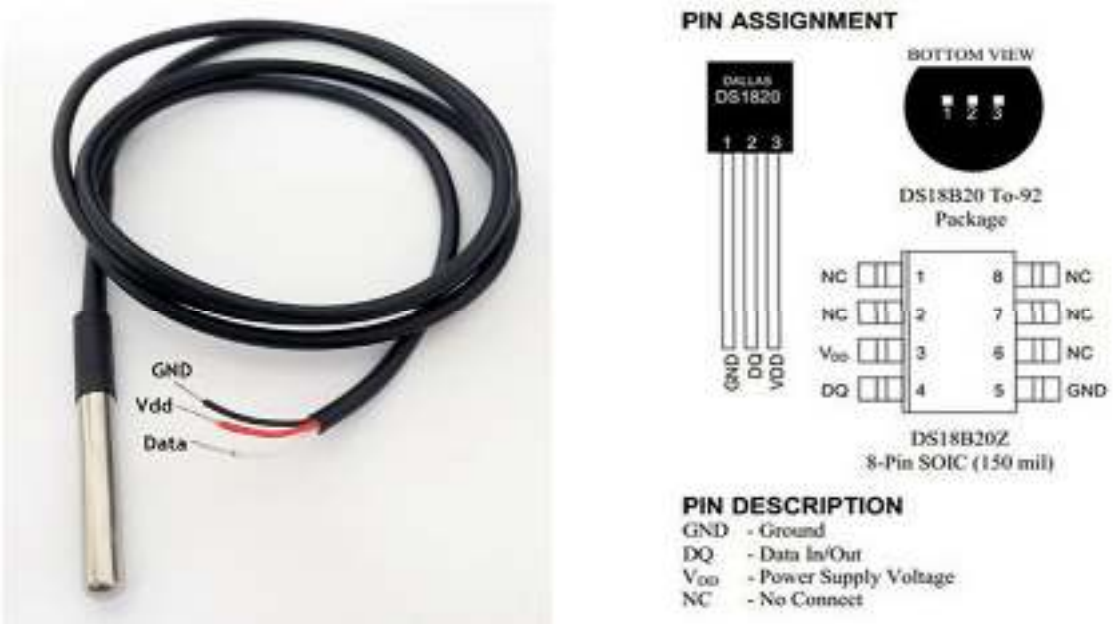

Figure 2.6: DS18B20 Water temperature sensor board transducer module[7]

5. Water flow sensor

The water flow sensor consists of a plastic valve body, a water rotor, and a hall-effect sensor. When water flows through the rotor, the rotor rolls. Its speed changes with different rates of flow. The hall-effect sensor outputs the corresponding pulse signal. The flow rate range for this sensor is from 1 to 60L/min, and storage temperature between -25℃ ~ +80℃.[19]

We want to use this sensor to calculate the rate of water consumption in a certain time by calculating the amount of water leaving the tank and the amount of water entering it.

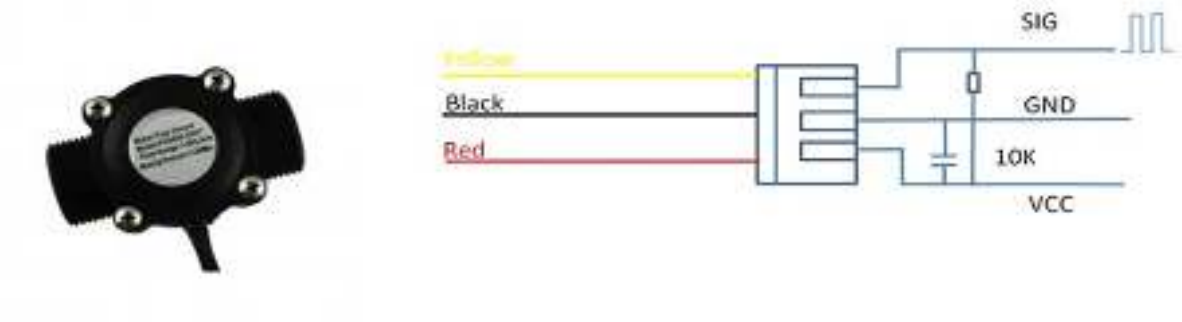

Figure 2.7: Water flow sensor

6. **Water purity sensor**

We use a water purity sensor, which includes a PH sensor and turbidity sensor when checking the turbidity of water. And when these sensors get interfaced with water, they will generate PH value, purity factors, and the temperature of the water as its output.[8]
We use this sensor to measure the percentage of turbidity of the water, when the percentage increases, we will close the tank and prevent the water from escaping from the tank to the consumer.

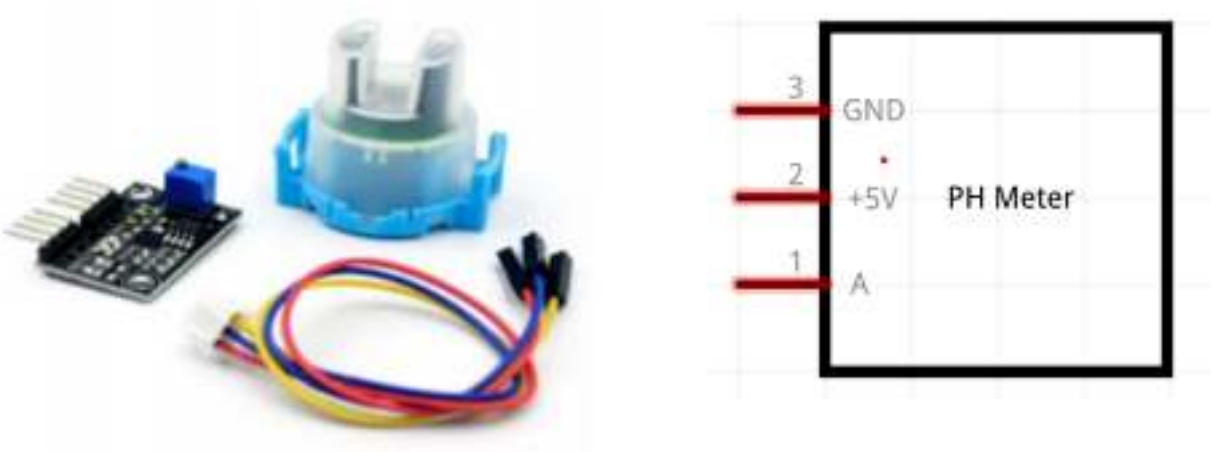

Figure 2.8: Water purity sensor [11]

2.6 Design alternatives.

This subsection illustrates the alternative for NodeMCU ESP32 and explains why we don't use it.

- **Raspberry pi**

Raspberry Pi is a single-board computer. It is a credit-card-sized computer with low cost, which plugs into a computer monitor or TV and to operate it, a user can use a standard keyboard and mouse to operate it. The single-board consists of a fully

functional computer with its dedicated memory, processor and it runs an operating system.[8]

This table shows the main difference in specifications for ESP32 and Raspberry Pi 3.[8]

Table 2-2.2: ESP 32 VS Raspberry Pi 3

FEATURE	RASPBERRY PI	ESP 32
Core count	Dual-core	Single/dual-core
Microcontroller	RP2040	Tensilica Xtensa LX6
Clock frequency	48kHz / 133kHz	160kHz / 240kHz
Internal Flash Memory	2 MB	4 MB
SPI	2	4
I²C	2	2
PWM	16	16
ADC	3(12-bits)	18(12-bits)
GPIO(total)	26	34
UART	2	3
Programming Language	MicroPython, C, C++	C, MicroPython with limited support
WiFi	Not Supported	802.11 b/g/n
Bluetooth	Not Supported	V4.2 (Supports Classic Bluetooth and BLE)
Ethernet	Not Supported	10/100 Mbps
Price	$4	$4 – $12

- **ESP 8266**

  All ESP8266 variants have an ESP8266EX core processor and a Tensilica L106 32- bit microcontroller unit. This is a low cost, high performance, low power consumption, easy to program, wireless SoC(System-On-Chip).[21]

This table shows the main difference in specifications for ESP8266 and ESP32.[21]

Table 3-2.3: ESP8266 VS ESP32.

FEATURE	ESP8266	ESP32
MCU	Xtensa Single-core 32-bit L106	Xtensa Dual-Core 32-bit LX6 with 600 DMIPS
802.11 b/g/n Wi-Fi	HT20	HT40
Bluetooth	X	Bluetooth 4.2 and BLE
Typical Frequency	80 MHz	160 MHz
SRAM	X	✓
Flash	X	✓
GPIO	17	34
Hardware /Software PWM	None / 8 channels	None / 16 channels
SPI/I2C/I2S/UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	X	✓
Ethernet MAC Interface	X	✓
Touch Sensor	X	✓
Temperature Sensor	X	✓(old versions)
Hall effect sensor	X	✓

Working Temperature	**-40ºC to 125ºC**	**-40ºC to 125ºC**
Price	**$ (3$ - $6)**	**$$ ($6 - $12)**

- **Arduino mega**

  Arduino Mega 2560 is a microcontroller board based on the ATmega2560, it is designed for projects that require more I/O lines, more sketch memory, and more RAM. Connect the board to the computer using the USB cable to load the program and power it up.[3]

  This table shows the main difference in specifications for Arduino Mega 2560 and ESP32 NodeMCU.[9]

Table 4-2.4: ESP32 NodeMCU VS Arduino mega

FEATURE	ESP32 NodeMCU	Arduino mega
Price	$11	36$ - 39$
Processor	ESP32	Atmega 2560
Clock speed	80 MHz / 160 MHz	16MHz
Digital I/O Pins	36	54
Digital I/O Pins with PWM	36	15
Analog Input Pins	15	16
SPI/I2C/I2S/UART	4/2/2/2	1/1/1/4
WIFI	yes	no
Ethernet MAC Interface	yes	no
Bluetooth	yes	no

2.7 Design constraints.

1. Permanent need for the internet.
2. The height of the tank should not be more than 4 meters.

3. This system works for one tank.

Chapter three

Chapter three

System design

3.1 Overview

The following section has a description of the system, detailed design, and necessary information about the design.

