

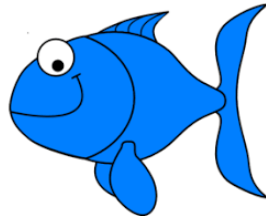


An-Najah National University

Faculty of Engineering & Information Technology
Department of Computer Engineering

Graduation Project II

Smart Aquarium



Prepared by:
Nora Al-Sadder
Mayar Abdulkareem

Supervised by:

Dr. Raed Qadi

Presented in partial fulfillment of the requirements for
Bachelor's Degree in Computer Engineering

May 2023

Acknowledgment

We would like to express our heartfelt appreciation to everyone who has been instrumental in the success of this project. First and foremost, we extend our deepest gratitude to our families for their unwavering support and encouragement throughout our college journey. Their belief in us has been a constant source of motivation.

We especially would like to acknowledge the invaluable support and guidance the Computer Engineering Department provided. Their expertise and mentorship have significantly impacted our growth and development. A special word of appreciation goes to our supervisor, Dr. Raed Qadi, for his invaluable mentorship and guidance.

We are also immensely grateful to our friends, whose companionship and encouragement have been invaluable. They have been there for us during moments of fatigue and frustration, lifting our spirits and reminding us of our capabilities.

Disclaimer

This report was written by Nora Sadder and Mayar Abdulkareem at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of the assessment, and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Table of Contents

Acknowledgment	2
Disclaimer	2
Abstract	5
Chapter 1: Introduction	6
1.1 Statement of the problem.....	6
1.2 Objective of the Work.....	6
Chapter 2: Constraints, Standards/ Codes, and Earlier Coursework	7
2.1 Constraints.....	7
2.2 Standards/Codes.....	7
2.3 Earlier Coursework	8
Chapter 3: Literature Review	8
Chapter 4: Methodology	9
4.1 Hardware Components:	9
4.1.1 Arduino Mega 2560.....	9
4.1.2 NodeMCU ESP32.....	10
4.1.3 Environment-controlling Components	10
4.1.5 Pumps and Solenoids	15
4.1.6 Relays:	16
4.1.7 Servo Motor:	16
4.1.8 Real Time Clock (RTC) DS1307:	17
4.1.9 12 Volts Power Supply:.....	17
4.1.10 Ultrasonic Sensors:.....	18
4.2 System Design:	19
4.3 System Process of Work	20
4.4 Flow Chart:	21
4.5 Software Application	22
Chapter 5: Results and Discussion	29
Chapter 6: Conclusion	30
6.1 Summary of Results and Interpretations	30
6.2 Recommendations for Performance Improvement.....	30
6.3 Conclusions Drawn	30
6.4 Future Work and Directions	31
.6.5 Open Problems.....	31
Chapter 7: References	32

List Of Figures

- Figure 1 Arduino Mega25609
- Figure 2 NodeMCU ESP32 10
- Figure 3 12V Led Strip 10
- Figure 4 Water Controlling Components 11
- Figure 5 JSN SR-40T Circuit 12
- Figure 6 JSN SR-40T Sensor 12
- Figure 7 DS18B20 Sensor 12
- Figure 8 DS18B20 Circuit..... 12
- Figure 9 Peltier Component 13
- Figure 10 Peltier Circuit..... 13
- Figure 11 PH Sensor 13
- Figure 12 TDS Sensor 14
- Figure 13 TDS Circuit..... 14
- Figure 14 Pump and Solenoid 15
- Figure 15 4-channel relay..... 16
- Figure 16 Servo Motor 16
- Figure 17 3D- Printed Auto Feeder 16
- Figure 18 RTC DS1307 17
- Figure 19 12V Power Supply 17
- Figure 20 Ultrasonic Sensor 18
- Figure 21 System Design 19
- Figure 22 Serial Communication 20
- Figure 23 Flow Chart 21
- Figure 24 First Screen After filling the tank information 22
- Figure 25 First Screen Before filling the tank information..... 22
- Figure 26 Home Page 23
- Figure 27 Track the Values Page 24
- Figure 28 Control Page..... 25
- Figure 29 Change The Default Ranges Page 26
- Figure 30 If Clean Water Bucket Size is small Notification 27
- Figure 31 Leakage Notification 27
- Figure 32 If Dirty Water Bucket isn't empty enough to be filled Notification 28
- Figure 33 If Clean Water Bucket doesn't contain sufficient amount of water notification..... 28
- Figure 34 If Dirty Water Bucket Size is small Notification 28

Abstract

Our project is an automated fish tank system controlled by Arduino to provide a convenient, and efficient way of keeping fish, while also promoting a healthy and comfortable environment for them by continuously monitoring and controlling various parameters like water quality, quantity, temperature, and light intensity which affects the health and well-being of the aquatic life in the tank. The system incorporates various sensors to measure those parameters and a microcontroller to process the data in order to make the necessary adjustments. An automatic feeder that dispenses food at set intervals and a water changing system that maintains the proper water quality are also provided in this system. The fish-keeper can easily monitor and track the system's operations and parameters through a user-friendly mobile application at any time.

Our system parameters:

1. **Water Quality:** pH level, and total dissolved solids.
2. **Water Quantity:** a water level sensor will continuously measure the water level and it will give an alert to the owner if it falls below a desired level.
3. **Water Temperature:** use a water temperature sensor and Peltier to monitor the water temperature and adjust it as necessary to maintain a stable and comfortable environment for our fish.
4. **Auto Feeder:** an auto feeder will be provided in the system to dispense food at set intervals, ensuring that the fish are well-fed.
5. **Lighting:** control a LED light as a simulator of the day-night lighting for the fish tank.
6. **Automatically changing the tank water:** the system will automatically change part of the tank water when the water quality falls down or when 2 weeks pass without changing it.

Chapter 1: Introduction

As the trend of keeping pets increases, there is a need to provide special care for animals like fish due to their unique environmental requirements. Manual maintenance of aquarium conditions can be challenging, leading to higher fish mortality rates. Automating this process can significantly reduce fish deaths and provide convenience for owners.

1.1 Statement of the problem.

The nature of our work revolves around creating a smart fish aquarium system that incorporates automation and remote monitoring capabilities. The design problem we are dealing with is the inefficiency and limitations of manual aquarium management. Manual tasks such as water changes, feeding schedules, and maintaining optimal environmental conditions can be time-consuming and prone to human error. Furthermore, inconsistent water quality and environmental conditions can negatively impact the health and well-being of the fish.

1.2 Objective of the Work

Objectives of the Work: The primary objective of our work is to implement automation and real-time monitoring in aquariums, by developing a user-friendly mobile application that enables aquarium owners to conveniently control and monitor their aquariums remotely. We aim to achieve the following:

- Improve the overall user experience in managing fish aquariums.
- Enhance the well-being and health of the fish by maintaining optimal environmental conditions.
- Provide convenience and flexibility by enabling remote access and control of the aquarium system.
- Reduce the manual effort and potential errors associated with traditional aquarium management.

Chapter 2: Constraints, Standards/ Codes, and Earlier Coursework

2.1 Constraints

1. Diverse Population of Fish

- Each fish type has specific requirements for pH levels, temperature, and other factors.
- Our system could only cater to the general case and allow users to manually adjust these parameters.
- This limitation prevented us from providing tailored conditions for each fish species.

2. Waterproof Sensors

- Due to the aquatic environment, waterproof sensors were necessary to ensure accurate data collection.
- The requirement for waterproof sensors resulted in additional project costs.

