SMART HYGIENIC DAPHNIA DETECTION SYSTEM

NUR ARIF IKHWAN BIN NAZARUDDIN

BACHELOR OF ELECTRICAL ENGINEERING WITH HONOUR

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023/2024

SMART HYGIENIC DAPHNIA DETECTION SYSTEM

NUR ARIF IKHWAN BIN NAZARUDDIN

Submitted in accordance with requirements of the University Technical Malaysia Melaka (UTeM) for bachelor's degree of Electrical Engineering with Honours

FACULTY OF ELECTRICAL TECHNOLOGY AND ENGINEERING (FTKE)

2023/2024

DECLARATION

I declare that this thesis entitled "SMART HYGIENE DAPHNIA DETECTION SYSTEM" is the result of my own research expect as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in the candidature of any other degree.

Signature : ………………………………………………………………...

Author's Name : NUR ARIF IKHWAN BIN NAZARUDDIN

Date : 19/1/2024

APPROVAL

This report is submitted to the Faculty of Manufacturing Engineering of Universiti Teknikal Malaysia Melaka as a partial fulfilment of the requirement for Degree of Electrical Engineering (Hons). The member of the supervisory committee is as follow:

Signature: ……………………………………………………………………………

Supervisor Name: TS. DR. MOHAMAD FAIZAL BIN BAHAROM

Date: 16/1/2024

DEDICATIONS

I dedicate this final year project to the unwavering support and encouragement of my parents NOORZIE BINTI ABU BAKAR and NAZARUDDIN BIN ABDUL GHANI, whose love and sacrifices have been my constant motivation. Your belief in my abilities has fuelled my determination to reach this academic milestone. I express my sincere gratitude to my dedicated and inspiring lecturer, TS. DR. MOHAMAD FAIZAL BIN BAHAROM, whose guidance and expertise have been invaluable throughout this journey. Additionally, heartfelt thanks to my friends for their encouragement, understanding, and shared moments that made this academic endeavour memorable. This accomplishment reflects the collective support I have received, and I am deeply grateful for the contributions of academic mentors.

ACKNOWLEDGEMENTS

In the Name of God, the Most Merciful, the Most Merciful. Alhamdulillah, all praise be to God Almighty, Lord of the universe and peace and blessings be upon Muhammad His servant and messenger. First of all, I must express my infinite gratitude to Allah, who Great, Grateful, for His help that has helped me in preparing and going through FYP 1 for this semester.

I would like to express my sincere gratitude to my supervisor, Ts. Dr. Mohamad Faizal, for his invaluable guidance, support, and encouragement throughout the entire duration of my Final Year Project 1. His expertise and insights significantly contributed to the success of this project.

I would also like to extend my appreciation to my esteemed panels, Dr. Saifulza bin Alwi @ Suhaimi and Ir. Dr. Norazhar bin Abu Bakar, for their valuable feedback and constructive inputs that greatly enhanced the quality of my FYP 1. A special thanks goes to the Faculty of Electrical Technology Engineering (FTKE) for providing the conducive academic environment and resources necessary for the completion of this project.

Finally, I am deeply grateful to my family and friends for their unwavering support, understanding, and encouragement throughout this academic journey. Their belief in me has been a constant source of motivation. Thank you to everyone who played a role in making this FYP a reality.

ABSTRACT

Daphnia magna is an essential live food stock for aquaculture, providing crucial nutrients to planktivorous fish in freshwater hatcheries, brackish waters, and ornamental fish environments. To optimize Daphnia breeding habitats, the "Smart Hygiene Daphnia Detection System" has been developed, integrating advanced monitoring technology. This system includes a pH monitor for water freshness, a gas detector for hazard identification, a humidity sensor for atmospheric monitoring, and a temperature sensor for maintaining suitable conditions. An ESP32 camera offers visual monitoring and colour-detecting features, which can detect changes in water colour that indicate chlorella green water nutrient recycler and Daphnia blooming process. When such changes are detected, the system promptly informs the breeder via telegram, allowing for timely intervention. Controllable outputs such as a heat lamp, air bubbler, solenoid valve, and water pump ensure precise environmental control. Studies have shown that Daphnia magna are highly sensitive to temperature fluctuations, which can significantly impact water quality. For instance, data collected using DS18B20 and DHT11 sensors indicated a 90% mortality rate when Daphnia were exposed to direct sunlight, highlighting the need for careful temperature management. In an analysis of environmental conditions, it was found that maintaining the water temperature between 20°C and 25°C significantly reduces the mortality rate and promotes optimal growth of Daphnia magna. Additionally, maintaining a pH level between 6.5 and 8.5 were identified as critical parameters for their survival and reproductive success. The "Smart Hygiene Daphnia Detection System" aims to promptly notify breeders of environmental changes, including colour shifts detected by the ESP32 camera, allowing for timely interventions to maintain hygiene and promote effective Daphnia growth. This system has the potential to enhance aquaculture management practices and support the production of nutritious live food stock.

ABSTRAK

Daphnia magna ialah stok makanan hidup yang penting untuk akuakultur, menyediakan nutrien penting kepada ikan planktivor di tempat penetasan air tawar, perairan payau dan persekitaran ikan hiasan. Untuk mengoptimumkan habitat pembiakan Daphnia, " Smart Hygiene Daphnia Detection System" telah dibangunkan, menyepadukan teknologi pemantauan termaju. Sistem ini termasuk pemantau pH untuk kesegaran air, pengesan gas untuk pengecaman bahaya, penderia kelembapan untuk pemantauan atmosfera dan penderia suhu untuk mengekalkan keadaan yang sesuai. Kamera ESP32 menawarkan pemantauan visual dan ciri pengesan warna, yang boleh mengesan perubahan dalam warna air yang menunjukkan pengitar semula nutrien air hijau chlorella dan proses mekar Daphnia. Apabila perubahan tersebut dikesan, sistem segera memaklumkan pembiak baka melalui telegram, membolehkan tindak balas tepat pada masanya. Keluaran yang boleh dikawal seperti lampu haba, gelembung udara, injap solenoid dan pam air memastikan kawalan alam sekitar yang tepat. Kajian telah menunjukkan bahawa Daphnia magna sangat sensitif terhadap turun naik suhu, yang boleh memberi kesan ketara kepada kualiti air. Sebagai contoh, data yang dikumpul menggunakan penderia DS18B20 dan DHT11 menunjukkan kadar kematian 90% apabila Daphnia terdedah kepada cahaya matahari langsung, menyerlahkan keperluan untuk pengurusan suhu yang teliti. Dalam analisis keadaan persekitaran, didapati bahawa mengekalkan suhu air antara 20°C dan 25°C dengan ketara mengurangkan kadar kematian dan menggalakkan pertumbuhan optimum Daphnia magna. Selain itu, mengekalkan tahap pH antara 6.5 dan 8.5 telah dikenal pasti sebagai parameter kritikal untuk kemandirian dan kejayaan pembiakan mereka. "Smart Hygiene Daphnia Detection System" bertujuan untuk memberitahu pembiak baka tentang perubahan persekitaran dengan segera, termasuk peralihan warna yang dikesan oleh kamera ESP32, membolehkan campur tangan tepat pada masanya untuk mengekalkan kebersihan dan menggalakkan pertumbuhan Daphnia yang berkesan. Sistem ini berpotensi untuk meningkatkan amalan pengurusan akuakultur dan menyokong pengeluaran stok makanan hidup yang berkhasiat.

Table of Contents

DECLARATION i
APPROVAL ii
DEDICATIONS iii
ACKNOWLEDGEMENTS iv
ABSTRACT v
ABSTRAK vi
LIST OF TABLES ix
LIST OF FIGURES x
CHAPTER 1 1
1.1 Background 1
1.2 Motivation 2
1.3 Problem Statement 4
1.4 Objective 5
1.5 Scope 6
CHAPTER 2 7
2.1 Introduction 7
2.2 Seafood Rising Trend 7
2.3 Planktonic Crustaceans 9
2.3.1 Daphnia Hatcheries System 10
2.4 Improvements in Aquaculture in Malaysia 10
2.5 Daphnia life needs 12
2.6 Internet of Things (IoT) in Hygiene Monitoring 13
2.7 Hardware Instrument of Daphnia Needs. 14
2.7.1 (Potential of Hydrogen) PH level 14
2.7.2 Ammonia Gas 14
2.7.3 Water Temperature 16
2.7.4 4-module relay 16
2.7.5 Visual Monitoring 16
2.8 Comparison of Previous Monitoring Title Paper. 17
CHAPTER 3 20
3.1 Introduction 20
3.2 Methodology 21
3.3 Gather data on Microcontroller. 21
3.4 Hardware Setup 22

3.5 Hardware requirement 23
3.6 ESP8266 Microcontroller 24
3.6.1 DIYMORE Liquid PH Value Detection Sensor & BNC Electrode PH Probe 26
3.6.2 MQ-135 Gas Detector Sensor 27
3.6.3 DS18B20 Temperature Sensor 28
3.6.4 ESP32-CAM 29
3.6.5 FT232RL TTL Serial Adapter 31
3.3.6 Humidity sensor 31
3.7 Software Requirement 33
3.7.1 MIT App Inventor 34
3.7.2 MQTT broker 36
3.7.3 Arduino IDE 36
3.7.4 Python 37
3.8 Working Flow of The Proposed System 38
3.9 Device development step 40
3.10 Summary 40
CHAPTER 4 41
4.1 Result 41
4.1.1 Dashboard MIT app inventor 41
4.1.2 Sensor monitoring system 42
4.1.3 Relay control system 43
4.1.4 ESP32 Camera Module Result 46
4.2 Analysis 47
4.2.1 ESP32 camera colour detecting analysis 47
4.2.3 Water & environment temperature sensor analysis 50
4.3 AutoCAD 3d casing Drawing. 52
4.4 Final 3d printed casing result. 53
CHAPTER 5 54
5.1 Conclusion 54
REFERENCES 55
APPENDICES 57

LIST OF TABLES

Table 2.0: Proposed project advantages compare to previous monitoring title paper. 17