3.2 Detailed description of the system

This system will be designed to track the water tank and detect any abnormalities. When the system is activated, The ultrasonic will measure the distance between it and the surface of the water, and then it will send it to the ESP32, as if the amount of water in the tank is nearing completion, the ESP32 will close the valve of the water tank, which allows the water to exit the tank, informing the user of the status Also, the temperature sensor will measure the temperature of the water inside the tank as if the temperature is close to freezing, The controller will close the valve that allows water to escape from the tank, then it will allow the water inside the pipes extending from the tank to be drained to the house through an external valve located at the closest point to the house.

There is also a water purity sensor that will check the water turbidity, as if the water turbidity is poor, the ESP32 will send a signal to the tank valves that allow water to enter and exit the tank until it is closed, informing the user of the status. Two flow sensors will calculate the rate of water consumption, the amount of water that entered the tank, and also the amount of water available in the tank. These sensors all send the data that you get to the ESP32 to deal with it and send it to the user.

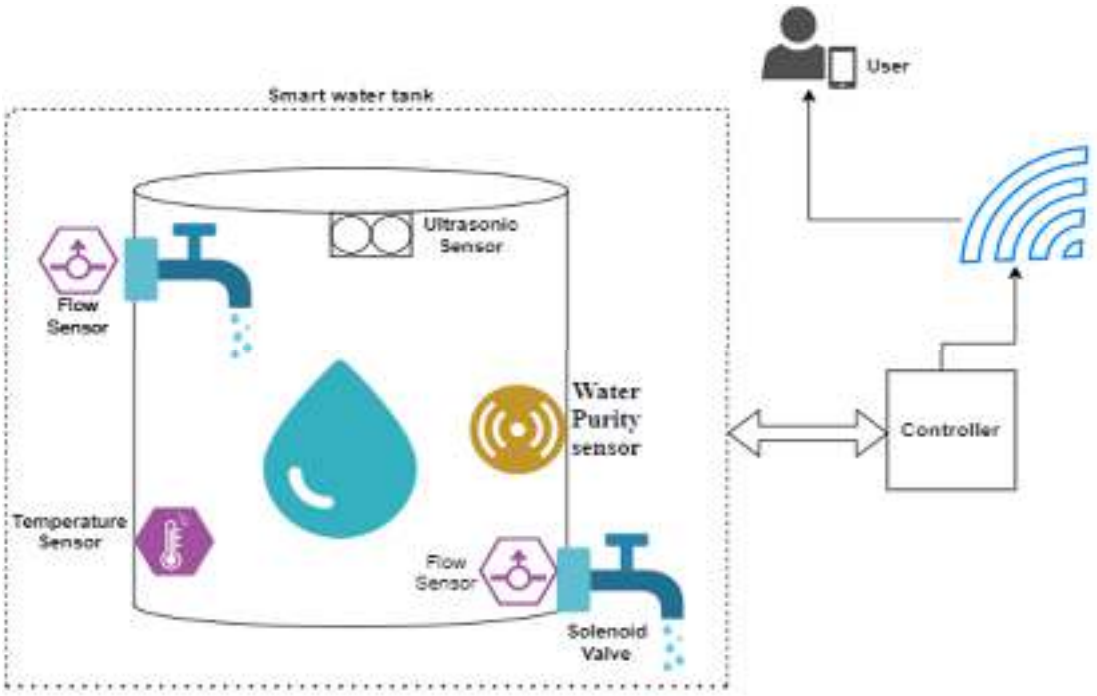

Figure 3.1: Graphical representation of smart water tank

3.3 System diagrams

In this subsection, three diagrams are represented for the understanding of the project concepts and design.

3.3.1 System block diagram

Figure 3.2 is the general block diagram of the project. As illustrated below, the sensors when the system is activated will collect the necessary data and send it to the ESP32. After the ESP32 gets it, it will examine this data and process it. Whereas, if the data obtained from the purity sensor is not within the normal rate of the water turbidity ratio, it will send a signal to the valves and close them. Also, if the data obtained from the temperature sensor indicates that the temperature is close to freezing, then it will send a signal to the valve that allows water to exit from the tank and allows water to leak so that it does not freeze inside the pipes. The data that he will obtain from the ultrasonic sensor and flow sensors will do some mathematical operations on it, and then if the percentage of water in the tank is close to running out, it will send a signal to the valve that allows the water to exit from the tank and close it. Then it will send all the data to the user.

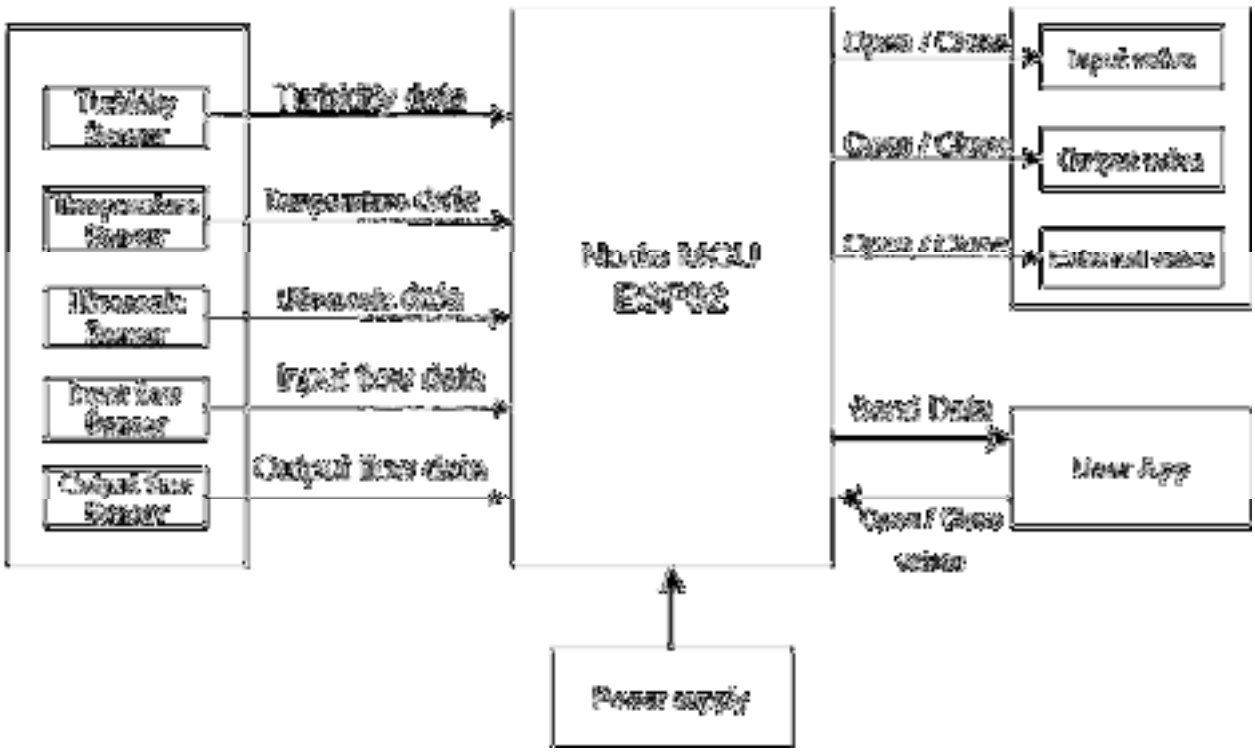

Figure 3.2: General block diagram of smart water tank

3.3.2 System demo

Figure 3.3 is the system prototype diagram of the project. This image is an illustration of what we want to design in this project to achieve the objectives of this project, as we note in the image below that we have three water tanks. The tank on the left represents the water that comes from the source, as the water will move from this tank to the main water tank in the middle, which contains the components of the project, and on the right is the tank that represents the place where the water will go, meaning that it represents the water consumer.

The data will be taken from the sensors in the tank located in the middle, where we also have three valves. A valve is located at the entrance to the tank and the other at the exit, and the last valve is located somewhere in the pipe extending from the tank to the consumer (home), which we will use to empty the water from the pipes when the temperature approaches freezing, after closing the valve that allows water to exit from the tank.