3. Difficulty in Finding an Appropriate Fish Tank

- We encountered challenges in finding a plastic fish tank of the desired size.
- Most available tanks were made of glass, causing delays in assembling the project's body and connecting the sensors and components.
- To overcome this issue, we used a plastic box typically used for organizing and storage purposes.

2.2 Standards/Codes

The following standards were adhered to during the development of the smart aquarium project:

1. **Programming Language and IDE:** The hydroponic features were programmed using the Arduino IDE, which is written in C/C++. The Arduino IDE served as the platform for developing and uploading the code to the Arduino Mega. Similarly, the ESP32 was programmed using the Arduino IDE to establish communication with the application via Wi-Fi, utilizing the MQTT protocol with the MQTT dashboard (MQTT Dashboard, n.d.).
2. **Application Development:** The MIT App Inventor (MIT App Inventor, n.d.), an online cloud-based tool, was employed for creating the application. This web-based tool allowed for the development of apps for both Android and iOS platforms using a web browser. By leveraging the capabilities of the MIT App Inventor, a user-friendly and intuitive application was built for controlling and monitoring the smart aquarium.

By following these standards, the smart aquarium project ensured compatibility, reliability, and ease of maintenance across different components and platforms.

2.3 Earlier Coursework

During our academic journey, we took several hardware courses that significantly contributed to the success of our smart aquarium project. Like the microcontroller course, which provided us with valuable knowledge and skills that directly benefited this project.

In addition to the theoretical knowledge gained from hardware courses, we also had the opportunity to apply our skills in practical settings. The PIC lab played an important role in our learning journey, providing hands-on experience in working with microcontrollers and implementing hardware functionalities. We gained a deeper understanding of utilizing Arduino for controlling I/O devices, implementing serial communication, and effectively managing stepper motor operations. The activities in the lab helped us apply what we learned in our coursework. It gave us the skills we needed to effectively use and combine different hardware parts in our smart aquarium project.

Chapter 3: Literature Review

In the context of smart aquarium systems, extensive research has been conducted to explore the application of IoT technologies for remote monitoring and control of aquarium environments. Two notable papers, "Smart Aquarium: An IoT Based System for Remote Monitoring and Control" (Khan et al., 2019) and "Automatic Fish Feeder and Telegram Based Aquarium Water Level Monitoring" (Dewantara et al., 2023), have contributed valuable insights in this field.

The paper titled "Smart Aquarium: An IoT-Based System for Remote Monitoring and Control" presents an in-depth exploration of an IoT-based solution for monitoring and controlling aquariums. The authors discuss the implementation of various sensors and actuators, such as temperature sensors, pH sensors, and automatic feeding systems, to ensure optimal conditions for aquatic life. They also propose a user-friendly mobile application for remote access and control of the aquarium parameters.

Similarly, the paper "Automatic Fish Feeder and Telegram Based Aquarium Water Level Monitoring" focuses on automating fish feeding processes and monitoring water levels in aquariums by utilizing Telegram, a widely used instant messaging platform. By integrating an automatic fish feeder mechanism with a Telegram-based monitoring system. The authors introduce a system that provide real-time tracking and convenience for aquarium enthusiasts.

By incorporating the knowledge and findings from these resources, our aim is to create a comprehensive and advanced system that can remotely monitor and control various parameters of the smart aquarium. Through the integration of IoT technologies, we seek to provide a convenient and efficient means of maintaining a healthy and comfortable environment for aquatic life while empowering users with real-time access and control through a user-friendly mobile application.

Chapter 4: Methodology

This chapter presents a comprehensive overview of our system, including the hardware components employed, their interconnections, system design, workflow process, and the software application employed in our project.

4.1 Hardware Components:

4.1.1 Arduino Mega 2560

The Arduino Mega 2560 is a microcontroller board, it's equipped with 4kb (4096 bytes) of EEPROM, which retains data even when power is disconnected. With the Arduino Mega 2560, you have access to 54 digital pins, 15 of which support PWM and 16 analog input pins. Additionally, the Mega 2560 offers the convenience of 4 hardware serial ports (UARTs), enabling seamless connectivity to multiple devices.

Also, it is programmable with the Arduino IDE which is written using C/C++ via a type B USB cable.

We used Arduino Mega 2560 because we needed many digital pins to interface the sensors and other components with.



Figure 1 Arduino Mega2560

4.1.2 NodeMCU ESP32

The NodeMCU ESP32 is a small yet powerful development board that utilizes the ESP32 microcontroller. It comes equipped with Wi-Fi and Bluetooth functionality, making it a perfect fit for IoT (Internet of Things) ventures. We employed the Arduino IDE to program the board, enabling communication and data exchange with the application via the MQTT protocol [3] which requires a Wi-Fi connection.

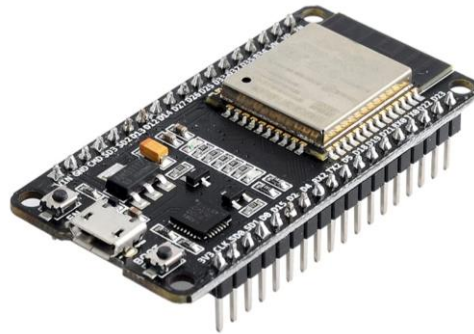


Figure 2 NodeMCU ESP32

4.1.3 Environment-controlling Components

To control the environment around the fish, we check the lights to provide them with suitable conditions to grow.

All fishes need a specific amount of light to grow and develop, regulate their internal biological clock, reduce their stress levels, and promote their overall mental health. so, we used RTC to schedule the lights to turn on/off every 12 hours through a LED strip to simulate day/night light.

▪ Led strip



Figure 3 12V Led Strip

4.1.4 Water-controlling Components

The water inside the fish tank should have a suitable amount of pH, and nutrients.

Also, the water temperature should be within the normal range.

We also monitored the water level using waterproof ultrasonic to make sure there is no leakage in the tank.

We used DS18B20 waterproof sensor to measure the water temperature in the tank, we used with it a Peltier to cool or heat the water depending on the value of this sensor. We turn on the Peltier on heating when the water needs to be warmed, and we turn it on cooling when the water needs to be cooled.

To make sure the water is having a sufficient amount of nutrients and pH, we used a TDS sensor to measure the total dissolved solids which indicates the nutrients in the water, we used also a pH sensor to indicate the pH value.

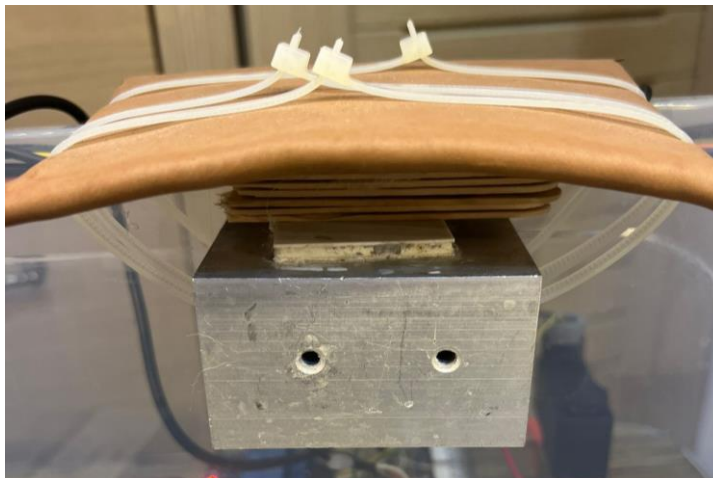


Figure 4 Water Controlling Components

▪ **Waterproof ultrasonic JSN SR-40T:**

to measure the water level, we used a waterproof ultrasonic sensor JSN SR-40T.