Table 3.0: ESP8266 pinout configuration description 25

Table 3.1: Potential Camera Platform support of Key Attributes and Features 30

Table 3.2: DHT11 specification 32

Table 4.0: MIT control dashboard vs relay condition 45

Table 4.1: ESP32 cam view vs real life view 47

LIST OF FIGURES

Figure 1.0:	Aquaculture sector seed yield 2022 results.	2
Figure 1.1:	Ornamental fish 2022 yield	3
Figure 1.2:	News articles cover zooplankton breeder success.	4
Figure 1.3:	Global revenue fish and seafood for 2018 year to expected 2028 yield	5
Figure 2.0:	Global Price per Unit fish and seafood for 2018 year to expected 2028 yield	8
Figure 2.1:	Malaysia IMF Primary Commodity Price Indices from 2005 to 2023	8
Figure 2.2:	a Newborn D. magna. **b** Adult D. magna. **c** Adult D. longispina. **d** Adult D. pulex. D. magna and D. pulex[	9
Figure 2.3:	Number of brackish waters, ornamental fish, and freshwater farmer in Malaysia from 2021 to 2022	11
Figure 2.4:	Number of freshwater farmer, brackish waters, seaweed system, aquatic plant and ornamental fish farmer and breeder in Malaysia	11
Figure 2.5:	Shows a weight of fish vs year.	12
Figure 2.6:	IoT infographic	13
Figure 2.7:	Shows how ammonia entering system by fish gill.	15
Figure 3.0:	Overall methodology flow of the system	20
Figure 3.1:	Block diagram proposed project setup	23
Figure 3.2:	Schematic diagram project setup	24
Figure 3.3:	ESP8266 pinout configuration	25
Figure 3.4:	PH sensor	26
Figure 3.5:	PH scale colour effect on litmus paper	26
Figure 3.6:	Gas sensor to detect ammonia presence.	27
Figure 3.7:	Temperature sensor	28
Figure 3.8:	One wire bus system advantage	28
Figure 3.9:	ESP32 camera module	31
Figure 3.10:	TTL serial adapter	22
Figure 3.11:	DHT11 sensor	31
Figure 3.12:	Designer interface on MIT app inventor	34
Figure 3.13:	Blocks interface on MIT app inventor	35

Figure 3.14:	MQTT scenario	36
Figure 3.15:	Arduino IDE interface	37
Figure 3.16:	Flow chart for my proposed project	38
Figure 3.17:	Shows how do proposed project work theoretically.	39
Figure 4.0:	Dashboard MIT app inventor result	41
Figure 4.1:	Arduino IDE serial monitor results	42
Figure 4.2:	MIT app dashboard result same as serial monitor	43
Figure 4.3:	Relay output configuration	44
Figure 4.4:	MIT app camera search result	46
Figure 4.5:	Weighing daphnia magna using sensitive mini scale	49
Figure 4.6:	Daphnia magna population vs 30 days graph	50
Figure 4.7:	Environment and water temperature difference	50
Figure 4.8:	Solution PH value vs 30 days	51
Figure 4.9:	Filtered chlorella green water after daphnia occupied for 30 days	52
Figure 4.10:	Proposed project casing with dimensions illustrated by AutoCAD	52
Figure 4.11:	Latest product result	53

CHAPTER 1

INTRODUCTION

1.1 Background

Aquatic systems, including oceans, rivers, lakes, and wetlands, are essential for biodiversity, climate regulation, and human activities. The health of these systems depends on a mix of physical, chemical, and biological factors. Important physical factors include temperature, light penetration, water flow, and depth, which influence metabolic rates, photosynthesis, nutrient distribution, and habitat suitability. Chemically, dissolved oxygen (DO), pH levels, nutrient concentrations, and salinity are critical. DO supports aquatic respiration, pH affects chemical reactions and toxicity, and nutrient levels, particularly nitrogen and phosphorus, can cause overgrowth of algae if too high[1].

Biologically, ecosystem health relies on species diversity and interactions. Primary producers like phytoplankton form the base of the food web, while consumers and decomposers keep the ecosystem stable. Human activities like pollution, overfishing, habitat destruction, and climate change greatly harm aquatic systems.[2] Pollution from agricultural runoff, industrial discharges, and plastic waste damages water quality, while overfishing and habitat destruction disrupt food webs and reduce biodiversity. Climate change makes these problems worse by changing temperatures, sea levels, and weather patterns. Tools like the Water Quality Index (WQI) and bioindicator species, such as Daphnia, help monitor the health of aquatic environments.

Water fleas, or Daphnia, are small crustaceans that do well in water temperatures between 18 and 22 degrees Celsius. They filter-feed on algae and particles in the water, playing a key role as primary consumers in aquatic ecosystems. Because they are nutritious, Daphnia are used as live food in aquaculture and are important for research on toxicity and ecology. Watching Daphnia populations gives important clues about water quality and ecosystem health, as their presence and health show changes and pollution in the environment.

The Smart Hygiene Daphnia Detection System is a big step forward for Daphnia breeders, helping them save money and work more efficiently. This system constantly checks water quality to ensure the best breeding conditions, preventing wasted resources on bad breeding attempts. By keeping the tank conditions ideal, breeders can adjust feeding schedules accurately, cutting down on waste and feed costs. The system also helps avoid losses from poor water quality, making the breeding process more efficient overall. This technology ensures a steady and productive supply of Daphnia, making it a cost-effective and sustainable way to provide nutritious live food for aquaculture and support healthy aquatic ecosystems.

1.2 Motivation

The conventional approaches faced challenges in maintaining ideal conditions for Daphnia populations, impacting cleanliness. To address this, the Smart Hygiene Daphnia Detection System, utilizing modern technologies and real-time monitoring, emerged as a solution. Two important industries that depend on daphnia for support is freshwater fish hatcheries and ornamental fish. Figures 1.2 & 1.3 below shown how importance this zooplankton to Malaysia in the freshwater fish hatcheries and ornamental fish industry where Daphnia is an essential live food, where a wide variety of vibrant fish are raised for aquariums and aesthetic uses [3].

Figure 1.0: Aquaculture sector seed yield 2022 results.

Figure 1.1: Ornamental fish 2022 yield

This innovative project aims to enhance the overall effectiveness and hygiene of Daphnia breeding conditions. Given the significance of Daphnia as a crucial live food source for planktivorous fish, the system contributes to the growth and health of species like small fish and invertebrates. The high nutritional value[3] and ease of cultivation make Daphnia highly admired by breeders.

Artificial intelligence is applied to identify changes in watercolor, which makes it easier to identify when daphnia are blooming. By ensuring real-time monitoring, this modern equipment optimizes cleanliness for Daphnia populations. A crucial aspect of the project involves creating user-friendly dashboard that combines important data and criteria. With its thorough perspective, this dashboard gives breeders the tools they need to monitor Daphnia populations effectively. The technology improves the general health of planktivorous fish reproduction by quickly adapting to changes in water conditions and creating an environment that promotes ideal growth and nutritional balance.

1.3 Problem Statement

Nowadays, breeding zooplankton, such as Daphnia, offers a decent return. Many individuals have transitioned to breeding due to the lucrative potential. However, the process is challenging because zooplankton, particularly Daphnia, are highly sensitive to unhygienic environments. Recognizing the benefits of breeding this species has led to the development of a device aimed at facilitating this process. Malaysia has previously faced food crises, highlighting the critical need for food security, especially regarding marine food sources. Freshwater fish hatcheries rely heavily on Daphnia as a primary food source, underscoring the importance of maintaining their populations in a healthy and hygienic environment. This makes the development of effective breeding and monitoring systems essential for supporting food security initiatives.

Figure 1.2: News article cover zooplankton breeder success.

In 2022, Malaysia aims to have around 20,925 registered aquaculture farmers, totaling nearly 22,000 farmers. Among them, 15,945 are focused on freshwater fish farming, 3,423 on brackish water fish farming, and 926 on ornamental fish farming. Specifically in Malacca, the number of breeders has risen to 4,465, marking an increase of 100 breeders from the previous year[4]. This data highlights the significant presence of aquaculture activities in Malaysia, showcasing the diverse range of fish farming endeavors across different regions of the country.

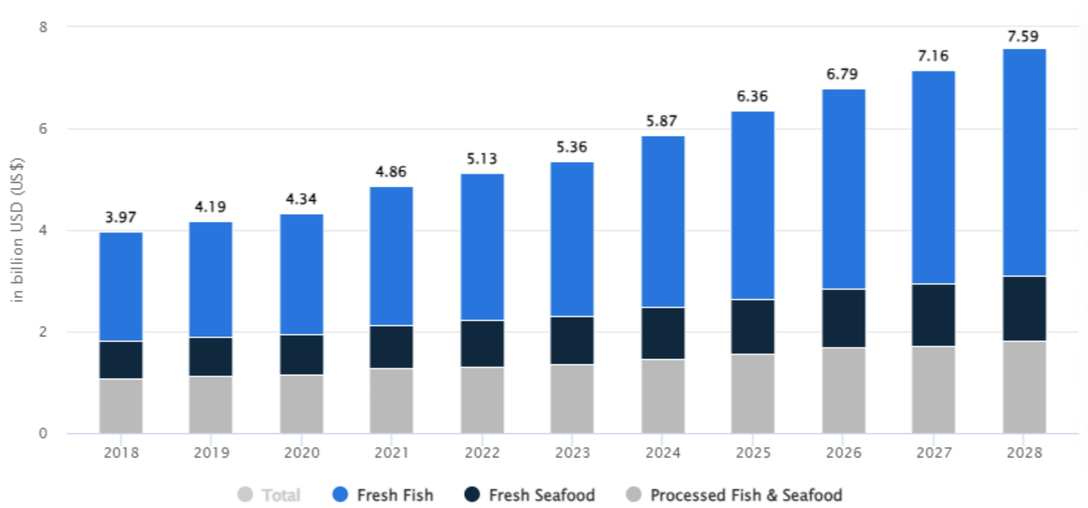


Figure 1.3: Global revenue fish and seafood for 2018 year to expected 2028 yield.[5]

The Fish & Seafood market is projected to reach a revenue of US$5.87 billion in 2024, with an anticipated annual growth rate of 6.64% (CAGR 2024-2028). Notably, China leads the global market, generating a substantial revenue of US$97 billion in 2024. In terms of per capita revenue, each person is expected to contribute US$172.80 to the market in 2024. Looking at volume, the Fish & Seafood market is forecasted to reach 375.70 million kilograms by 2028, showcasing a 5.3% growth in 2025. On an individual level, the average volume per person in the Fish & Seafood market is estimated to be 9.5 kilograms in 2024.

1.4 Objective

1) To develop hardware a device that monitor the temperature, ammonia gas level and pH levels of the water during the Daphnia breeding process.

2) To use the dashboard to monitor water quality to increase productivity.

3) To detect and monitor the presence of the green water for daphnia and blooming of the daphnia using camera.