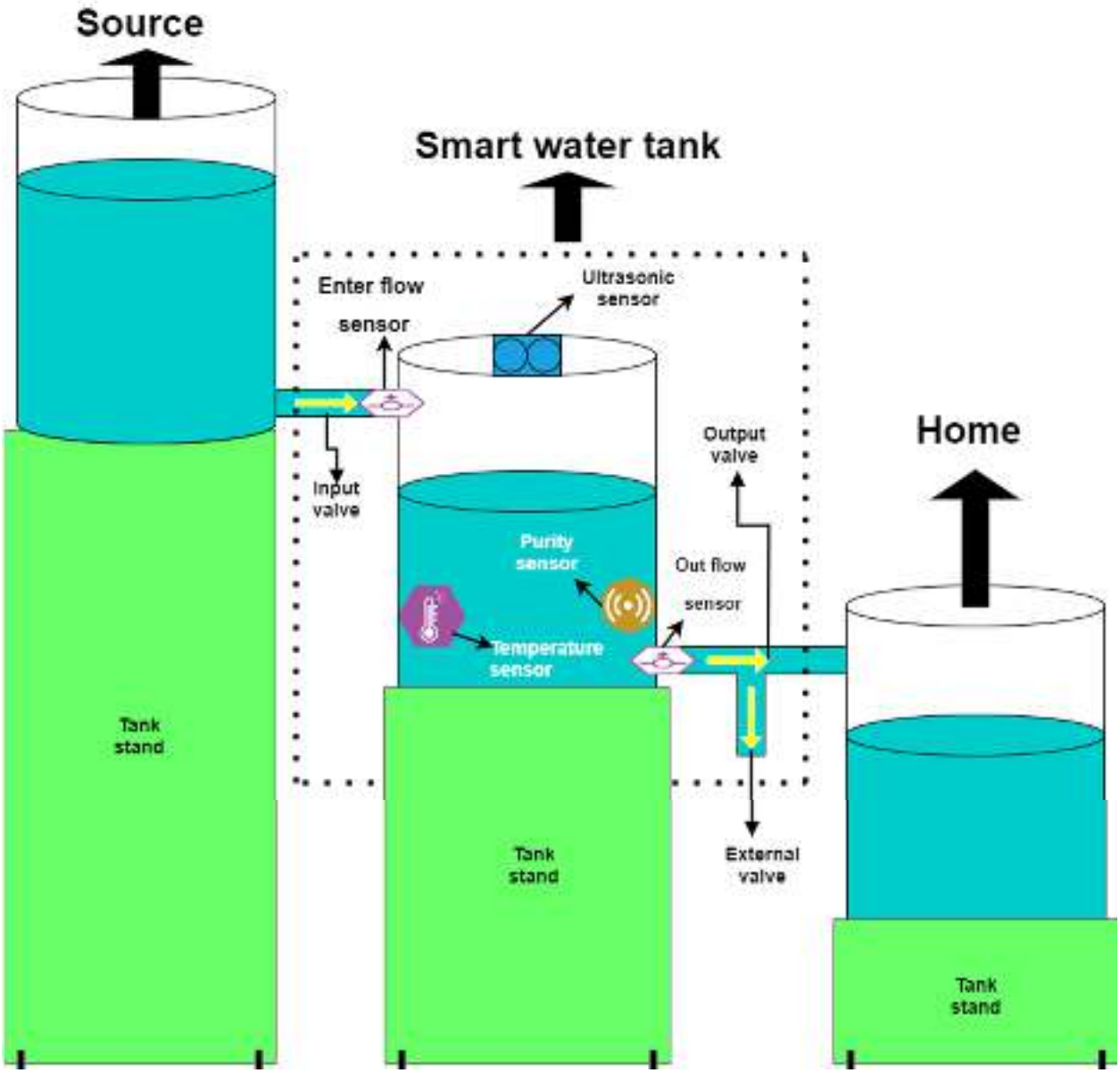

Figure 3.3: System prototype of smart water tank

3.3.3 Schematic diagram

Figure 3.4 figure describes the system elements and how they are connected.

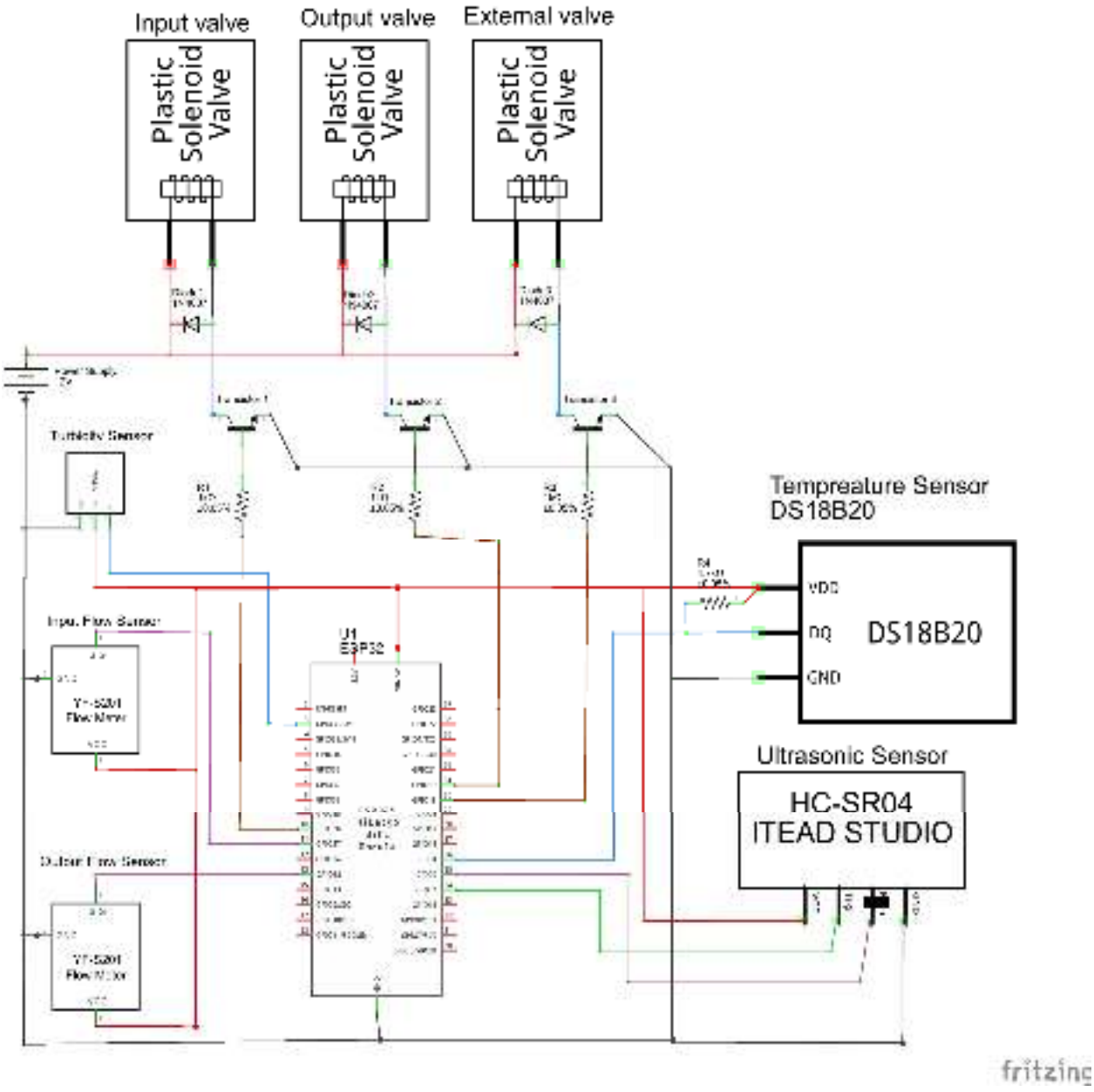

Figure 3.4: Schematic diagram of smart water tank

1. **ESP32:** The main component in the system, which will be linked to the other components. Represent the inlet and outlet of the signals and information. It will also control the valves connected to it based on the data it receives from the senators connected to it, and it will also send the data to the cloud to display on the mobile application.

2. **Turbidity sensor:** It is used to find the water turbidity rate, as if the percentage is high, it will send a signal to the ESP32.

3. **Ultrasonic sensor:** It is used to measure the distance between it and the surface of the water, and if it is close to running out or to fill, it sends a signal to the ESP32.

4. **Temperature sensor:** It is used to measure the temperature of the water, as if it is close to freezing, it will send a signal to the ESP32 to allow water to leak.

5. **Valves:** It is used to allow water to enter and leave the tank based on the signal coming to it from the ESP32.

6. **Flow sensors:** It is used to calculate the rate of water consumption, the amount of water that entered the tank, and also the amount of water available in the tank.