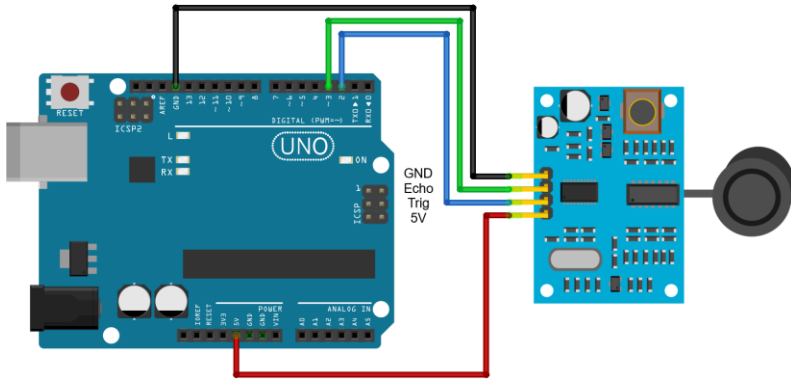


Figure 5 JSN SR-40T Circuit



Figure 6 JSN SR-40T Sensor

▪ **Water temperature - DS18B20:**

We used DS18B20 waterproof sensor to measure the water temperature.



Figure 7 DS18B20 Sensor

DS18B20 circuit

We used a DS18B20 sensor on with a digital pin from Arduino, 4.7k ohm resistance, 5 volts, and ground as shown in the figure.

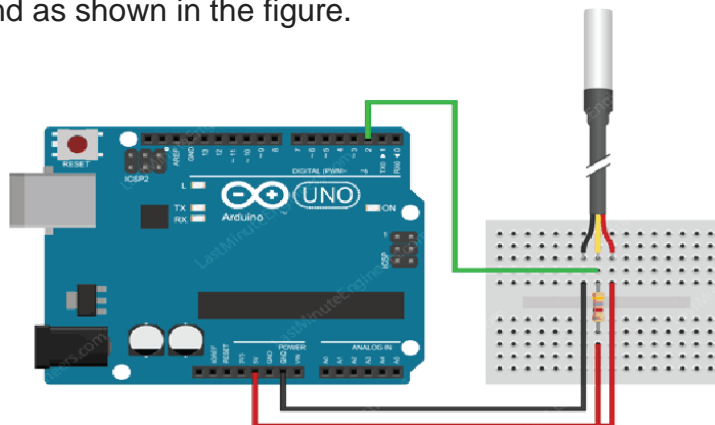


Figure 8 DS18B20 Circuit

▪ Peltier:

The Peltier cooler and heater is used for changing the water temperature when it connects to power (12-volt) one side will be hot and the other side will be cold, we used it with a 2-channel relay to control the side that touches the water. The other side's heat will not affect the tank water. Also, we connected a metal part to the side which touches the water to distribute the temperature.

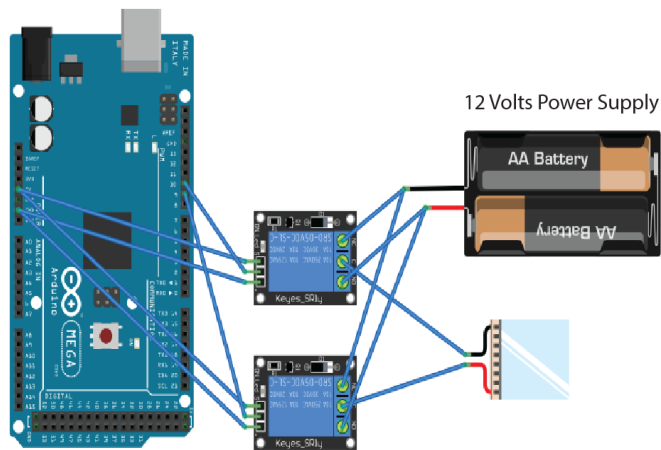


Figure 10 Peltier Circuit



Figure 9 Peltier Component

▪ PH sensor:

We used a waterproof pH sensor to measure the water's alkalinity and acidity.



Figure 11 PH Sensor

▪ **TDS sensor:**

We used a waterproof TDS to determine the number of dissolved particles in the water, which indicates the amount of dissolved nutrients.

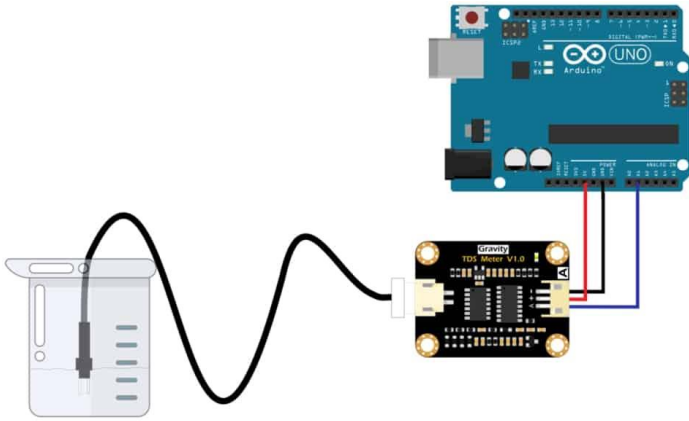


Figure 13 TDS Circuit

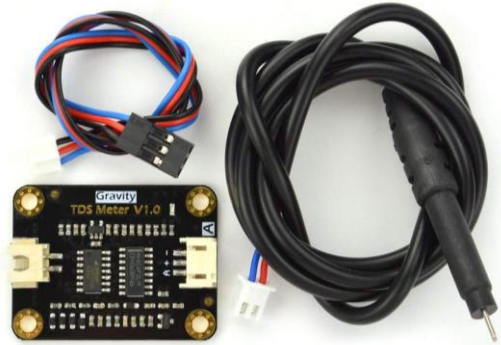


Figure 12 TDS Sensor

4.1.5 Pumps and Solenoids

These 2 pumps are used in the change water process, they are 12V pumps, and each one is connected to the Arduino through a relay as shown in Figure.

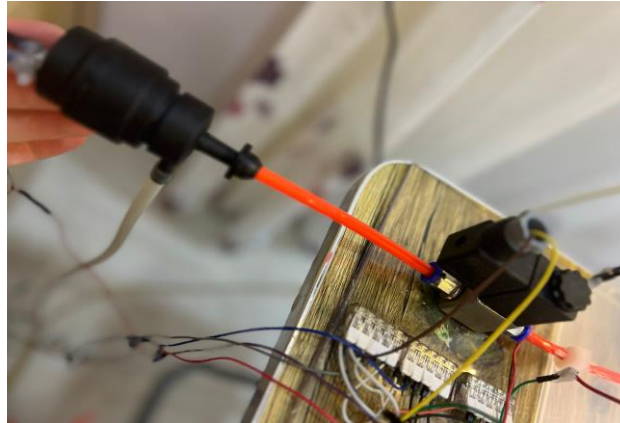


Figure 14 Pump and Solenoid

1. **Remove the dirty water pump and solenoid:**

this pump is used to remove 15% of the dirty water in the fish tank whenever a change water process is done. The pump removes 1 liter/minute, so based on the tank size the user entered from the application, it calculates the time it needs to remove 15% of the water. The solenoid valve is connected to the bottom of the fish tank and connected to the other side with the pump so water can flow whenever the valve is opened and the pump is on due to the water pressure. The other side of the pump is connected to a small tank to store the dirty water. The use of the solenoid is because the pump flows the water even when it's off because of the water pressure.