1.5 Scope

This project aims to approach to aquaculture management that goes beyond the traditional method of water monitoring. The proposed project is intended to identify significant color changes in the water, with the main goal being the effective development of Daphnia populations. The ESP32 camera will notifies breeders via telegram when the water changes from its original green state, indicating the successful Chlorella culture combination. This alert breeder to include the daphnia in the tank and starting the breeding process. Breeder can also stream an online situation and control 4 importance element for the tank such as to turn ON or OFF heat lamp, air bubbler, solenoid valve and water pump by using developed app. The device keeps an eye on the water until it notices the desired transition from green to light red, which indicates a perfect environment for a Daphnia bloom. Besides that, the device also aims to monitoring daphnia tank environment parameter like pH value, humidity, water temperature and ammonia presence in the tank. This system also aims to help freshwater fish hatcheries industries and planktivorous fish breeders in Melaka, Malaysia.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The development of a Smart Hygienic Daphnia Detection System is important because Daphnia play a key role in monitoring water quality. Traditional methods, such as manual counting and microscopy, are labor-intensive and often inaccurate. This review covers the evolution of Daphnia detection from these traditional methods to modern automated systems that use optical, imaging, and sensor-based technologies. The integration of machine learning has further improved these systems, enabling real-time monitoring. Ensuring hygienic design is crucial to prevent contamination and maintain accuracy. This review discusses hygiene strategies, cleaning protocols, and real-world applications of these systems. It also addresses current challenges, technical issues, and future research directions. By examining the pros and cons of different detection methods and the impact of smart technologies, this review aims to provide a clear understanding of the current state and future potential of smart hygienic Daphnia detection systems for effective environmental monitoring.

2.2 Seafood Rising Trend

The graph from 2018 to 2028 shows the global trend in seafood prices, which is expected to continue rising. It shows a constant rise. The data given by the Malaysian Department of Fisheries indicates that Malaysia is not an exception to this trend. The data from the IMF Primary Commodity Price Indices, which covers the period from 2005 to 2023, emphasises the upward trend in seafood prices even more[5]. The cost of seafood is continuing to rise, as seen from both a national and international standpoint, with possible consequences for consumers and industry participants.

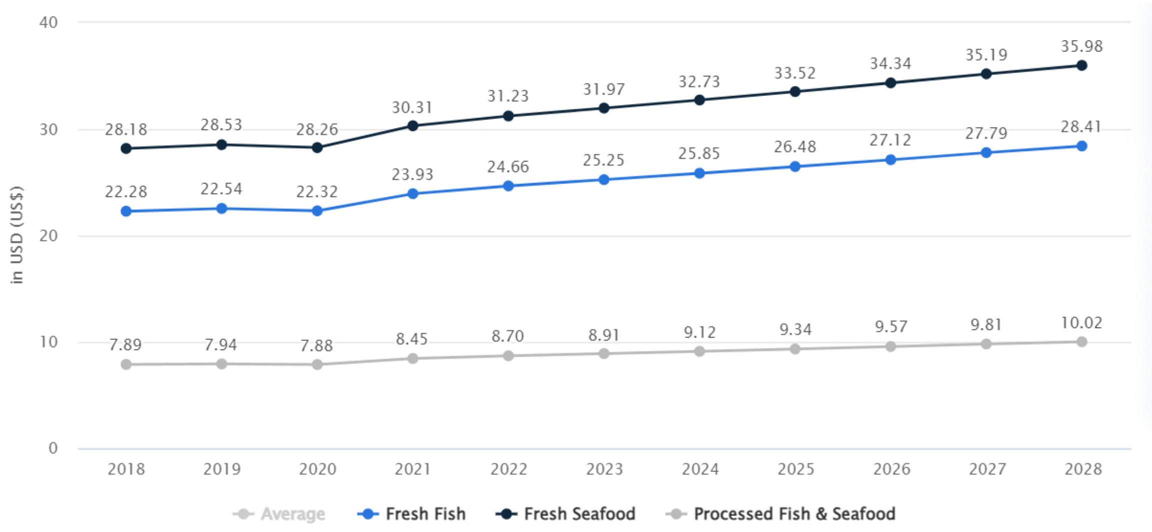


Figure 2.0: Global Price per Unit fish and seafood for 2018 year to expected 2028 yield.[5]

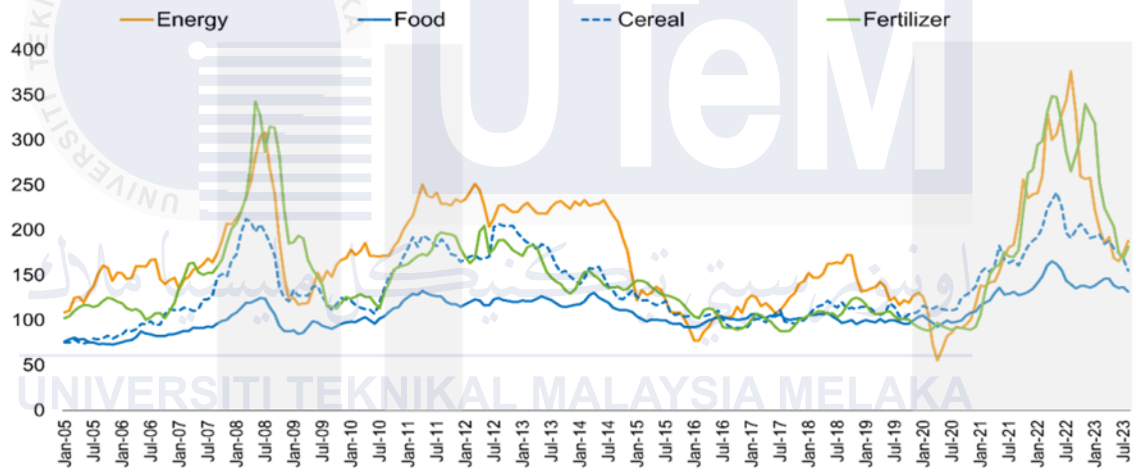


Figure 2.1: Malaysia IMF Primary Commodity Price Indices from 2005 to 2023[6]

The "Hygiene Daphnia Detection System" addresses the need for efficient water quality monitoring in aquaculture, specifically focusing on the breeding environment of Daphnia. As the prices of seafood increase, aquaculture practices become more economically significant, making the sustainability and efficiency of these practices' paramount. The detection system, equipped with advanced sensors and artificial intelligence algorithms, contributes to maintaining a healthy breeding environment for Daphnia, which are essential in the broader aquatic ecosystem.

2.3 Planktonic Crustaceans

Daphnia, a genus of small planktonic crustaceans, are widely distributed in freshwater environments like ponds and lakes. They are crucial in aquatic ecosystems, serving as primary consumers and filter-feeding on small particles, especially unicellular algae. With over 100 species, Daphnia are characterized by a large head with a simple compound eye and a body encased in a bivalve-like shell. They are transparent to avoid predators and dodge several times before and after reaching maturity. Newborn Daphnia resemble adults but lack the developed dorsal brood pouch.[7]

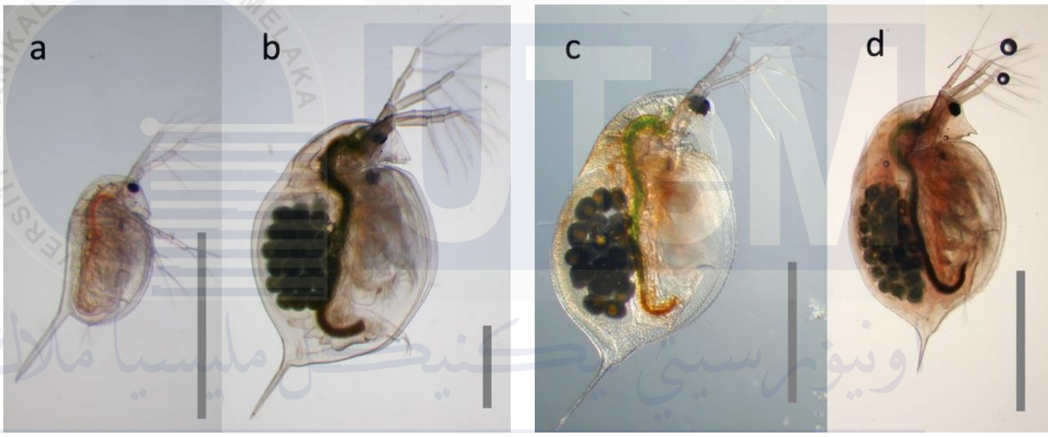


Figure 2.2: a Newborn D. magna. **b** Adult D. magna. **c** Adult D. longispina. **d** Adult D. pulex. D. magna and D. pulex[7]

Daphnia are typically found in standing freshwater, from small pools to large lakes. Although they can colonize saltwater lakes or estuaries, they rarely inhabit seawater. They play a key role in aquatic food webs, acting as prey for fish and invertebrate predators. Daphnia can reproduce asexually through parthenogenesis, allowing them to propagate quickly under favourable conditions.

When environmental conditions worsen, they switch to sexual reproduction, producing males and haploid eggs that require fertilization. This reproductive flexibility helps Daphnia populations survive in changing environments.

2.3.1 Daphnia Hatcheries System

In Daphnia culture manual and traditional method, breeder will be collecting Daphnia in a bunch of tanks from nature such as pond, sewage, dan ditch. Each tank gets food at different times to grow algae, which the Daphnia eat. First, breeder put a pure Daphnia culture in the tanks, and as they grow, they eat algae, bacteria, and other tiny stuff. After 7 days, breeder will collect the Daphnia, and then use the tanks for a new batch[8].

The old way of taking care of daphnia had a problem because for example, if one person had to look after ten tanks of daphnia, it was a lot of work. Each tank needed careful attention to make sure everything was just right for the daphnia. If there are not enough people to help, it could not be able to give each tank the care it needed. This could also affect how many daphnia were produced and their quality[9].

As a result of this issue, proposed to develop an innovation known as the "smart hygienic daphnia detection system". One of the unique functions of this system is to be monitoring the water in the tanks. When the water turns from green, the daphnia should be added to the tank and allowed to grow well and when the daphnia turn to light red, notification will be sent to breeder telegram app that mean the daphnia are blooming and ready to feed fish. This is detected by a camera. In addition, the system monitors critical parameters such as the water's pH, temperature, and ammonia concentration in the tank. This system is designed to help daphnia and fish breeders in Melaka, Malaysia, making it easier to take care of the tanks and get more daphnia.

2.4 Improvements in Aquaculture in Malaysia

In 2021, there were 21,226 fish farmers in Malaysia involved in freshwater, brackish water, and ornamental fish farming. However, the following year saw a decrease in the number of farmers, dropping by almost a thousand. Several factors could contribute to this decline, including changes in market demand, economic factors, or challenges in fish farming practices.