3.4 Flowcharts

This diagram shows the flowchart of the system:

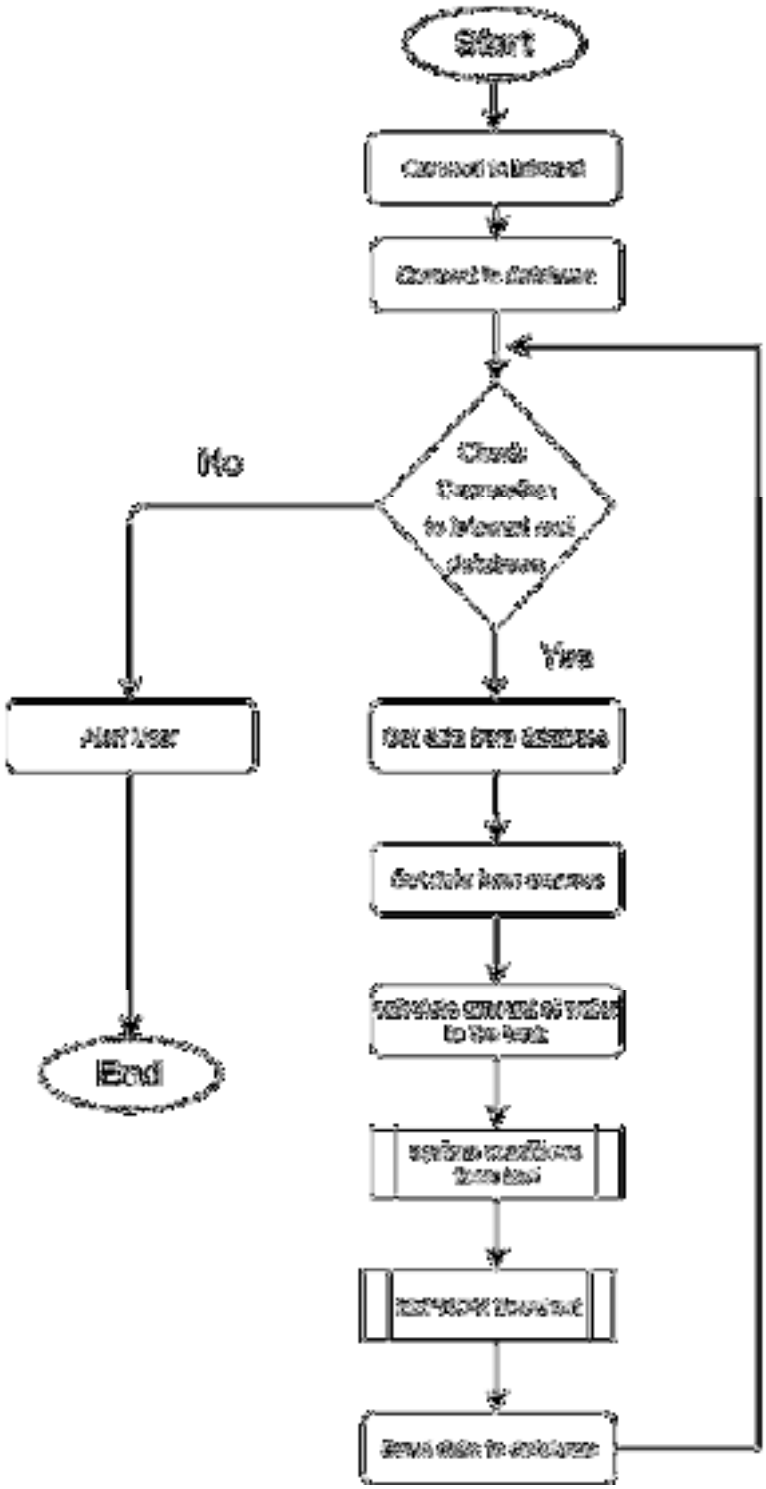

Figure 3.5: Flowchart of smart water tank system

This diagram shows the flowchart of the system conditions:

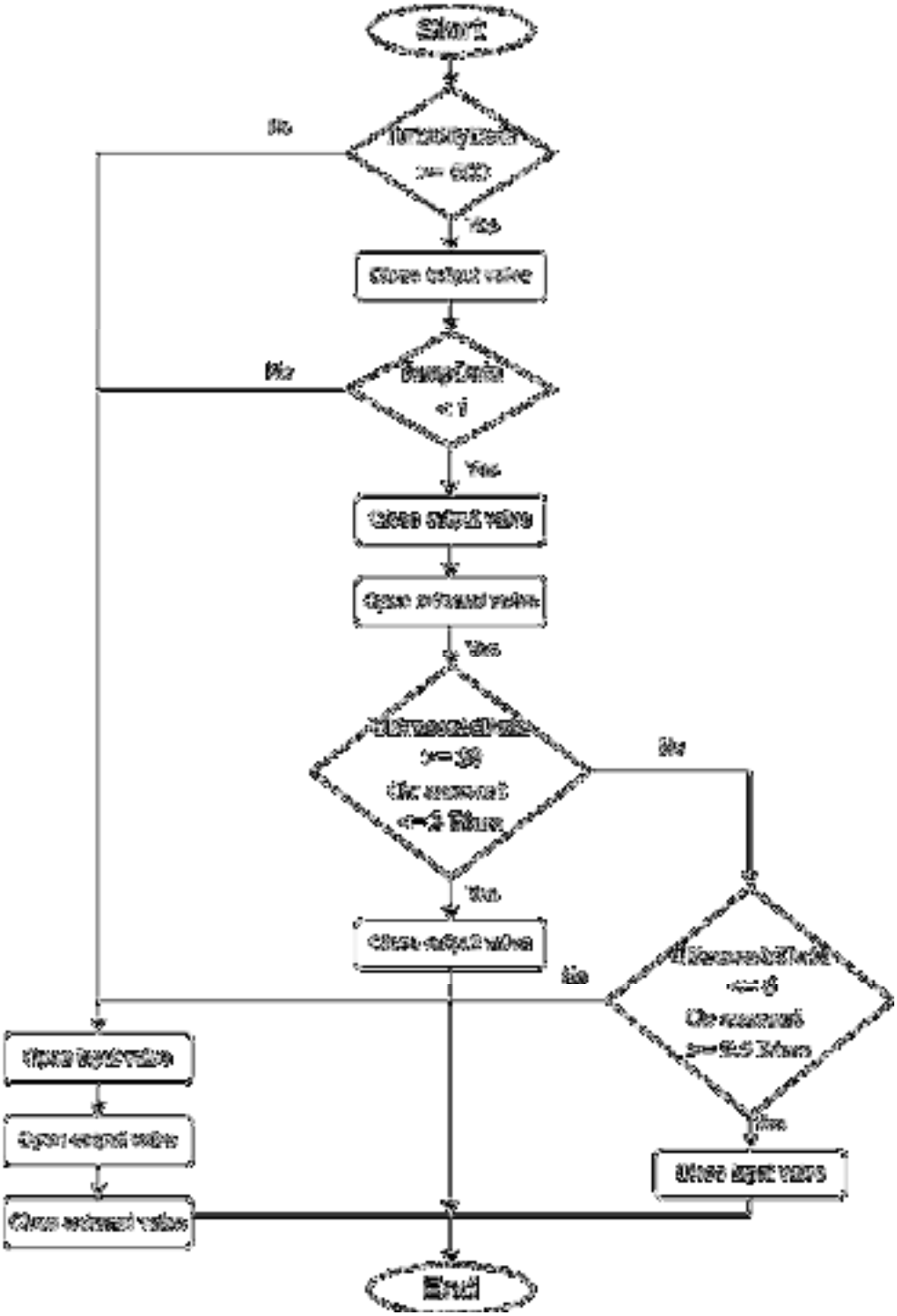

Figure 3.6: Flowchart of system conditions

This diagram shows the flowchart of the read and writes data on EEPROM:

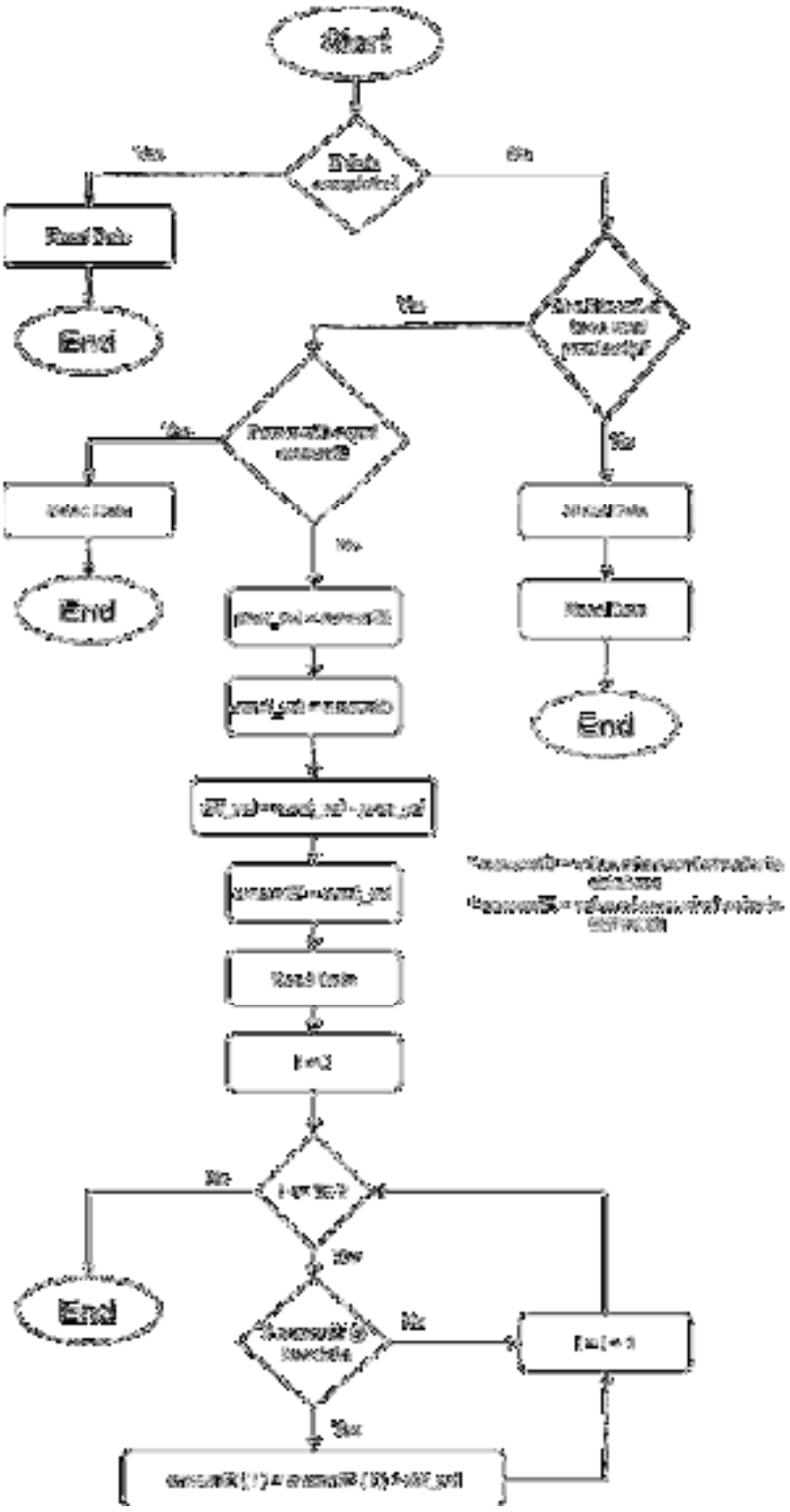

Figure 3.7: Flowchart of EEPROM data

3.5 Pseudocode of the system

The following subsections have a description of the pseudocode used in the system:

```
connect to internet
connect to database
while connecting to the internet and database
   get data from the database
   get data from sensors
   calculate the amount of water in the tank
   if turbidity data greater than or equal 500
         Sending a signal to input and output valves to be closed
         AlertUser that the water is turbid
   if turbidity data greater than 50 OR turbidity data less than 500
      AlertUser that the water is close to turbidity (cloudy)
   if temperature data is less than 1
      Sending a signal to the output valve to close
      Sending a signal to the external valve to open
      AlertUser that the water is too close to freezing.
   if ultrasonic data greater than or equal 30 OR amount less than or equal 2
      Sending a signal to the output valve to be closed
      AlertUser that the water is close to running out of the tank
   if ultrasonic data less than or equal 6 OR amount greater than or equal 9.5
      Sending a signal to the input valve to be closed
      AlertUser that the tank is full
   // EEPROM
   if data completed
      Read data
   else
      if this data readed previously
         if the amount in the database equals the amount in EEPROM
            Read data
         else
            update this value with a difference
            update each readed data previously with the difference
```

3.6 Detailed design

In this section, we will talk about Hardware and Software be used in our

A project that summarized below:

3.6.1 Hardware

1. **NodeMCU we will choose ESP32, specifications for this:**
   ESP32 is a low-cost, low-power system on a chip (SoC) series with Wi-Fi & dual-mode Bluetooth capabilities. At its heart, there's a dual-core or single-core Tensilica Xtensa LX6 microprocessor with a clock rate of up to 240 MHz. ESP32 is highly integrated with built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. Engineered for mobile devices, wearable electronics, and IoT applications, ESP32 achieves ultra-low power consumption through power-saving features including fine resolution clock gating, multiple power modes, and dynamic power scaling.[22]

(Arduino mega 2560, [22])

3.6.2 Software

- **Arduino**
  The Arduino Integrated Development Environment - or Arduino Software (IDE) - makes it easy to write code and upload it to the board. This software can be used with ESP32 boards. And this IDE 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 Genuino hardware to upload programs and communicate with them. And the IDE.

(Arduino IDE logo, [2])

Chapter four

Chapter four

System implementation, system testing, and discussion

4.1 Overview

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

4.2 Description of the implementation

- Was purchased the necessary sensors for the project, the plastic containers that represent the tanks in this project, and the stands for the tanks.
- Assembled the valve that enables water to enter the tank from the source with the input flow sensor, and then arranged them between the tank representing the source and the tank representing the main tank.
- Assembled the valve that allows the exit of water from the tank with the output flow sensor with the external valve, and then we combined them between the main tank with the tank that represents the water consumer.
- The sensor power lines (VCC) were connected to the breadboard, and the necessary voltage was connected to them from the controller, and we did the same for the ground.
- The sensors' signals were directly connected to the microcontroller by wires.
- The wires from the three valves were attached to corresponding electronic circuits connected to the breadboard, and then the signal from each circuit was connected to the controller.
- Wires and parts were arranged to do the necessary tests.

4.3 How does the system works

- The ESP32 connects to the Internet and then connects to the database.
- The ESP32 fetches the previous state of the valves and also the value of the previous water quantity from the database.
- Get data from each sensor.
- The amount of water available in the tank is calculated.
- It is ensured that the remaining amount of water is currently stored in the EEPROM, by taking the value of the distance, which was considered as the index, and checking whether it has a value or not. If it has a value, it is ensured that this value is equal to the value that was calculated or not to be updated and then obtained, and if it does not have a value, a

value is stored in this place and we take this value. This process continues until all values that can be obtained at each tank space are stored.

- The valves that allow the entry and exit of water from the tank are controlled based on the conditions loaded on the controller, which depend on the values obtained from the sensors. Then the status of each valve is sent to the database.
- All values obtained from the sensors, as well as the calculated values, are sent to the database.
- When the application runs, it fetches data from the database and displays it to the user through several interfaces.
- The application displays the data to the user and interprets it for him.
- The user can control the valves on the tank through the application, and if these valves are controlled by the user, the status of these valves is sent to the database so that the controller can obtain them and change their status.
- This process continues as long as the Internet connection is not lost.

4.4 Implementation issues and challenges

- The valve needs to be connected to a voltage of 5 volts from the controller to operate. And the controller gives us only 3.3 volts: we solved this problem by connecting the valve, an electronic circuit with a diode, a transistor, and a resistor so that we can provide 5 volts for the valve.
- Sometimes there is a problem in the process of calculating the amount of water through the values obtained from the database: We solved this problem by storing the values in the EEPROM.

4.5 Validation result

4.5.1 Hardware testing

A NodeMCU ESP32 was tested with all sensors and connected directly to the laptop as well as the 12V power supply for the valves.
This image represents the project that was built in the university lab and that we did the tests on.

Figure 4.1: Hardware Components

This picture represents the assembly of wires on the breadboard and the controller.

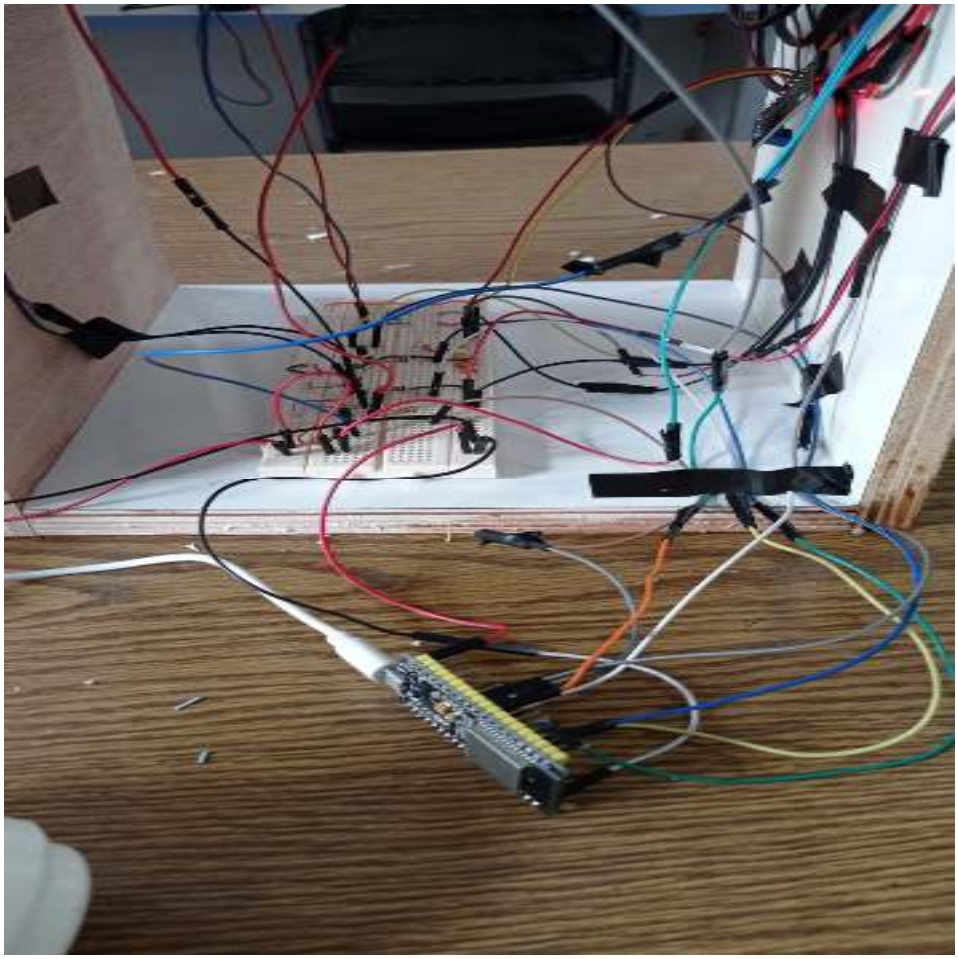

Figure 4.2: Hardware Components with NodeMCU ESP32

The data is stored in the Database in the fields shown in the following figures on the Firebase Realtime Database website:

JSON Data


```
{
  "Dataa" : {
    "amountArr" : 2.846,
    "amount_l" : 2.846,
    "amount_m" : 2846,
    "amount_perc" : 28.459999561,
    "disArr" : 22,
    "distance" : 22,
    "external_valve" : 0,
    "in_flow_l" : 3.856,
    "in_flow_m" : 3856,
    "in_flow_rate" : 0,
    "input_valve" : 1,
    "out_flow_l" : 1.01,
    "out_flow_m" : 1010,
    "out_flow_rate" : 0.62439,
    "output_valve" : 1,
    "temp_c" : 23.625,
    "temp_f" : 74.525,
    "turbidity_ntu" : 456,
    "turbidity_v" : 4.46
  }
}
```

Figure 4.3: All data Query

4.5.2 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.

Table 5-4.1: Unit Test

#	Case	Expected Output	Obtained Output	Pass/Fail
1	Connect to the internet	Connect to the internet	Connect to the internet successfully	Pass
2	Get necessary data from the database	Get data from DB	Get data successfully	Pass
3	Get data from sensors	Get data from all sensors	Get data successfully	Pass
4	Get and store data from EEPROM	Read and write data from EEPROM	Read and write successfully	Pass
5	Send data to DB	Send data to DB	Send data successfully	Pass

Table 6-4.2: System Test

#	Case	Expected Output	Obtained Output	Pass/Fail
1	Close input valve if the tank is full	Close input valve if the tank is full	Close input valve successfully	Pass
2	Close output valve if the tank is empty	Close output valve if the tank is empty	Close output valve successfully	Pass
3	Close output valve if water is turbid	Close output valve if water is turbid	Close output valve successfully	Pass
4	Close output valve	Close output valve if	Close output valve	Pass

	if the temperature is low	the temperature is low	successfully	
5	Open external valve if the temperature is low	Open external valve if the temperature is low	Open external valve successfully	Pass

Table 7-4.3: Application Pages Test

#	Case	Expected Output	Obtained Output	Pass/Fail
1	API returns data	show data in a user interface	show data successfully	Pass
2	Show current tank status	show data in the user interface	show data successfully	Pass
3	Show if there is a problem	show text in a user interface	show data successfully	Pass
4	API does not return data	Show circle progress in a user interface	Show circle progress successfully	Pass

Table 8-4.4: Application Valves control Page Test

#	Case	Expected Output	Obtained Output	Pass/Fail
1	API returns the status of valves	show status of valves in a user interface	show status successfully	Pass
2	Change status of valves	change status in user interface and DB	change status successfully	Pass
3	Control status of valves	control valves status in user interface and DB	Control valves status successfully	Pass

The app was developed using Cross-Platform and the Flutter framework, as it is an excellent tool for implementing mobile applications, whether, for Android or iOS. We also built an API technology to link Firebase real-time databases with the app.

When the user opens the application, the program directs the user to the home page, which contains data on water quantity, shows the status of the tank, shows note if there is a problem, shows the value of temperature, and turbidity, issues, valves buttons that send the user to there pages. On the turbidity page, users can see the value of water turbidity with its interpretation of whether it is good or not. On the issues page, the user can see whether there are problems in the tank or not, as if there are problems, it displays all of them. On the valves page, the user can view the current state of the valves, and can also control them in terms of closing or opening them. And when the user clicks on the value of the amount of water available in the tank, the application will take him to another page that shows the amount of water that entered the tank and the amount that came out of it, and also shows if there is currently a consumption of water or not. It also displays if there is water reaching the tank at present. Here are some screenshots were taken while booting the system and reading the data:

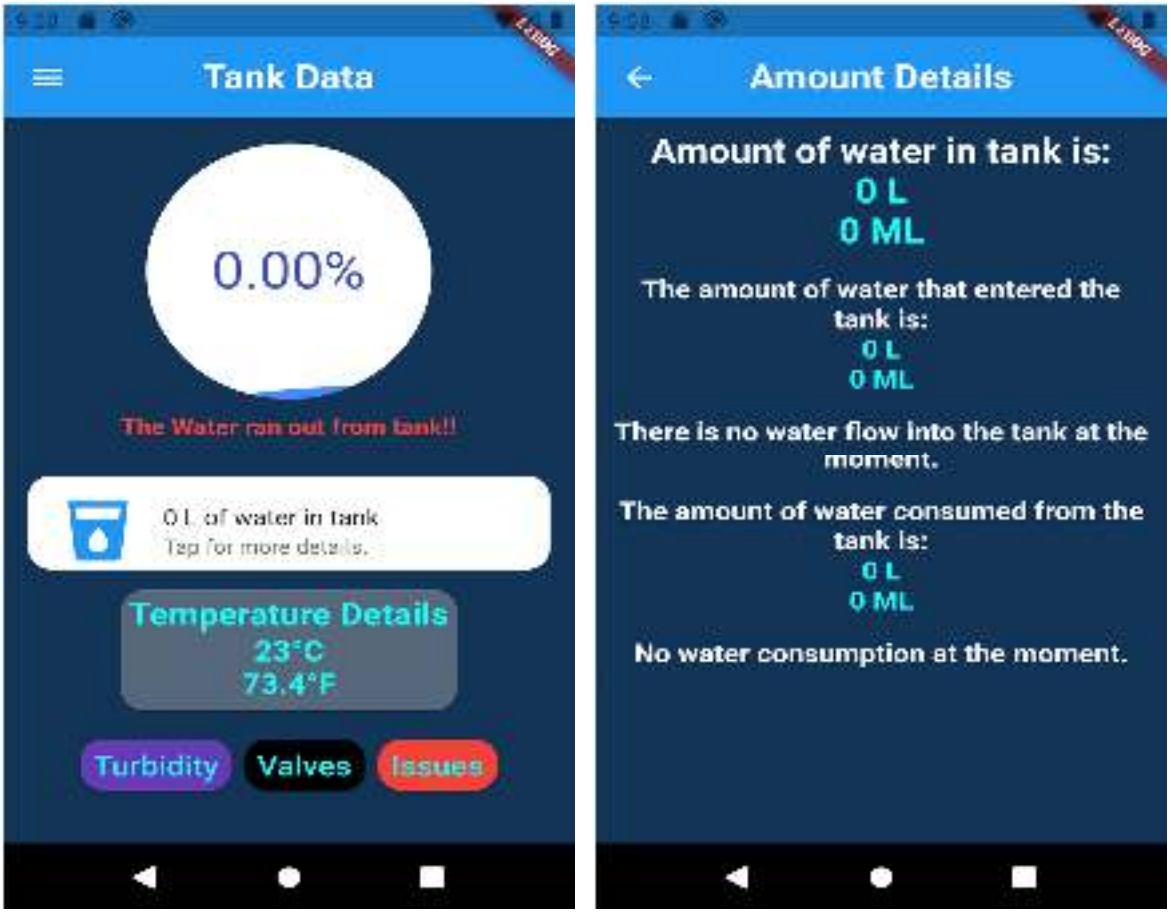

Figure 4.4: Screenshot that shows the data at first

This screens shows the data that displays to the user:

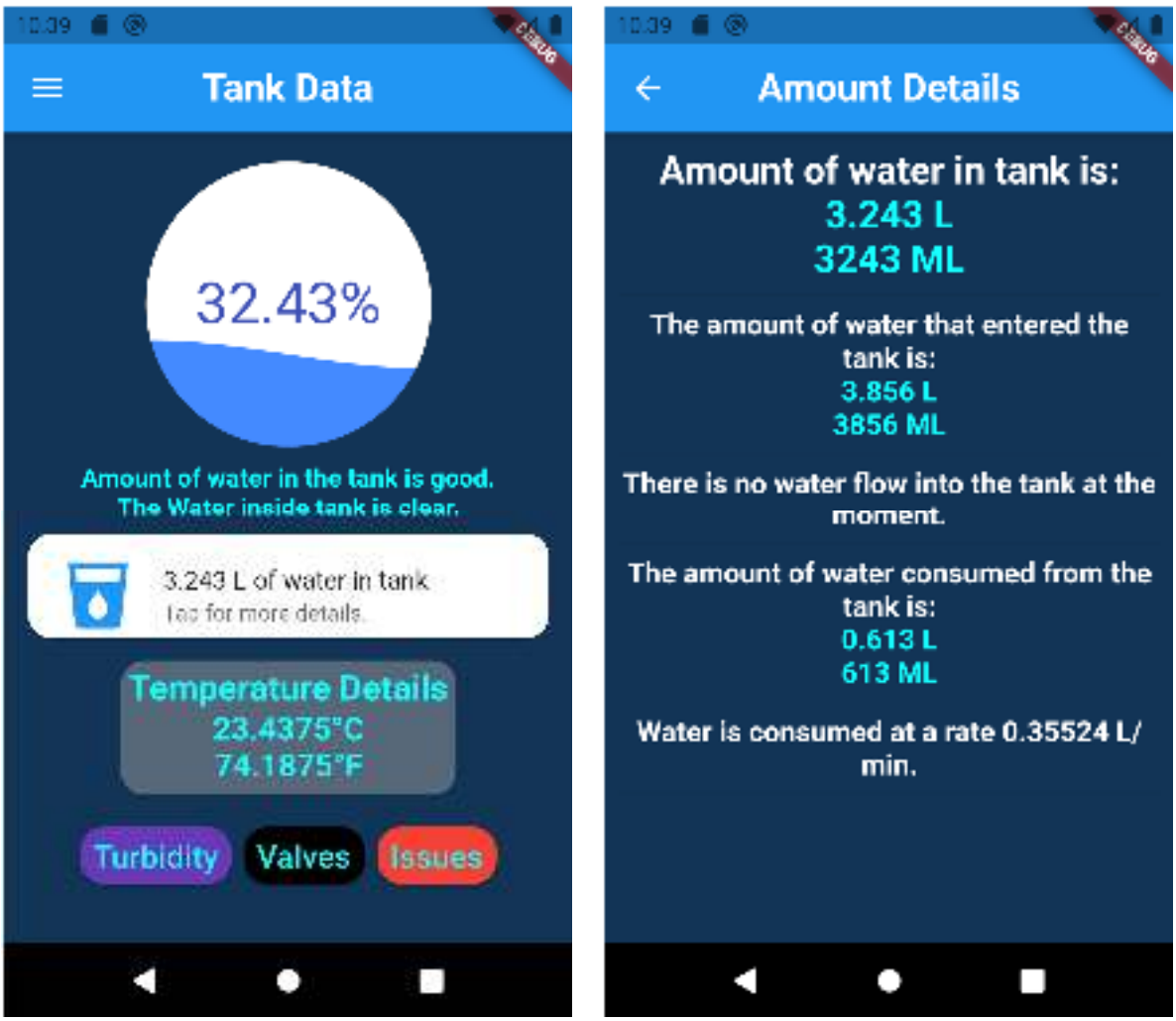

Figure 4.5: Screenshot that shows the data after a while

This screen shows the status of valves in different cases and the user can control them:

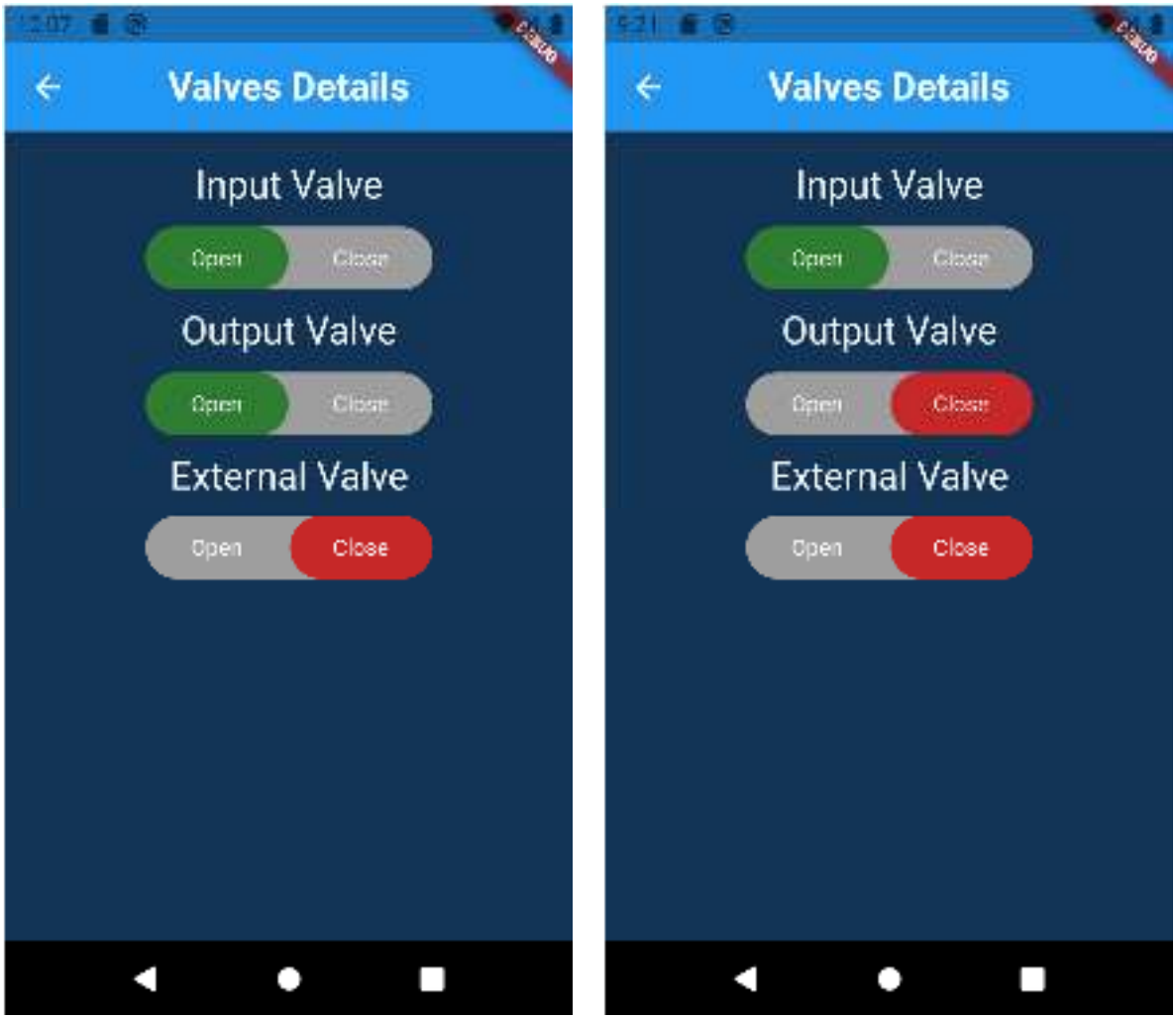

Figure 4.6: Screenshot that shows the valve's status in different cases

This screen shows if there is an issue in different cases:

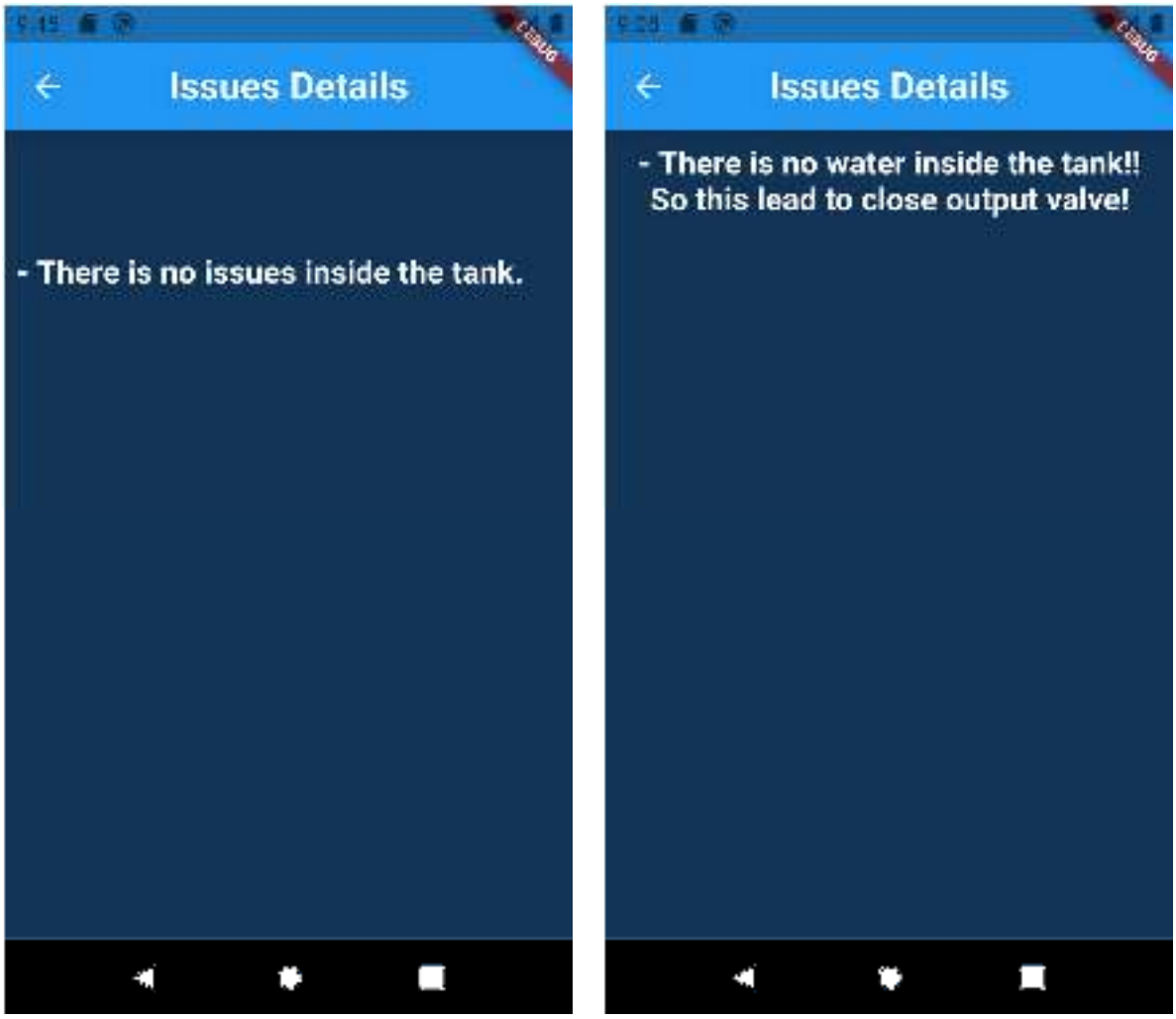

Figure 4.7: Screenshot that shows the status of the issue in different cases

4.6 Discussion

In the beginning, there were several goals in this project to solve many problems. Work has been done to achieve this through this system, which consists of many sensors and solenoid valves. Where these components have been programmed to achieve the requirements of this system, and after building this system and doing the necessary tests to ensure that it works in the required manner and whether it achieves the goals of its construction or not, we can say that it achieved the desired goals and that it worked as required for a simple system. As this system took the data from the sensors correctly and the controller processed it and performed the calculations on the obtained values and controlled the valves based on the obtained values. The process of sending and receiving data from the database was done correctly, and it was displayed and controlled as required from the mobile application.

Chapter five

Chapter five

Conclusion

5.1 Overview

Tack in this part about the summary of all parts, future directions, and future work.

5.2 Conclusion

A water tank system can solve many of the problems facing people, such as checking the quality of water, which if it is bad, affects the health of individuals, and solving the problems that may occur in the water tank in terms of air withdrawal or explosion of pipes at low temperatures, and also to calculate the amount of available water In the tank, which saves a lot of time and effort and eliminates the danger that may occur when manually checking the amount of water remaining in the tank, especially in tall buildings. In this system, users can view all their data through the phone application and from wherever they are, and they can control the valves on the tank. The control also deals with some things that may lead to damage or losses automatically, which may sometimes occur without the knowledge of the user, which reduces time, effort, and cost.
This system has been built and verified to work as required to achieve the requirements of this project, and it was found that it works as required and meets the needs of this project.

5.3 Future work

In the future, we look forward to adding important features to the system, the most important of which are:

- Develop the system to deal with more than one tank.
- Developing the system to deal with pumps and motors for pumping water from wells in places when water runs out of the tank.
- Connecting the system with the Water Authority to know the amount of water that this user has used and the rate of its consumption, as well as to supply the amount of water he needs.

References

[1].Hackerearth. Available from:

https://www.hackerearth.com/blog/developers/arduino-programming-for-beginners/

[2].Flaviocopes. Available from:
https://flaviocopes.com/arduino-programming-language/

[3].Arduino Store. Available from:
https://store.arduino.cc/usa/mega-2560-r3

[4].Fierce Electronics. Available from: https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor#:~:text=An%20ultrasonic%20sensor%20is%20an,sound%20that%20humans%20can%20hear

[5].Components101. Available from:
https://components101.com/wireless/esp8266-pinout-configuration-features-datasheet

[6].BCRobotics. Available from:
https://bc-robotics.com/tutorials/controlling-a-solenoid-valve-with-arduino/

[7].Ebay. Available from:
https://www.ebay.com/itm/DS18B20-Water-Temperature-Sensor-Board-Transducer-Module-for-ark-/263276784260

[8].diyi0t. Available from:
https://diyi0t.com/technical-datasheet-microcontroller-comparison/

[9].Arrow. Available from:
https://www.arrow.com/en/research-and-events/articles/arduino-uno-vs-mega-vs-micro

[10]Areen Nadi Shalaldeh (2018). "Water Level Monitoring". Available from:
http://scholar.ppu.edu/handle/123456789/1313

[11]. Researchgate.Arrow. Available from:
https://www.researchgate.net/publication/346705339_Integrated_Water_Monitoring_and_Control_System-_IWMCS

[12]. The smarthome hookup. Available from:
https://www.thesmarthomehookup.com/nodemcu-esp32-and-esp8266-pin-modes-analog-and-digital-inputs-and-outputs/

[13]. Simple projects. Available from:
https://simple-circuit.com/mplab-xc8-hc-sr04-ultrasonic-sensor-pic/

[14]. Components101. Available from:
https://components101.com/wireless/esp8266-pinout-configuration-features-datasheet

[15]. Researchgate.Arrow. Available from:
https://www.researchgate.net/publication/336361248_ANTI_COLLISION_SECURITY_SYSTEM_RECENT_INNOVATIONS_IN_ENGINEERING_AND_TECHNOLOGY

[16]. Elprocus. Available from:
https://www.elprocus.com/ds18b20-temperature-sensor/
[17]. Circuit digest. Available from:
https://circuitdigest.com/microcontroller-projects/measuring-turbidity-of-water-to-determine-water-quality-using-arduino-turbidity-sensor
[18]. Researchgate.Arrow. Available from:
https://www.researchgate.net/publication/322918493_Flexible_Automatic_Water_Level_Controller_and_Indicator
[19]. Hotmcu. Available from:
https://www.hotmcu.com/g34-water-flow-sensor-p-311.html
[20]. Flutter. Available from:
https://flutter.dev/?gclid=Cj0KCQjw5auGBhDEARIsAFyNm9FYsH5AeYbd-L2GC-QlqrIrzlCbM7orZ9oirOzu0igF1mZL6Odw6bYaAoG_EALw_wcB&gclsrc=aw.ds
[21]. Maker advisor. Available from:
https://makeradvisor.com/esp32-vs-esp8266/
[22].ESP32. Available from:
http://esp32.net/

Appendices

Appendix A

Datasheets for needed components:

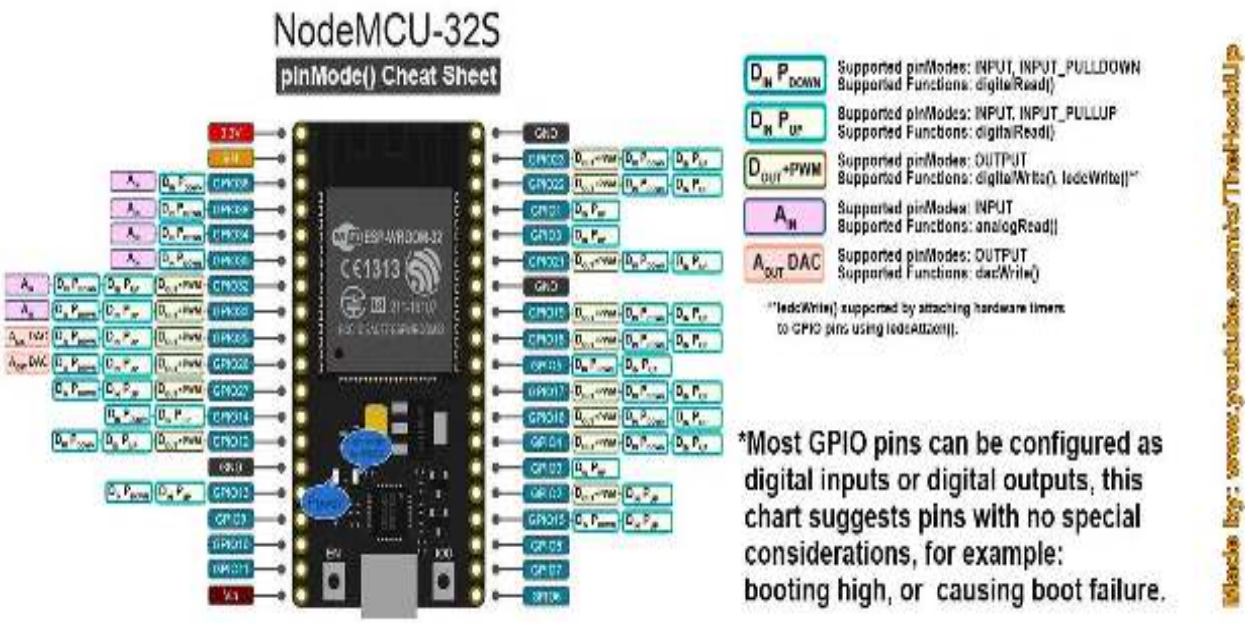

Figure 1: NodeMCU ESP32 pinout