2. **Add a clean water pump and solenoid:**

this pump is used to add clean water to the fish tank after removing the dirty water whenever a change water process is done. The pump keeps adding water to the tank until the water level reaches the normal water level in the tank based on the tank height value the user entered from the application. The solenoid valve is connected to the bottom of the clean water tank and connected to the other side with the pump so water can flow to the pump whenever the valve is opened and the pump is on due to the water pressure. The other side of the pump is put inside the fish tank to provide it with clean water. The use of the solenoid is to prevent the water in the fish tank to return to the clean water tank since the pipe will be filled with water.

4.1.6 Relays:

Since the Arduino operates on 5-volt, it can't control the 12-volt devices such as the pumps, solenoids, Peltier, and LED strip. But we can use a 5-volt relay to switch the high voltage and use the Arduino to control the relay. We used 5 relays, 2 of them to control the water pumps, one to control the LED strip, and 2 channels for the Peltier to control heating and cooling.



Figure 15 4-channel relay

4.1.7 Servo Motor:

We used a servo motor to control the 3D-printed auto feeder which feeds the fish daily with a suitable amount of food regarding the number of fish in the tank.



Figure 16 Servo Motor

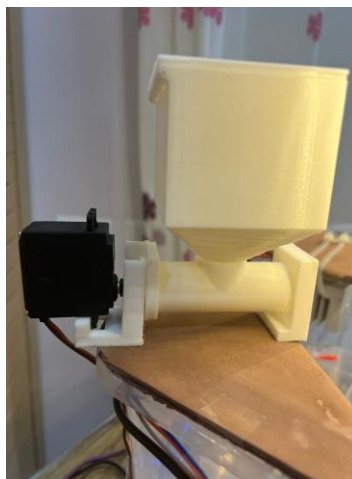


Figure 17 3D- Printed Auto Feeder

4.1.8 Real Time Clock (RTC) DS1307:

we used RTC to control the automatic operations which occur regularly like feeding the fish daily, changing the water every 2 weeks, and turning lights on/off every 12 hours.

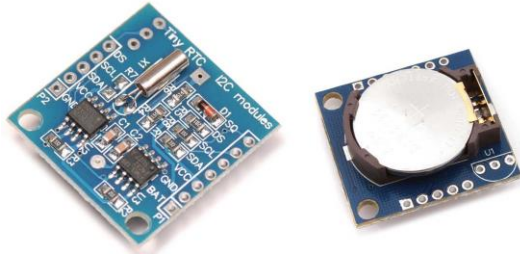


Figure 18 RTC DS1307

4.1.9 12 Volts Power Supply:

Since Arduino works only on 3.3 or 5 Volts, we need an external power supply to provide 12 volts for some components like pumps, led strip, and Peltier.



Figure 19 12V Power Supply

4.1.10 Ultrasonic Sensors:

We used 2 Ultrasonic Sensors to measure the water level in the clean and dirty water buckets to make sure the changing water operation works well.



Figure 20 Ultrasonic Sensor

4.2 System Design:

Our project is a plastic fish tank which consists of 6 main parts, first part is the basket which contains the water quality sensors that measures the water pH, and total dissolved solids (TDS) and the pipe which pumps the clean water to the tank, the second one is the hole in the bottom of the tank which is connected to a solenoid and pump to remove the dirty water, the third one which is above the tank is the led strip which provides the fish with the required light and water level sensor to measure the water level, the 4th part is the automatic feeder in the corner which is connected to a servo motor to dispense the food in the water, the 5th part is the Peltier which cools/heats the water and the last part is the 2 water buckets which are used for storing the clean water needed to be pumped to the tank and storing the dirty water pumped from the tank.

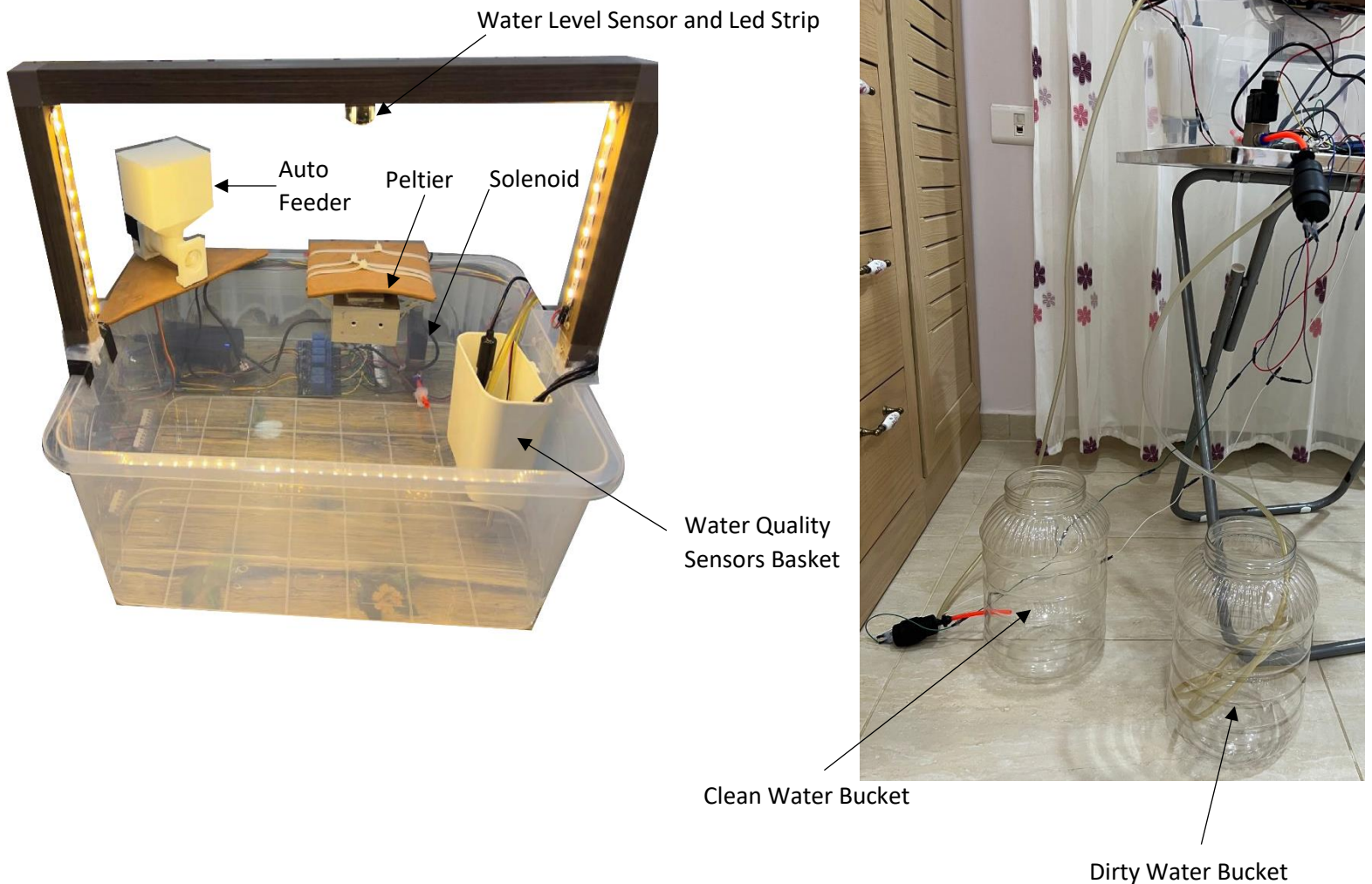


Figure 21 System Design

4.3 System Process of Work

When the owner of the fish starts the system, at first, he must enter the number of fish in the tank, tank size and tank height. Then, the sensors will keep tracking the temperature and the water quality parameters like pH, and nutrients in the water. If the temperature falls under the minimum value, the heater will turn on. And if the temperature goes above the maximum value the cooler turns on. If one of the water quality values falls down or rises above the normal range, the pumps and solenoids will work and water changing operation will occur.