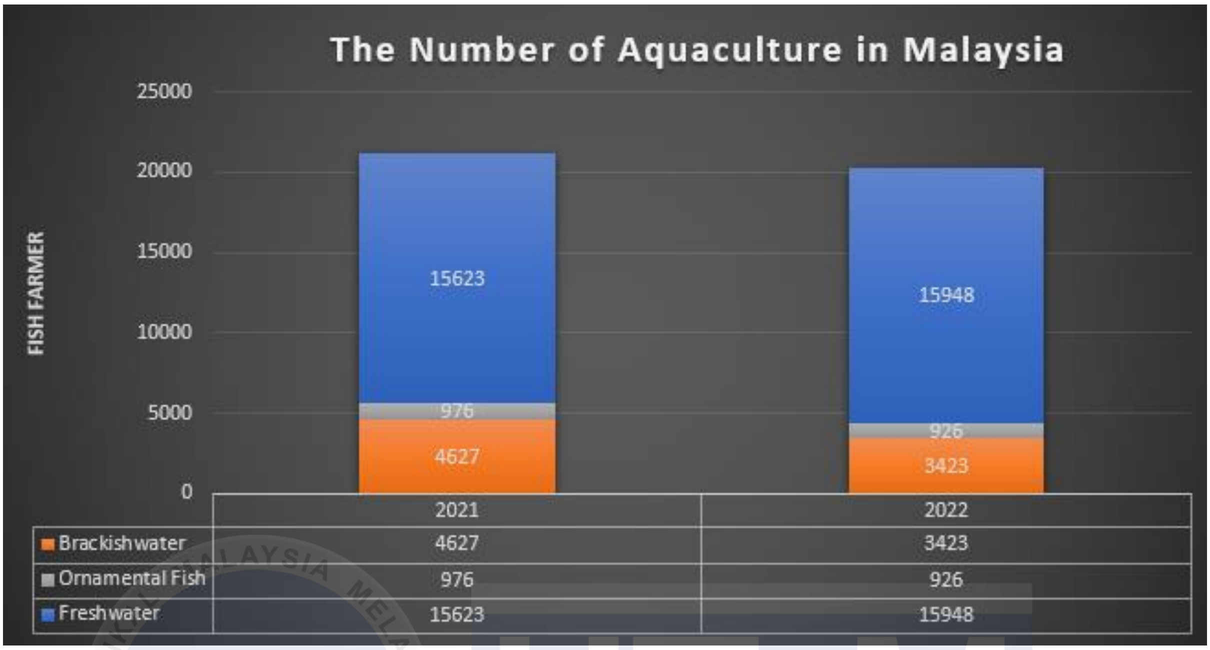


Figure 2.3: Number of brackish waters, ornamental fish, and freshwater farmer in Malaysia

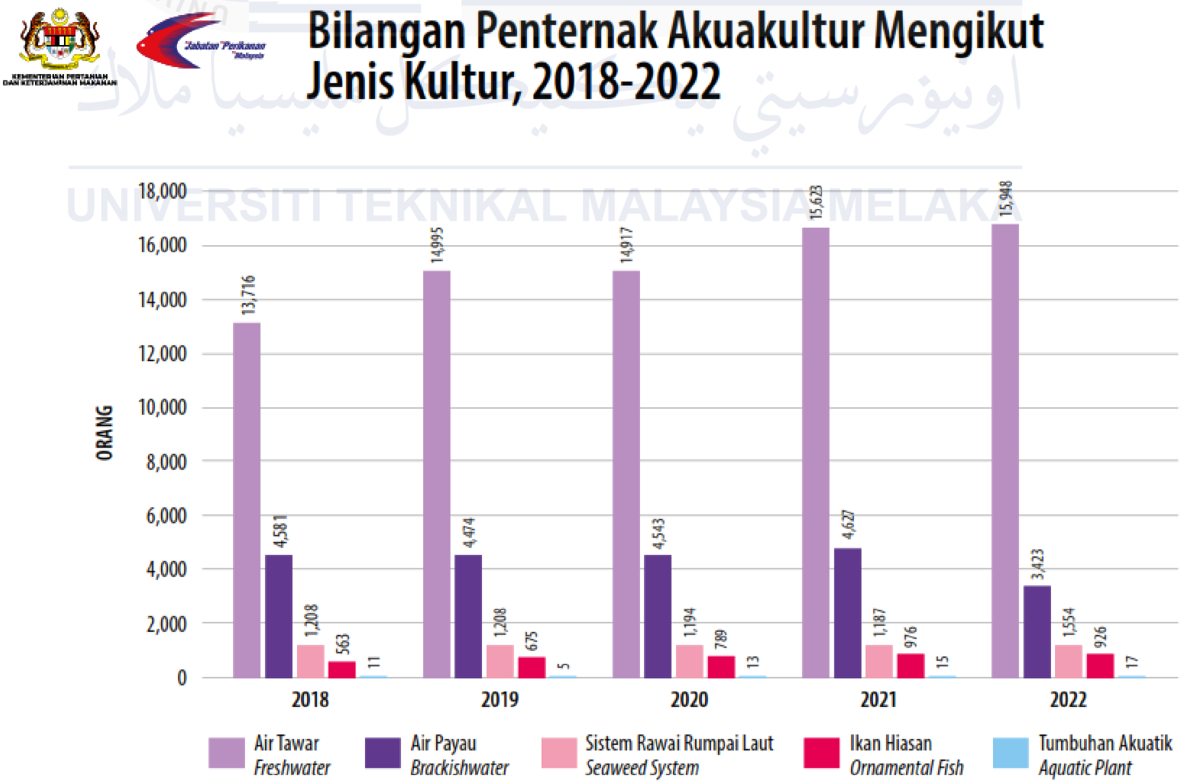


Figure 2.4: Number of freshwater farmer, brackish waters, seaweed system, aquatic plant and ornamental fish farmer and breeder in Malaysia [4]

In 2021, there were 21,226 fish farmers in Malaysia involved in freshwater, brackish water, and ornamental fish farming. However, the following year saw a decrease in the number of farmers, dropping by almost a thousand. Several factors could contribute to this decline, including changes in market demand, economic factors, or challenges in fish farming practices.

To address the challenges faced by farmers and breeders, a new concept called the “smart hygienic daphnia detection system”. This device is designed to assist farmers and breeders in maintaining their live food stock for freshwater fish hatcheries and ornamental fish. By providing real-time monitoring and smart detection of water conditions, the system aims to enhance the efficiency of fish farming operations, ensuring a healthier environment for the live food stock, and ultimately supporting the growth of the aquaculture sector.

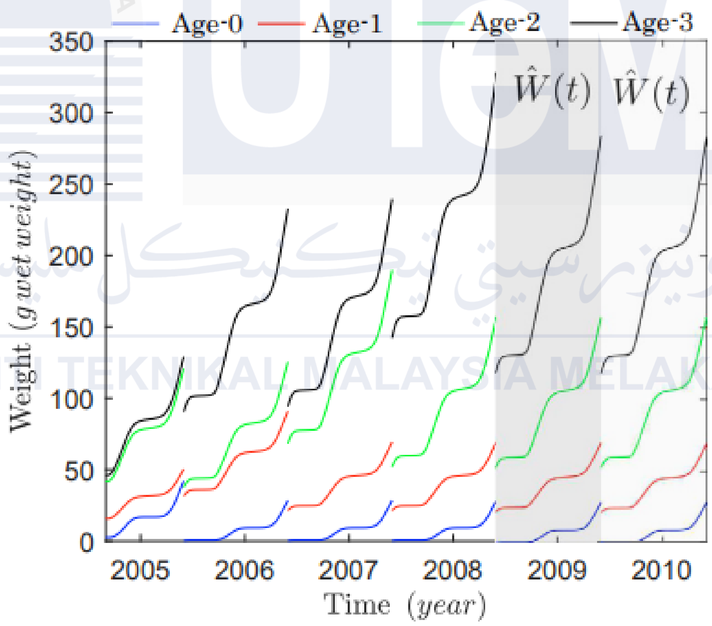


Figure 2.5: Shows a weight of fish vs year.

This paper [10] and table 2.2 shown prove that daphnia is very importance to freshwater habitat where they aiming to increase water transparency and reduce harmful algal blooms by removing planktivorous fish on the ecosystem but end up remaining fish gain an extreme weight and healthy.

2.5 Daphnia life needs.

To ensure Daphnia are healthy, and able to breed, it is important to maintain ideal pH levels, stable water conditions, proper temperatures, cleanliness, and a constant supply of food. These elements provide a healthy atmosphere and increase their metabolic activity. Monitoring their life conditions is important for accurate water quality analysis and to the keeping of a healthy aquatic community in proposed projects due to as a healthy life food.

2.6 Internet of Things (IoT) in Hygiene Monitoring

Figure 2.6: IoT infographic

To increase production of the daphnia magna, we need to have one device call IoT system devices. The Internet of Things (IoT) is a system that makes common items smart and connected by combining technologies like mobile computing, sensing, and communication [11]. For example,[12], this paper shown how the author to solve eliminate hunger and poverty while also ensuring the sustainability of agriculture and food systems by developed smart automated indoor hydroponics and aeroponics, however this system has a disadvantage of measuring only the water level and the temperature of the nutrient solution.

In this case, it completely transformed hygiene monitoring by linking devices to the internet and offering real-time data collection, analysis, and management. This innovative method monitors temperature, gas and water quality, and other hygienic indicators using sensors provided by the Internet of Things (IoT). These sensors continuously collect data, which is later sent to cloud-based platforms for keeping and analysis. Via connectivity with other smart systems, mobile accessibility, and user-friendly interfaces, the overall user experience had been improved. For example, breeder can reduce in terms of transportation logistically because they can directly monitor through their mobile phone with is encourages user to contribute to maintaining safety and clean.

2.7 Hardware Instrument of Daphnia Needs.

Based on this proposed project, aquatic creature needs a sensitive care which could take their life if breeder made wrong simple mistake. So proposed project could tackle the problem with combination of sensor, relay and camera.

2.7.1 (Potential of Hydrogen) PH level

A PH sensor, the DIYMORE Liquid pH Value Detection Sensor & BNC Electrode PH Probe, because daphnia consume a pH level between 7.2 and 8.5. This decreases the difficulty breeders experience in keeping Daphnia's pH at the proper level. Breeders can help maintain the pH in the desired range by using the sensor's continuous monitoring feature. In these conditions, breeders may more easily take good care of the Daphnia and they will have the ideal habitat to maintain their health.

2.7.2 Ammonia Gas

MQ-135 Gas Detector Sensor is used to meet Daphnia's requirements, particularly regarding ammonia present. This takes care of a common issue that breeders have regarding keeping clean tank water.

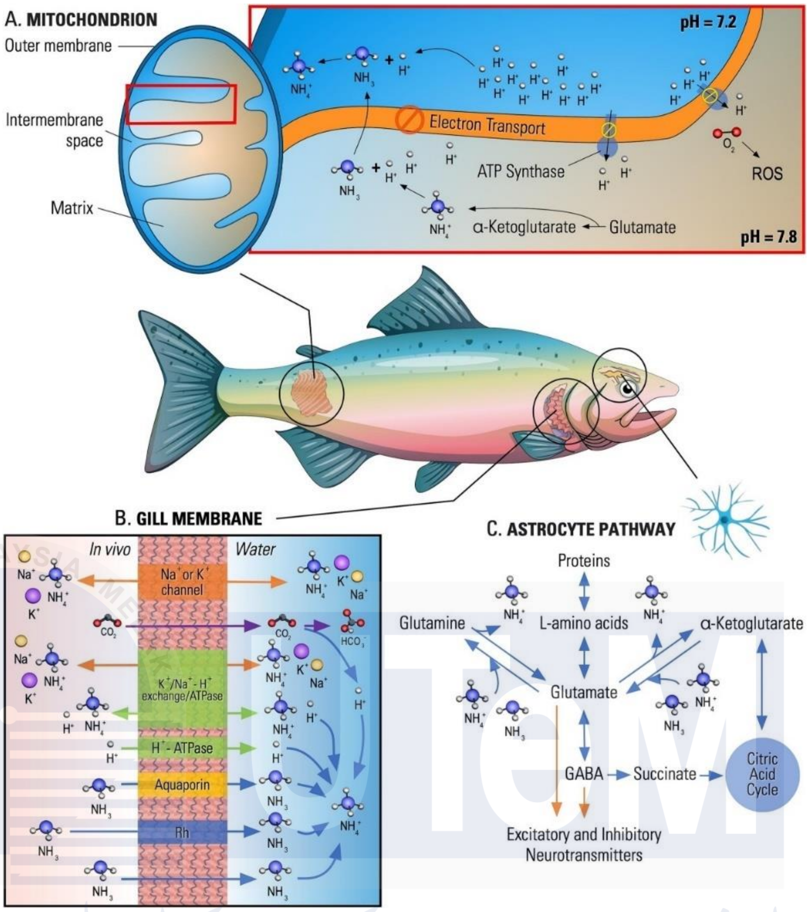


Figure 2.7: Shows how ammonia entering system by fish gill.

Two additional forms of "ammonia" are anhydrous ammonia (NH_3) and ammonium hydroxide (NH_4OH). Anhydrous NH_3 is fertilizer in the form of ammonia gas (NH_3). Buildup of ammonia can cause stress, gill and internal organ damage, and eventually death[13]. Breeders can respond quickly and maintain a hygienic and secure environment for their Daphnia thanks to the sensor's real-time ammonia level detection and monitoring capabilities. By improving the general hygiene of the tank water, this modern approach helps breeders create ideal circumstances for Daphnia health and supports productive breeding operations. This paper [14] demonstrates the writer's advanced proficiency in forest fire detection using the ESP32 camera. Despite this, the project lacks a straightforward approach to detecting incoming carbon dioxide gas produced during a fire.

2.7.3 Water Temperature

Temperature sensor DS18B20 was apply due to Daphnia need temperatures in the range of 18 to 22°C. Breeders can benefit from this since it might be tough to maintain the ideal temperature for them. Breeder can continuously monitor the temperature and verify that it maintains within a comfortable range by using the DS18B20 sensor. The Daphnia can live and grow comfortably in these conditions, and breeding becomes simpler and more effective.

2.7.4 4-module relay

A 4-module relay has been applied in the system, which can assist breeders in improving aquaculture management techniques and supporting the production of high-demand food stocks. This relay allows for the control of four types of outputs: a heat lamp, an air bubbler, a solenoid valve, and a water pump. The heat lamp helps regulate water temperature, ensuring it remains within a suitable range for the zooplankton. The air bubbler provides necessary oxygenation, vital for the health and growth of Daphnia. The solenoid valve is used to introduce new water into the system, helping maintain water quality by diluting waste products and providing fresh nutrients. Conversely, the water pump removes dirty water from the system, preventing the buildup of harmful substances and ensuring a clean environment. These outputs are controlled via a developed app, allowing breeders to monitor and adjust conditions remotely. This capability enhances efficiency and effectiveness in aquaculture management, ultimately supporting the sustainable production of essential food stocks.

2.7.5 Visual Monitoring

ESP32 camera with Phyton color detection coding are selected because the ability to recognize when green water is present in breeding farm and to track the change from green to light red mature blooming Daphnia. Real-time monitoring of watercolor changes specifically, the indication of a successful Chlorella culture mix leading to Daphnia blooming made possible by the camera's visual monitoring capabilities.

When the water changes from its initial green color, the camera alerts breeder via telegram and will add Daphnia starters to start the breeding process. All of this paper[14] [15], [16], [17], [18], [19] focus on monitoring, data collection, and analysis abilities. However, none of these projects possess the capability to capture a visual situation.

2.8 Comparison of Previous Monitoring Title Paper.

Table 2.0: Proposed project advantages compare to previous monitoring title paper.

Ref	Reference title	Technique	Specification	Justification
[14]	IoT enabled advanced forest fire detecting and monitoring on Ubidots platform	Less humidity levels mean the forest fires occurring probability is high and high humidity means less probability of forest fires	ESP32 board rain sensors, sound sensor, DHT11 sensor, PIR sensor	This paper project missing an importance thing due to detecting fire which could achieve success more easily. He not including a gas detector to detect presence of carbon dioxide. This offers advantage compared to my proposed project which can detect presence of ammonia gas.
[15]	Hydroponic and Aquaponic Farming: Comparative Study Based on Internet of things IoT technologies.	The pH sensor measures the pH level of water in the cultivated basin and directly sends the data to the ESP32 microcontroller.	Arduino Uno R3. Raspberry Pi 3 Model B pH meter. Litmus paper. Electrical Conductivity	Even though the papers on hydroponics are

		Conventionally, plants require a pH degree of 6.5 to 7 in the hydroponics-cultivated basin. Therefore, if the pH rises above 7°, the ESP32 microcontroller will instruct the acid pump to add phosphoric acid to reduce the acidity to a reasonable level. shows the block diagram of the main structure of the developed monitoring hydroponics system.	Sensors. Bread Board	titled monitoring and security, it's interesting that none of them include any visual monitoring. This offers a clear benefit for my project, since mine includes an ESP32 camera to specifically recognize the light red color that show mature Daphnia in my hydroponic system and detect the presence of Chlorella green water with artificial intelligence support.
[16]	SMART GROW Low-cost automated hydroponic system for urban farming.		Espressif ESP32 development board, PH4502C pH sensor DFR0300 EC sensor, DS18B20 temperature sensor, JSN-SR04T ultrasonic sensor	
[17]	Smart Farming using IoT, a solution for optimally monitoring farming conditions.		ESP32s Node MCU, breadboard, DHT11 Temperature and Humidity Sensor, Soil Moisture Sensor, SI1145 Digital UV Index / IR / Visible Light Sensor	
[18]	Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things.		ESP32 microcontroller with total dissolved solids TDS, pH, water level, and temperature sensors	

[19]	Door Security System for Home Monitoring Based on ESP32	Android using Internet of Things (IoT) technology to monitor the status of the door, controlling the door and increasing security in a house. MQTT cloud is utilized as the communication protocol between smartphone and door lock system	Step Down 12v to 5v, Esp32, Pcb board Button reset, PIR sensor, Magnetic sensor, Internal touch sensor, Alarm buzzer	

CHAPTER 3

METHODOLOGY

3.1 Introduction

The "Smart Hygiene Daphnia Detection System" is a practical tool designed to help monitor key factors that are essential for successful Daphnia breeding. This system keeps an eye on important conditions like pH levels, temperature, humidity, and ammonia levels, all in real-time. It includes hardware with advanced sensors that constantly check these conditions to make sure they stay within the ideal range. The main goal of this system is to assist breeders in maintaining the best possible water conditions, which is crucial for the health and reproduction of Daphnia. By using this system, breeders can more effectively manage their aquaculture operations. This means they can quickly spot and fix any problems with the water conditions, improving the chances of successful Daphnia breeding. Overall, the "Smart Hygiene Daphnia Detection System" helps make Daphnia breeding more efficient and reliable, benefiting both the breeders and their aquaculture practices.

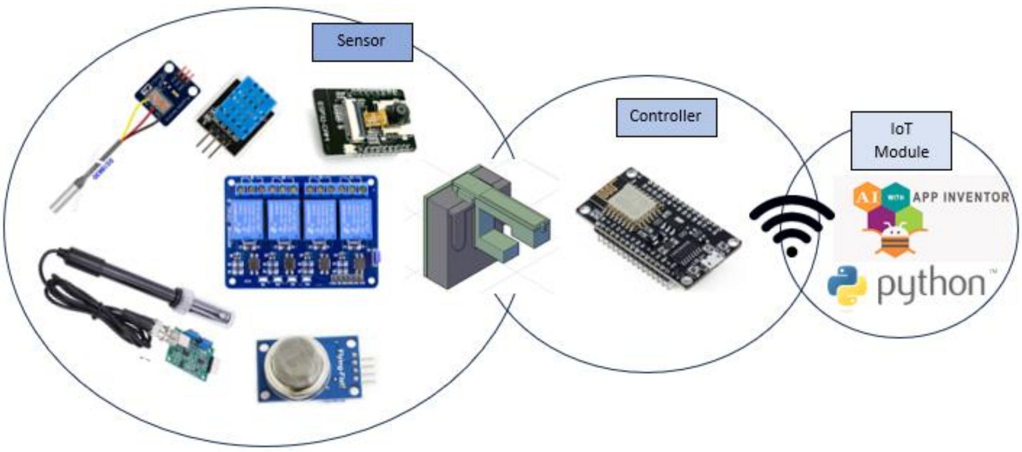


Figure 3.0: Overall methodology flow of the system

3.2 Methodology

For the hardware development of this system, microcontroller ESP8266 along with various sensors will be used to monitor the environmental conditions. To display and visualize the output, utilize MIT App Inventor as the IoT platform. This platform allows to create a user-friendly interface where the data collected by the sensors can be easily accessed and understood in real-time. The combination of the ESP8266 microcontroller and the MIT App Inventor ensures that the system is both efficient and accessible, providing breeders with the information they need to maintain optimal conditions for Daphnia breeding.

3.3 Gather data on Microcontroller.

The process of collecting data from various sensors, analyzing it, and sending it to a dashboard depends on the employment of a microcontroller, like the widely used ESP8266 in Internet of Things applications. The pH sensor, temperature sensor, gas detector sensor, humidity sensor and ESP32 camera are among the sensors that are connected to microcontroller pins. The installation of required libraries or drivers is done to facilitate communication, and the microcontroller is configured to periodically retrieve data from these sensors. Processing of the unprocessed sensor data may include steps like calibration or digital value conversion. When the microcontroller is fitted with a communication module such as Wi-Fi, it uses the credentials that have been supplied to establish a connection to a network. When it is an IoT platform or a custom web application, the dashboard must be configured to properly receive, process, and display the incoming data. Security measures, such as the use of secure communication protocols, are implemented to protect data during transmission. The smooth integration of sensor data into the dashboard is ensured by the programming, as demonstrated with the ESP8266 in the Arduino IDE. The establishment of a thorough monitoring and visualization system for important parameters like temperature, pH, humidity, and gas levels depends heavily on this integration.