On the other hand, there are automatic operations that occur regularly like feeding the fish daily with suitable quantity based on the number of fish in the tank, changing 15% of the tank water every 2 weeks, and turning lights on/off every 12 hours.

Also, the system keeps tracking the water level, if it goes down 75% of the tank height, a notification will be sent to the owner on his phone to alert him about leakage in the fish tank.

The owner will be able to track all the values of the water quality parameters and the time of the last feed and changing water through an application, he will be also able to make any operation manually like feeding, changing water, adding water, turning lights on/off, turning the heater on/off, and turning the cooler on/off.

- **Serial Communication:**

We used Serial Communication to transfer data between NodeMCU and Arduino using RX/TX lines. For Arduino mega there are 4 serial connections, we used serial2, pins

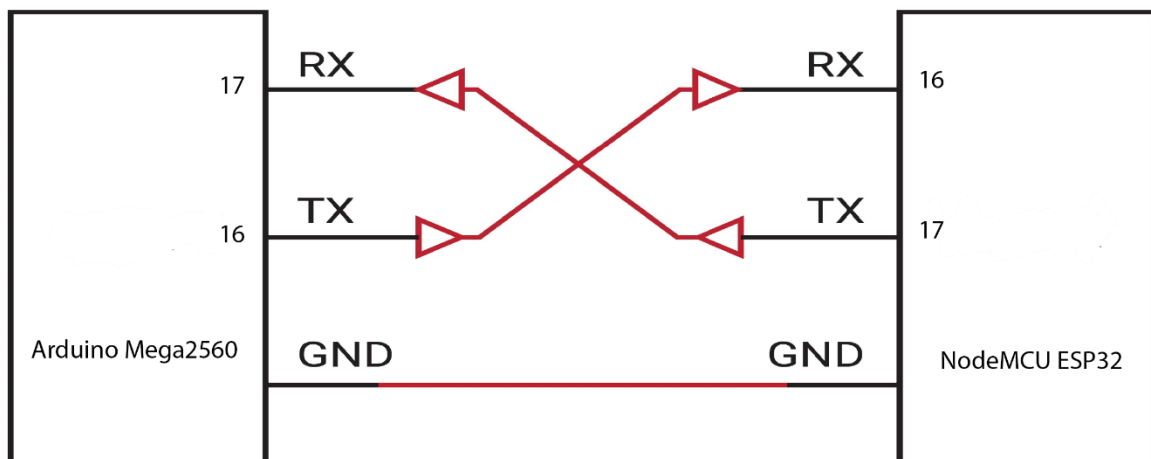


Figure 22 Serial Communication

4.4 Flow Chart:

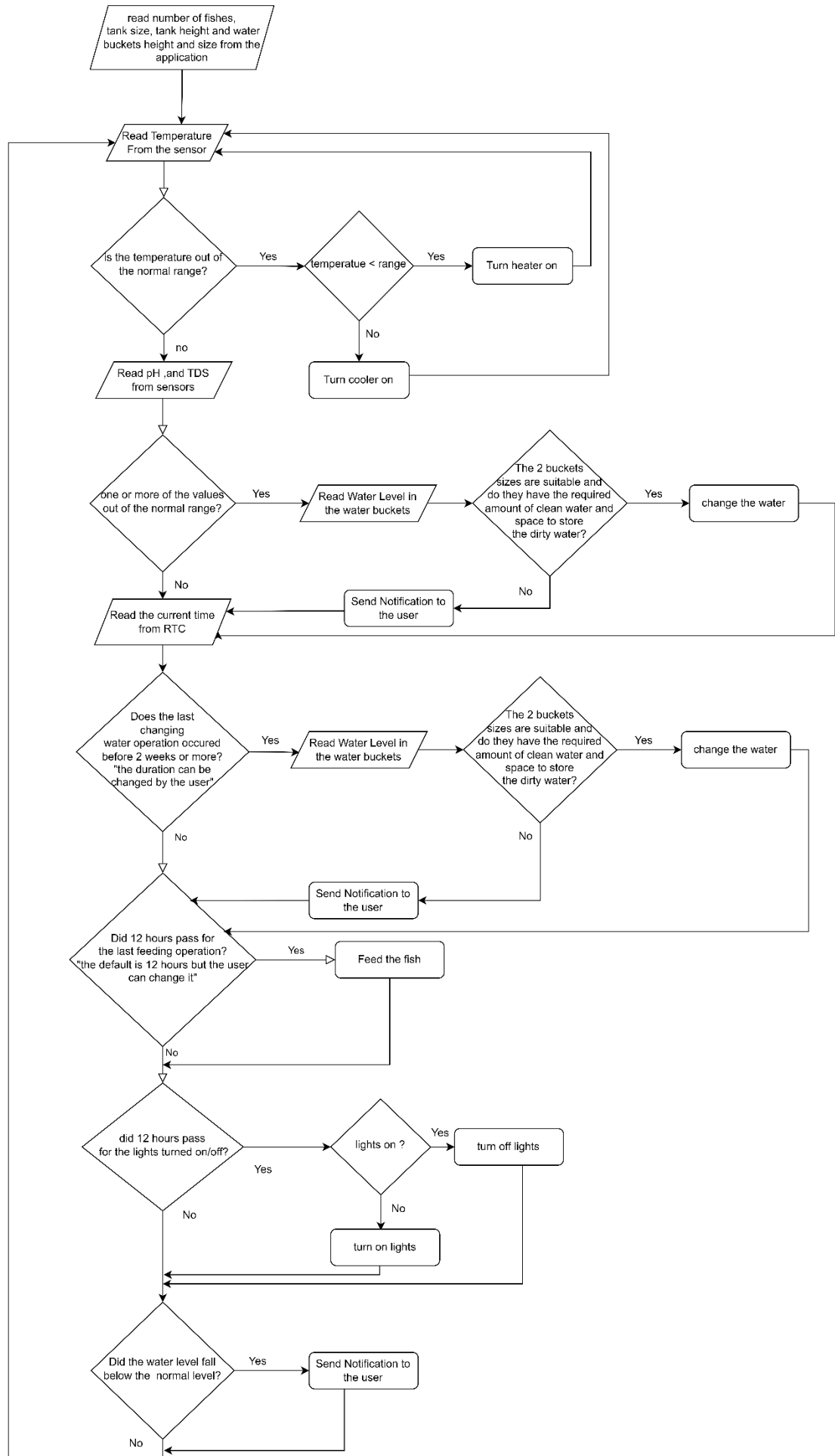


Figure 23 Flow Chart

4.5 Software Application

We used the MIT app inventor online open source to build the application, to connect it to the project we used the IoT in the NodeMCU ESP32, through the MQTT protocol with an online broker dashboard, through the Wi-fi. We used 3 topics to publish data from the application to the NodeMCU through MQTT and 9 topics to subscribe and get data from NodeMCU to the application.