An ESP32-CAM module is powered by the 5V and ground pins of an ESP8266 microcontroller. The ESP32-CAM streams live video based on programming done through the Arduino IDE, which provides the camera's IP address. This IP address is then used in a Python script designed for color detection. The Python script processes the video feed to detect different colors. Once the color detection is implemented, the Python script provides an IP address that can be accessed online, enabling live streaming with integrated color detection features. This setup allows for remote monitoring and color analysis of the streaming video, making this device can differentiate between chlorella green water and blooming daphnia. Also, An ESP8266 microcontroller is used to control a 4-module relay system, allowing breeders to manage high voltage 220V outputs through their smartphones. The system can control various elements essential for maintaining an optimal environment for Daphnia. These elements include a luminance heat lamp for providing necessary light and warmth, an oxygen bubble generator for ensuring adequate aeration in the aquarium, a solenoid valve for regulating the inflow of clean water, and a water pump for expelling dirty water. By integrating these components, the system ensures that the Daphnia are kept in a stable and suitable environment, which is crucial for their health and productivity. This setup demonstrates how microcontroller technology can automate and improve the management of aquaculture systems, providing breeders with a convenient and efficient means to monitor and control environmental conditions remotely. This automation improves the living conditions of the Daphnia and reduces the manual effort required in maintaining the daphnia tank.

3.4 Hardware Setup

The hardware infrastructure used is crucial to the success of this project's investigation. Each piece of hardware plays a unique and crucial part in advancing the research enquiry. The figure shows the exact hardware components used in this project, which were carefully selected to enable the creation and realization of the study objectives. These components include various sensors, a microcontroller ESP8266, and other necessary devices that work together to monitor and collect data on environmental conditions. The careful selection and integration of these hardware components ensure that the system functions effectively and provides accurate, real-time data for the study. This setup is essential for achieving the project's goals and ensuring reliable and meaningful results from the investigation.

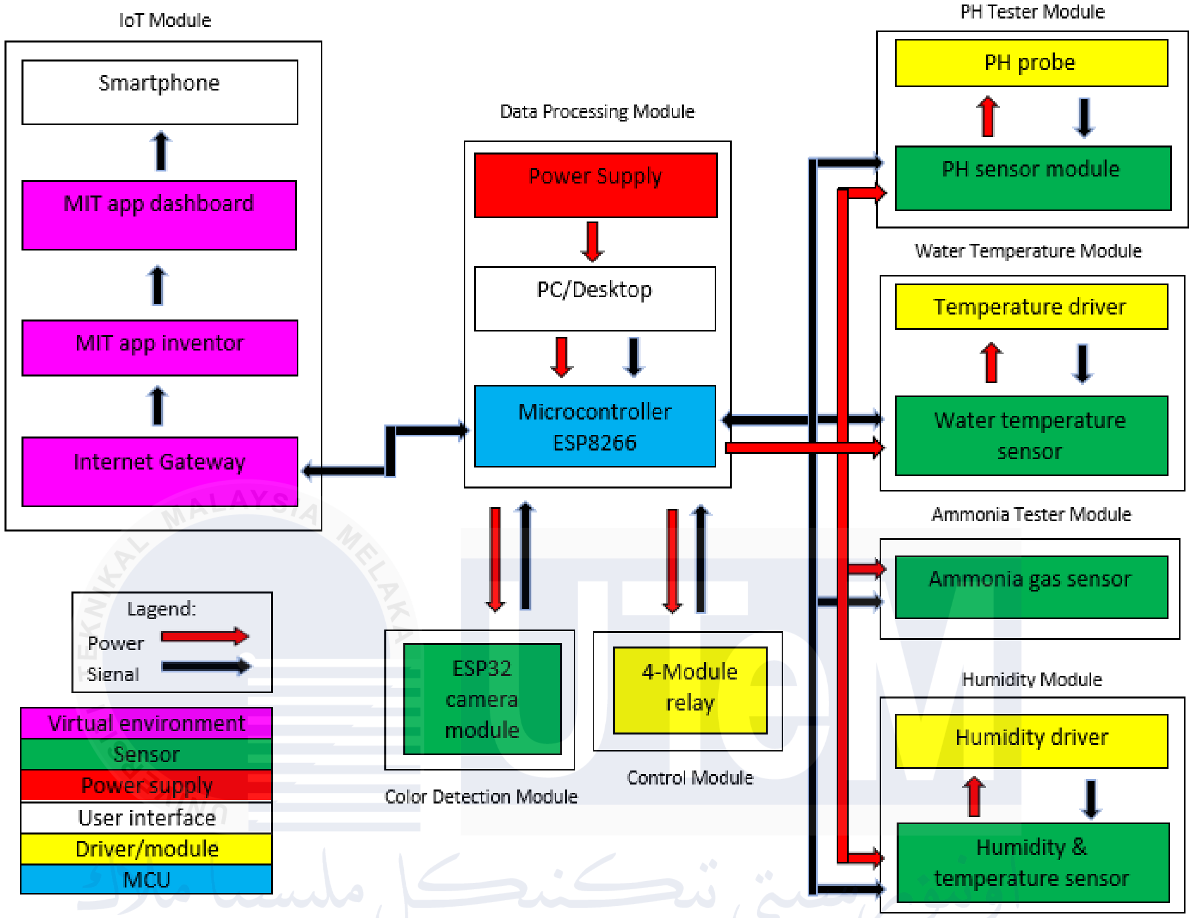


Figure 3.1: Block diagram proposed project setup

3.5 Hardware requirement

The figure displays the hardware connectivity arrangement for this system in detail. The central component of the system is the ESP8266 microcontroller, which connects the hardware to the IoT cloud server using the MIT App Inventor platform. This microcontroller manages communication and data transfer between various sensors, the camera, relay modules, and the cloud server. The ESP8266 facilitates real-time monitoring and control by enabling the transfer of sensor data and control signals through the IoT network. Additionally, the ESP32 camera module is used to color change detection, playing a vital role in monitoring the changes.

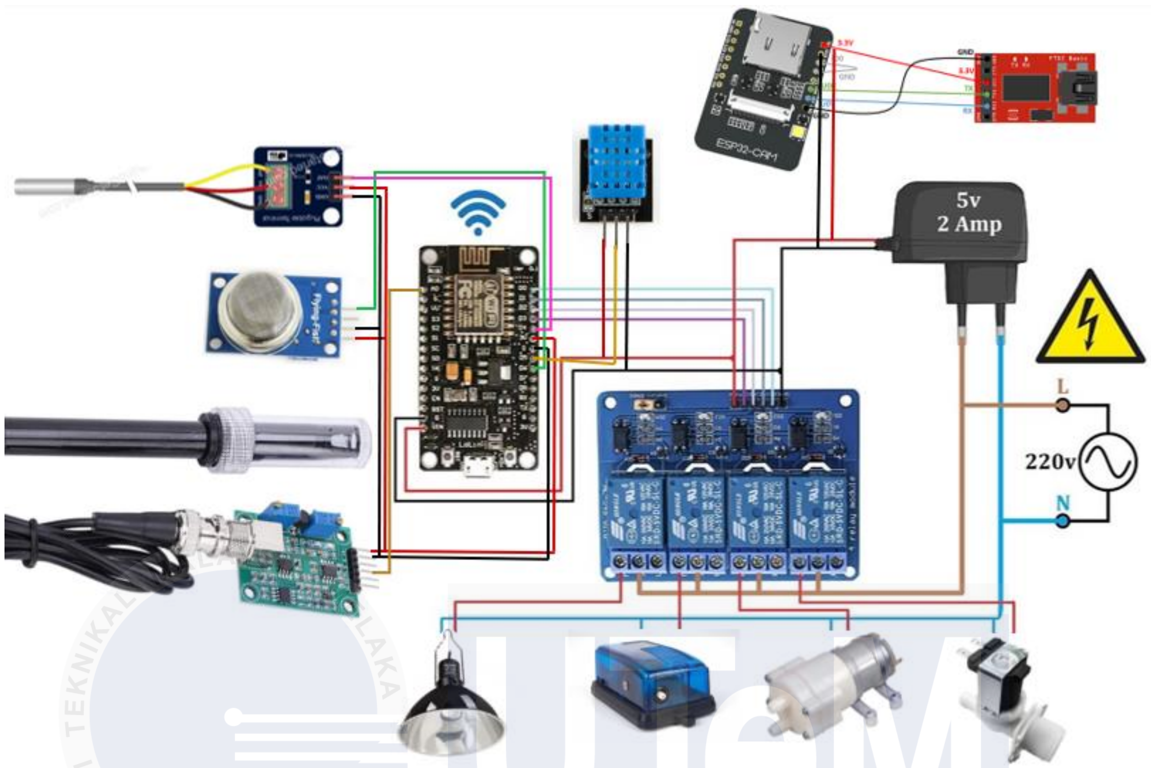


Figure 3.2: Schematic diagram project setup

3.6 ESP8266 Microcontroller

Versatile microcontroller and system-on-chip (SoC). With its dual-core processor, integrated Wi-Fi and Bluetooth connections, low power consumption, and a wide range of peripherals, the ESP8266 is a popular choice for a variety of applications, including wearables, home automation, and Internet of Things devices. Its compatibility with the Arduino IDE facilitates programming, and it gains from an engaged community of open-source software users.

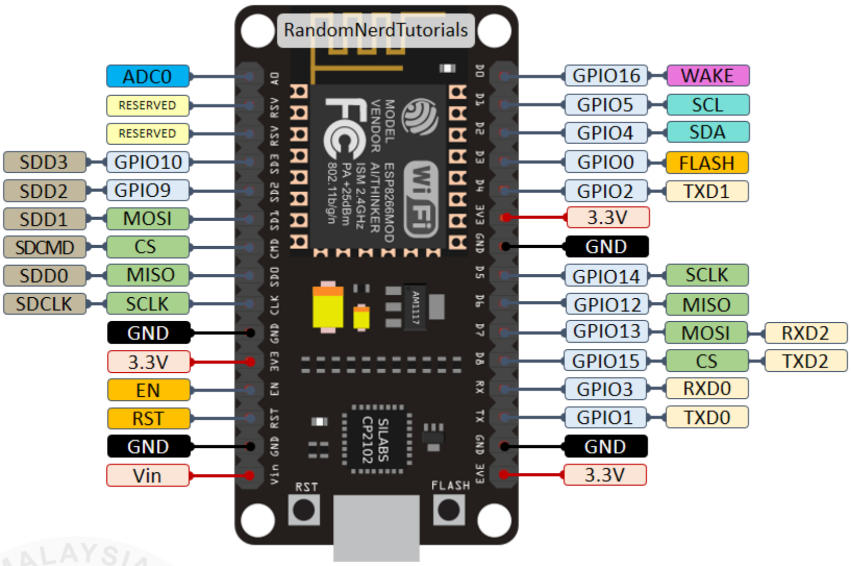


Figure 3.3: ESP8266 pinout configuration

Table 3.0: ESP8266 pinout configuration description

Content	Pin	Description
Power	Micro-USB, 3.3V, GND, Vin	These are used to power external components or sensors.
Control Pins	EN, Boot	To enable pin. It is used to reset the microcontroller.
Analog Pins	A0	Capable to reading analog signal, making them suitable for interfacing with analog sensors.
GPIO Pins	GPIO1 to GPIO39 (Only 25 pin)	Provides numerous general-purpose input/output (GPIO) pins which can be used for various digital input/output functionalities.
UART Pins	TXD, RXD	This function is essential for programming the ESP32 via a USB cable and for communicating serially with a computer or other USB host devices.
12C PIns		ESP32 support 12C communication, which is widely used for interfacing with sensors and other peripherals.

3.6.1 DIYMORE Liquid PH Value Detection Sensor & BNC Electrode PH Probe

Figure 3.4: PH sensor

Hydrogen ions are common in most chemical reactions. pH stands for "power of hydrogen" and indicates the acidity or basicity of a solution, particularly in aqueous solutions.

$$H_2O = H^+ + OH^-$$

Equilibrium occurs in water between alkali (OH-) and acid H^+. pH determines the basicity or acidity of the substance. Mathematically, pH value can be written as

$$pH = -\log \alpha H^+$$

The acidity or basicity of a solution is indicated by its pH value at a given temperature. The pH range is typically between 1 and 14. A solution with a pH of 1 to 6 is acidic, a pH of 7 is neutral, and a pH of 8 to 14 is alkaline[20].

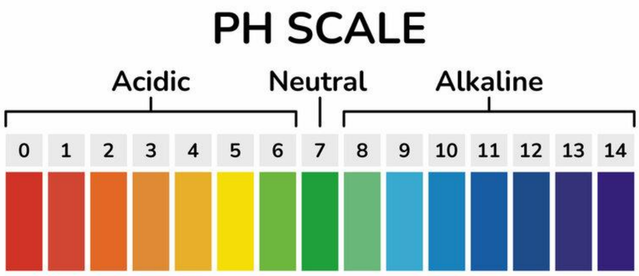


Figure 3.5: PH scale color effect on litmus paper

An indicator of a liquid's pH level, the DIYMORE Liquid pH Value Detection Sensor with BNC Electrode pH Probe measures the liquid's acidity or alkalinity. For precision readings, this sensor's BNC electrode pH probe can be dipped in liquid. Simple interface with electronic devices, such microcontrollers, is made possible by the BNC connector. It gets used in many different applications, such as hydroponics, water quality analysis, and environmental monitoring. The DIYMORE pH sensor is a useful tool for researchers, supporters, and experts working in water quality management since it offers an easy-to-use and accurate way to measure pH levels in liquids.

3.6.2 MQ-135 Gas Detector Sensor

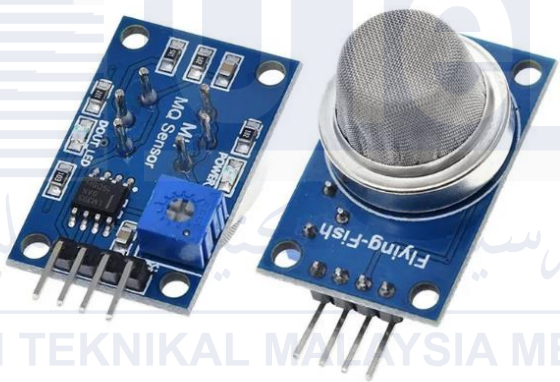


Figure 3.6: Gas sensor to detect ammonia presence.

A module was known the MQ-135 Gas Detector Sensor is made to identify and gauge different types of gases present in the atmosphere. This sensor is especially sensitive to dangerous contaminants and gases like smoke, benzene, and ammonia. It functions on the basis of resistance changing in reaction to various gases present. A sensitive semiconductor material in the module changes resistance to certain gases, which causes voltage variations. The MQ-135 sensor measures these variations to determine the presence of gases in the surrounding air. The MQ-135 sensor, which is frequently used in gas detection and air quality monitoring systems, is helpful in guaranteeing safety in a variety of environments, such as residences, workplaces, and labs.

3.6.3 DS18B20 Temperature Sensor

Figure 3.7: Temperature sensor

The DS18B20 is a digital temperature sensor that uses the 1-wire protocol. It can measure temperatures ranging from -55°C to +125°C (-67°F to +257°F) with an accuracy of ±0.5°C. The data it provides can range from 9-bit to 12-bit resolution, depending on the user's requirements. Because the DS18B20 follows the 1-wire protocol, it can be controlled with a single pin of a microcontroller. This feature simplifies the wiring and makes it suitable for applications where multiple sensors need to be connected to a single microcontroller. The 1-wire protocol also allows for power and data to be transmitted through the same wire, reducing the complexity of the circuit design[21]. Main advantage of 1 wire protocol is that we can control multiple 1-wire devices via single pin of Microcontroller.

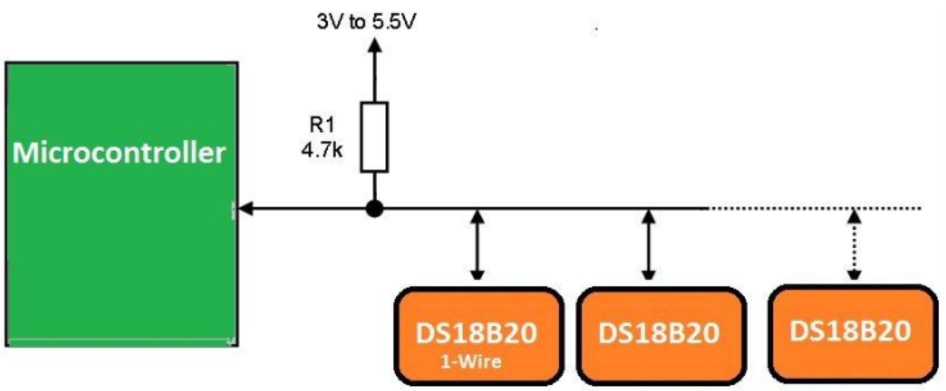


Figure 3.8: One wire bus system advantage

This sensor utilizes the One-Wire communication protocol, enabling multiple DS18B20 sensors to be connected to a single microcontroller pin. With a broad temperature range, the DS18B20 offers accurate temperature readings. It is a common choice for temperature sensing in many different electronic projects and systems due to its easy-to-use interface and adaptability. The sensor is commonly used in various applications, including environmental monitoring, industrial systems, and consumer electronics, due to its wide temperature range and ease of integration.

3.6.4 ESP32-CAM

Figure 3.9: ESP32 camera module

The ESP32-CAM module is designed primarily for building IoT (Internet of Things) devices, combining extensive programmability with a camera in a compact, low-power module. It is one of the smallest and least expensive options available, featuring a 2MP OmniVision OV2640 camera paired with a dual-core 32-bit processor. The module supports 802.11 Wi-Fi, Bluetooth, and various wired I/O interfaces, along with a microSD slot and low power modes. All these features are supported by the Arduino programming environment and a rich collection of open-source libraries.

Attribute or Feature	Mirrorless	Webcam	CHDK PowerShot	AI-Thinker ESP32-CAM
Image quality	≥20MP,≥12bpp	≥0.3MP,≥8bpp	20MP, 12bpp	2MP (1600×1200), 10bpp
Exposure control	Extensive	Basic	Extensive	Basic plus some features
Interchange lens	Yes	Some models	No	Simple modification
Sensor size	≥17.3×13mm	≥2.4×1.8mm	∼6×4.5mm	3.59×2.684mm
Near infrared (NIR)	Hard mod	Some models	Very hard mod	Simple mod
Wired connectivity	USB	USB	USB	UART, SPI, & I2C
Wireless	WiFi	No	Some models	WiFi & Bluetooth (with BLE)
Tethered control	Proprietary	Yes, UVC	Yes, CHDK PTP	Yes, programmable
Autonomous operation	Very limited	No	Yes	Yes
Programmable display	No	No	LCD	Options via connectivity
Programming support	No	No	CHDK C & Lua	Arduino & Espressif C/C++
Processor	Various ARM	?	Dual 80MHz ARM	Dual 240MHz Xtensa
Usable main memory	Varies	None	Several MB	520KB SRAM & 4MB PSRAM
Flash memory	SD card	No	SD card	4MB Flash & TF card
Power management	Minimal	No	Minimal	Modes from 310mA to 6μA
Sensor inputs	Camera UI	No	Camera UI	9 I/O pins; ADC, I2C, & SPI
Control outputs	Flash sync.	No	No	9 I/O pins; PWM, I2C, & SPI
Real time sync support	Remote	Some models	RTC, USB detect	RTC, programmable sync
Ease of embedding	Very hard	Moderate	Hard	Easy: 27x40.5x4.5mm board
Cost	≥ \$500	\$8 – \$150	≥ \$100	∼ \$7

Table 3.1: Potential Camera Platform support of Key Attributes and Features[22]

The ESP32 microprocessor can collect and process pictures and videos. By integrating a camera sensor with the microcontroller, the ESP32-CAM module effectively combines the camera's functionality with the robust capabilities of the ESP32, including Wi-Fi and Bluetooth connectivity. The camera sensor can capture videos and take still photos at different resolutions, making it versatile for various applications.

The ESP32-CAM module is widely used in applications that require visual data, such as image processing for IoT devices, smart home applications, and security systems. Its compact size and adaptable features offer a practical way to integrate visual capabilities into ESP32-based projects. The combination of powerful processing, connectivity options, and camera functionality makes it an excellent choice for developers looking to add visual data capabilities to their projects.