We first connect the NodeMCU to the Wi-fi and then connect it to the broker, when it's connected it starts to get and send data to the application and then to the Arduino serially.

Before the system starts, the owner must enter the tank information like tank height in cm, tank size in liters, and number of fish in the tank and the clean/dirty water buckets information like height and size. Then he will be able to use the application. Also, he can return to this screen whenever he wants to edit the values.

Number Of Fish In Tank

Size Of Tank in Litres

Height Of Tank in CM

Clean Water Bucket Information:

Height (CM) Size (Litres)

Dirty Water Bucket Information:

Height (CM) Size (Litres)

* Please fill all the blanks

Save Changes Home Page

Figure 25 First Screen Before filling the tank information

Number Of Fish In Tank

Size Of Tank in Litres

Height Of Tank in CM

Clean Water Bucket Information:

Height (CM) Size (Litres)

Dirty Water Bucket Information:

Height (CM) Size (Litres)

Saved successfully

Save Changes Home Page

Figure 24 First Screen After filling the tank information

from the homepage, the user can go to any screen he wants

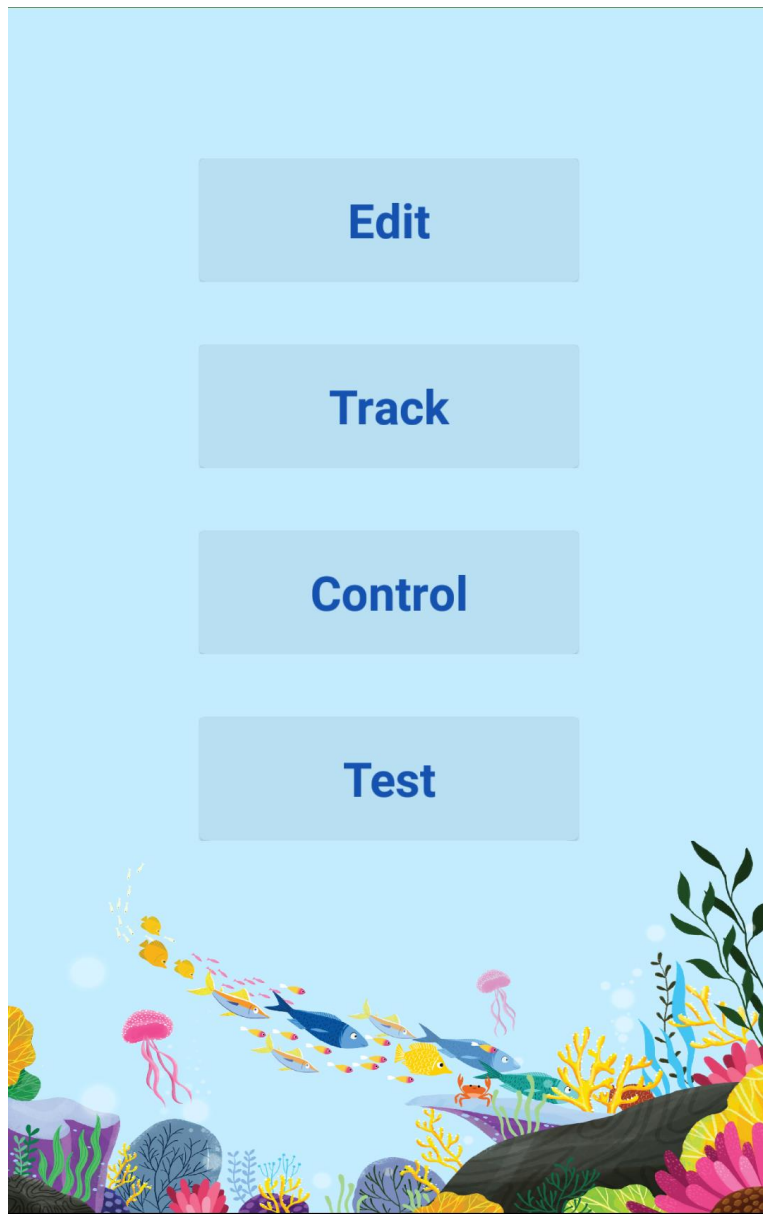


Figure 26 Home Page

the user can track the values of the application as shown



Figure 27 Track the Values Page

the user can also control any operation manually

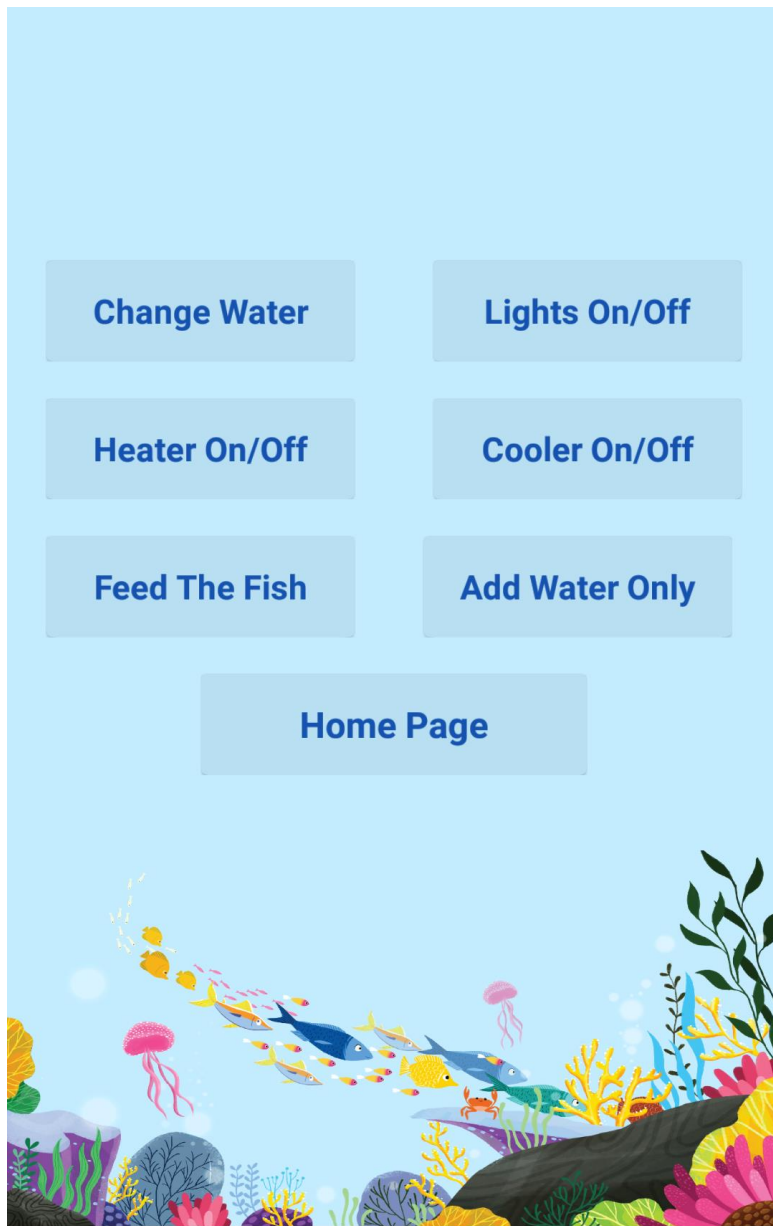


Figure 28 Control Page

the user also can change the default ranges of pH, TDS, and temperature

min temp	12	max temp	27
min ph	3	max ph	28
min TDS	0	max TDS	660
feed time (Hrs)	1	change water (days)	7

Save Home leakage

Figure 29 Change The Default Ranges Page

The Notifications in the application:

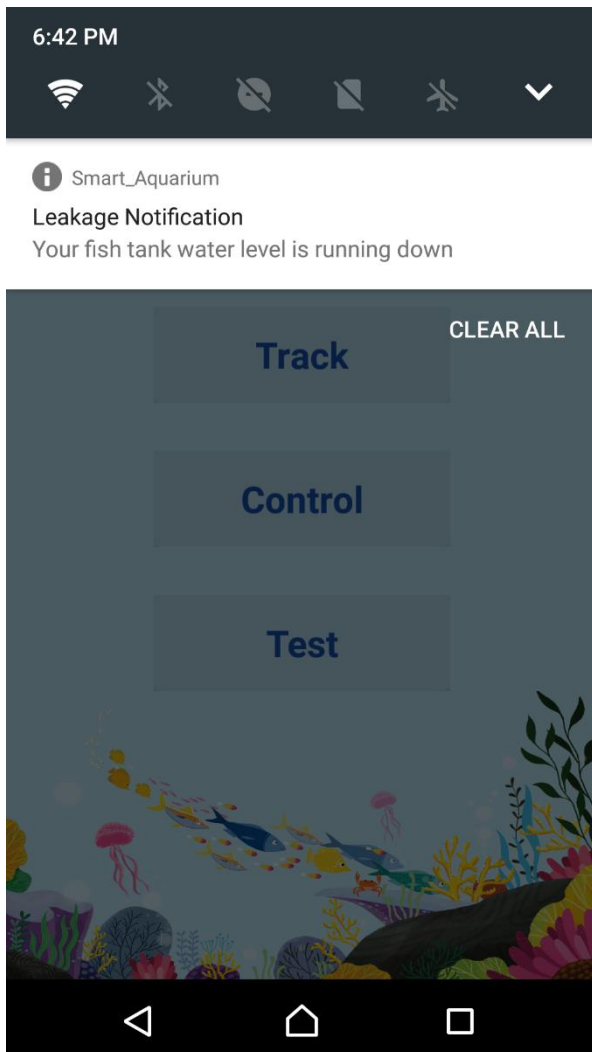


Figure 31 Leakage Notification

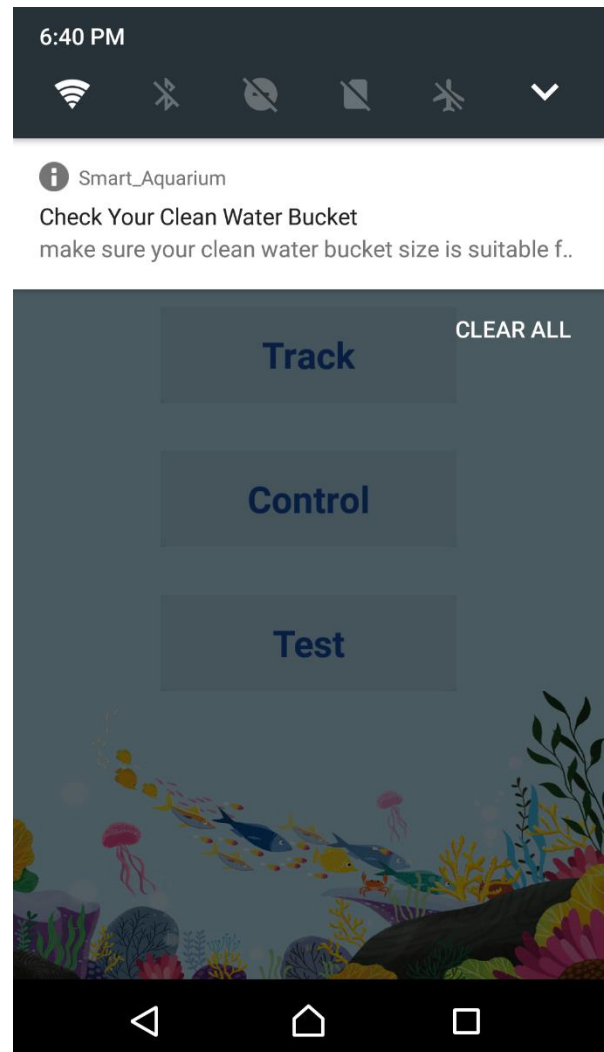


Figure 30 If Clean Water Bucket Size is small Notification



Smart_Aquarium

Check Your Dirty Water Bucket

make sure your dirty water bucket size is suitable fo..

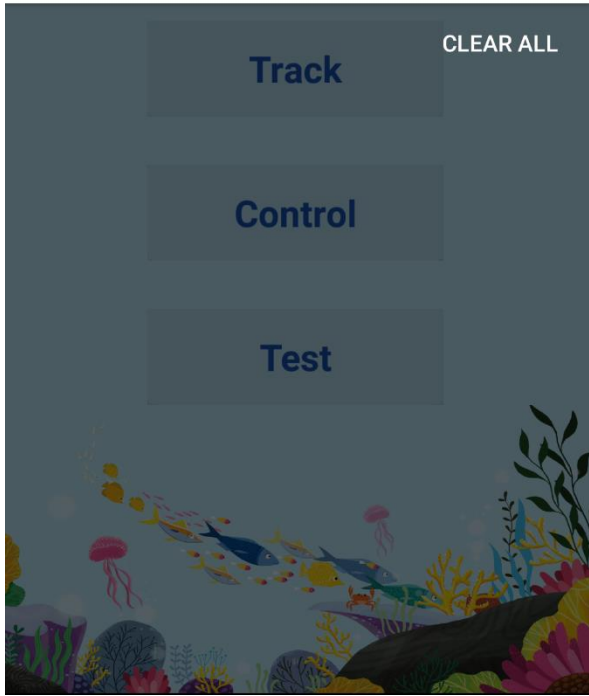


Figure 34 If Dirty Water Bucket Size is small Notification



Smart_Aquarium

Check Your Clean Water Bucket

make sure your clean water bucket contains the req..

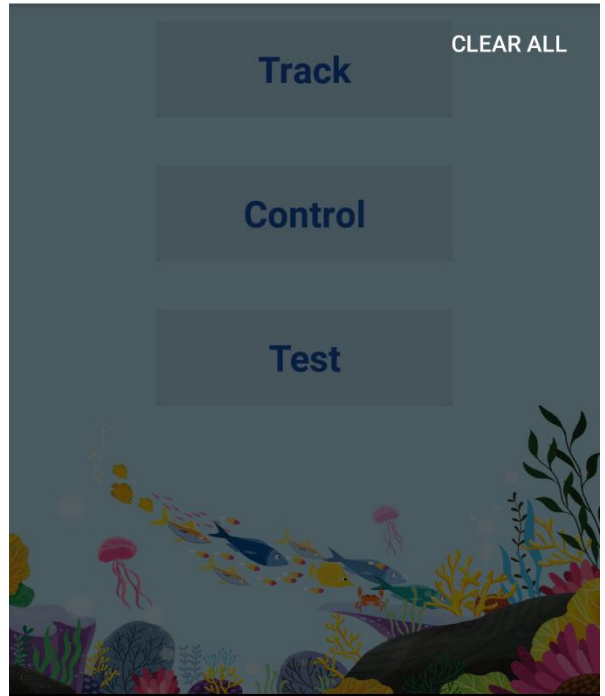
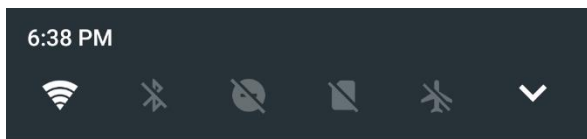


Figure 33 If Clean Water Bucket doesn't contain sufficient amount of water notification



Smart_Aquarium

Check Your Dirty Water Bucket

make sure your dirty water bucket is empty enough ..

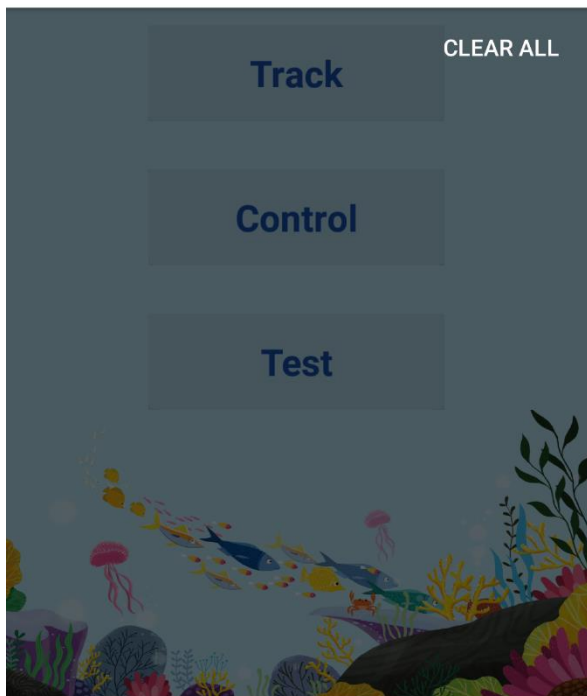


Figure 32 If Dirty Water Bucket isn't empty enough to be filled Notification

Chapter 5: Results and Discussion

In the end, we succeeded in creating the desired result, a smart aquarium system that incorporates various sensors to monitor key water parameters such as pH level, temperature, and total dissolved solids in water. Real-time data is displayed through a user-friendly mobile application, enabling users to track and analyze the water conditions. The system automatically adjusts and maintains optimal parameters, such as temperature by turning the heater or the cooler on depending on the situation, also, pH and TDS by changing the water when they go out of the normal range, to create a stable and comfortable environment for the aquatic life. Remote monitoring and control are facilitated through the mobile application, allowing users to access real-time data and make adjustments from anywhere at any time. The system's intuitive interface and graphical representation of data enhance the user experience. Overall, our smart aquarium system transforms the way aquariums are monitored and maintained, providing convenience, efficiency, and improved well-being for aquatic life.

Chapter 6: Conclusion

In this chapter, we present the conclusions drawn from our smart aquarium project and provide recommendations for future improvements. This section aims to summarize the important results and interpretations while offering cost-effective and feasible ways to enhance the system's performance. We will also discuss the key learnings from the project and outline future directions for further research and development.

6.1 Summary of Results and Interpretations

In summary, our smart aquarium system has successfully demonstrated advancements in automation, remote monitoring, and control of aquarium conditions. We have created a user-friendly and efficient system by integrating various technologies and employing clean coding practices.

The automation features, including automated water changes, feeding schedules, and lighting control, have proven effective in reducing manual effort and maintaining consistent and optimized conditions for aquatic organisms. The remote monitoring and control capabilities through the mobile application have provided users with real-time access to water parameters and the ability to make adjustments conveniently from anywhere.

6.2 Recommendations for Performance Improvement

Based on our findings, we recommend several cost-effective and feasible ways to improve the performance of the smart aquarium system:

1. Enhance water quality monitoring: Explore advanced sensors and technologies for monitoring water parameters with increased accuracy and reliability, ensuring a more comprehensive understanding of the aquarium's environment.
2. Explore alternative communication protocols: Investigate different communication protocols to improve system reliability, especially in scenarios with limited or unstable Wi-Fi connectivity.

6.3 Conclusions Drawn

From this project, we draw the following conclusions:

1. Automation and remote monitoring significantly improve aquarium management by reducing manual effort and enabling real-time access to crucial information.
2. Clean coding practices are essential for creating a reliable and maintainable application, ensuring the long-term usability and scalability of the system.
3. Integration of hardware and software technologies provides a comprehensive and user-friendly solution for smart aquariums, enhancing the well-being of aquatic organisms and user experience.

6.4 Future Work and Directions

Moving forward, there are several areas for future research and development in the field of smart aquariums:

1. Due to the diverse range of fish species present, determining the ideal pH, temperature, and other environmental parameters for each type poses a challenge. Therefore, we recommend retrieving these values from an API as it provides up-to-date and species-specific information. Additionally, considering that the appropriate food quantity depends on the size of the fish, we propose incorporating a user input feature to allow for entering the fish's size.
2. Expand sensor capabilities: Explore the integration of additional sensors to monitor a wider range of water parameters and environmental conditions like Oxygen and NH₃ sensors.
3. Notification for Rapid and Unsatisfactory Water Changes: add water filter to the added water section to make sure the added water has good quality and send a notification if multiple water changes occur within a short period and the new water is still unsatisfactory.

6.5 Open Problems

While our smart aquarium system has achieved significant advancements, there are still open problems that require further exploration:

1. Scalability and interoperability: Scaling for larger aquarium setups and ensuring compatibility with diverse hardware components. This includes optimizing data processing and accommodating different hardware options for the integration.
2. Finding the ideal balance of water parameters, such as pH level, temperature, and other environmental factors for different types of fish. Each fish species may have unique preferences and requirements, making it difficult to create an optimal environment that satisfies all the fish simultaneously.

In conclusion, our smart aquarium project has successfully demonstrated the potential of automation, remote monitoring, and control technologies in improving aquarium management. Through our findings, we recommend implementing machine learning algorithms, enhancing water quality monitoring, and exploring alternative communication protocols to further enhance system performance. Our project contributes to the growing body of knowledge in this field and opens avenues for future research and development.

Chapter 7: References

MQTT Dashboard. (n.d.). MQTT Dashboard. <https://www.mqtt-dashboard.com/>

MIT App Inventor. (n.d.). MIT App Inventor. <https://appinventor.mit.edu/>

Khan, J., Onib-Ur-Rehman, Jalil, Z., Ali, S., & Danish, A. S. (09 2019). Implementation of Smart Aquarium System Supporting Remote Monitoring and Controlling of Functions using Internet of Things. 9, 9–28.

Dewantara, B., Sulistiyowati, I., Jamaaluddin, J., & Others. (2023). Automatic Fish Feeder and Telegram Based Aquarium Water Level Monitoring. Buletin Ilmiah Sarjana Teknik Elektro, 5(1), 98–107.