3.6.5 FT232RL TTL Serial Adapter

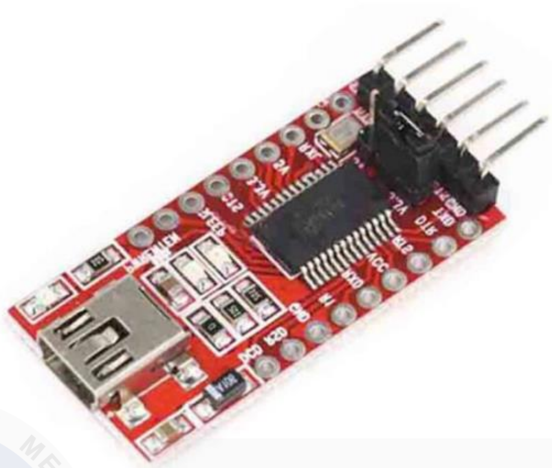


Figure 3.10: TTL serial adapter

An assistant for connecting your PC to small electronic components that interact via certain signals is the FT232RL TTL Serial Adapter. The FT232RL chip is used to convert the USB signals from the computer into the proper signals that these little parts can understand. Thus, techniques for communication allow simple connection and communication with objects such as microcomputers or other devices. For students working on electronics projects or understanding how devices communicate with one another, it is a useful tool.

3.3.6 Humidity sensor

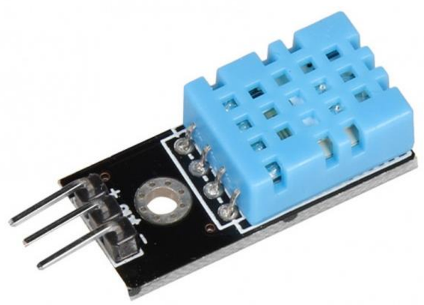


Figure 3.11: DHT11 sensor

The DHT11 sensor is an essential component used in this research for collecting environmental data. The DHT11 sensor is widely employed for measuring temperature and humidity levels, offering a cost-effective solution with reliable performance. It utilizes a digital signal output, making it compatible with various microcontrollers and IoT platforms. The DHT11 sensor operates within a temperature range of 0-50°C and a humidity range of 20-90%, with an accuracy of ±2°C for temperature and ±5% for humidity[23]. Its compact size and ease of integration make it ideal for applications where monitoring temperature and humidity are critical.

No.	Specification				
	Parameters	*Conditions*	*Min.*	*Typica l*	*Max.*
1	Humidity				
2	Resolution		1%RH	1%RH	1%RH
3	Repeatability			1%RH	
4	Accuracy	25'C		4%RH	
		0-50'C			5%RH
5	Interchange ability	Fully Interchangeable			
6	Measurement Range	0'C	30%RH		90%RH
		25'C	20%RH		90%RH
		50'C	20%RH		80%RH
7	Respond Time (Seconds)	1/e(63%)25 'C, 1m/s Air	6s	10s	15s
8	Hysteresis			1%RH	
9	Long-Term Stability	Typical		1%RH /year	
10	Temperature		1'C	1'C	1'C
11	Resolution		8 Bit	8 Bit	8 Bit

Table 3.2: DHT11 specification[24]

In this research, the DHT11 sensor plays a crucial role in ensuring that the environmental conditions for Daphnia cultivation are optimal. By providing accurate and stable measurements, the sensor enables researchers to gather valuable data for analysis and decision-making. The integration of the DHT11 sensor with the ESP8266 microcontroller ensures that real-time data can be monitored and controlled remotely, enhancing the efficiency of the system. The affordability and efficiency of the DHT11 sensor make it a suitable choice for environmental monitoring in various research studies, contributing to the overall success of the project.

3.7 Software Requirement

For the software requirements, Arduino IDE are used for programming the ESP8266 to enable communication with the ESP32-CAM, humidity sensor, MQ135 gas sensor, pH sensor, temperature sensor DS18B20, and the 4-relay module. The Arduino IDE provides a user-friendly environment for writing and uploading code to these microcontrollers, facilitating their interaction. Additionally, Python were used to program the ESP32-CAM for color recognition. The Python scripts teach the ESP32-CAM to detect color changes. When a specific color change is detected, the system sends a notification message to a Telegram bot. This allows for real-time alerts based on the color detection capabilities of the ESP32-CAM.

Furthermore, MIT App Inventor software to integrate all these elements into a cohesive application. The MIT App Inventor dashboard provides a platform where breeder can monitor the digital readings from the sensors, observe the visual color detection status, and control the 4-relay module. This integration allows for comprehensive monitoring and control of the system through a single, user-friendly interface. Overall, the combination of Arduino IDE, Python, and MIT App Inventor software enables the development of a robust and interactive system. It facilitates communication between various sensors and modules, provides real-time color detection and notifications, and offers a centralized dashboard for monitoring and control, making it suitable for a wide range of engineering applications.

3.7.1 MIT App Inventor

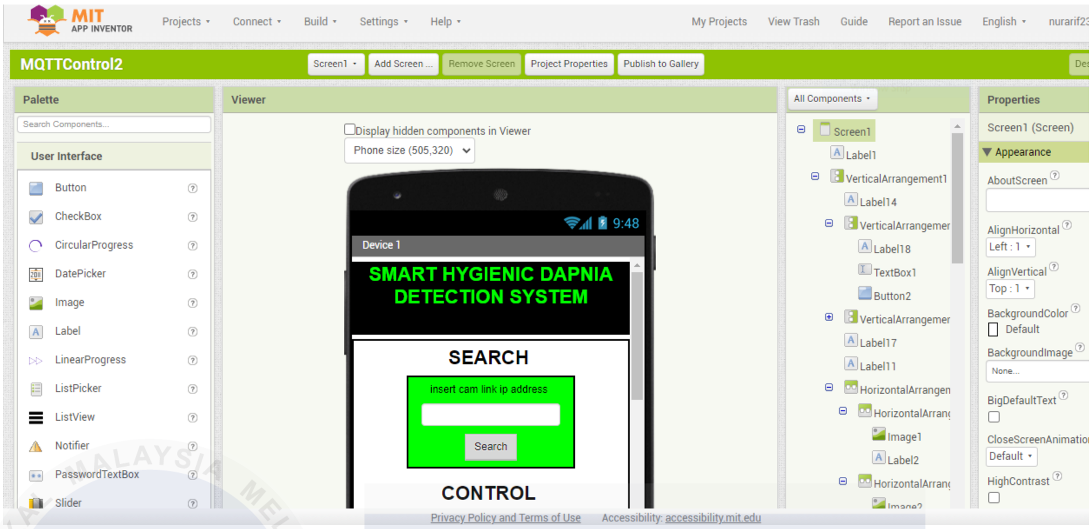


Figure 3.12: Designer interface on MIT app inventor

The MIT App Inventor user interface comprises two primary editors: the design editor and the blocks editor. The design editor, also referred to as the designer (see Fig. 3.1), facilitates a drag-and-drop interface for arranging the elements of the application's user interface (UI). This intuitive tool allows users to visually organize buttons, text fields, images, and other components to create the desired layout and appearance of their app. Complementing this, the blocks editor (see Fig. 3.2) provides a space where app developers can visually map out the logic of their applications using color-coded blocks. These blocks, which resemble puzzle pieces, represent various programming constructs and functions, enabling users to describe the program's behaviour without the need for traditional coding methods[25].

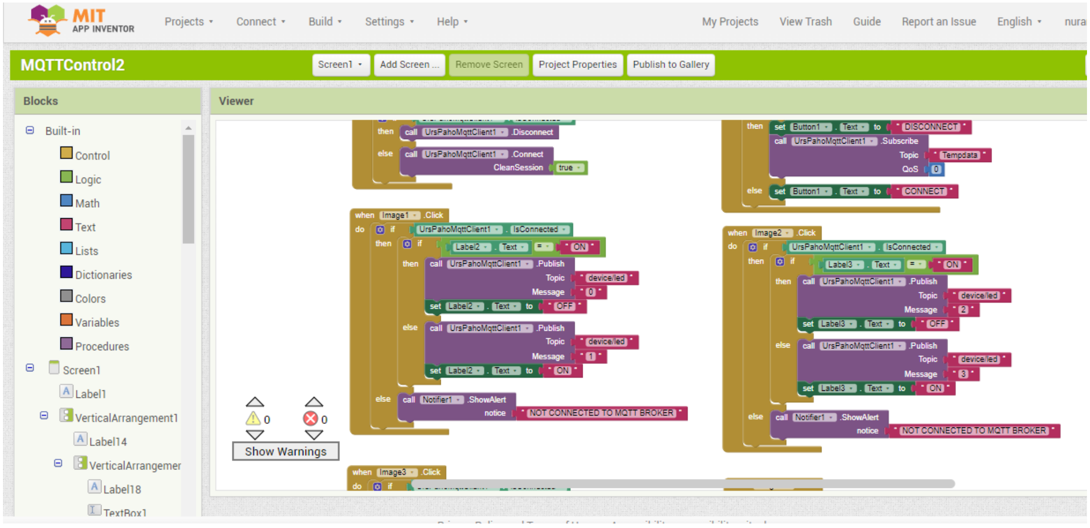


Figure 3.13: Blocks interface on MIT app inventor

To streamline development and testing processes, App Inventor offers a mobile app known as the App Inventor Companion, or simply "the Companion." This application empowers developers to test and refine their apps in real-time, providing instant feedback on changes made within the design and blocks editors. By seamlessly connecting the Companion app to the App Inventor interface, developers can observe how modifications affect app functionality directly on their mobile devices. This immediate feedback loop accelerates the development cycle, allowing for rapid iteration and adjustment of app features. With its user-friendly design editor, visual blocks editor, and real-time testing capabilities, MIT App Inventor democratizes mobile app development, enabling individuals with varying levels of programming experience to create and refine mobile applications efficiently[25].

3.7.2 MQTT broker

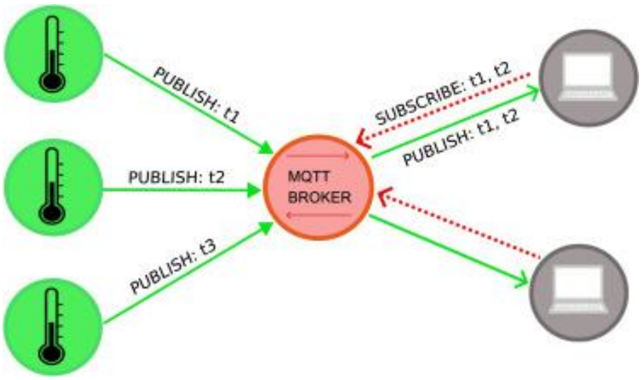


Figure 3.14: MQTT scenario[26]

Figure above represents a simple MQTT scenario in which three temperature sensors publish their values, each with a specific topic (“t1,” “t2,” and “t3”). Clients interested in the sensor temperatures subscribe to these topics; for example, the top-right client in Figure 2 subscribes to topics t1 and t2. Once the messages become available, the broker sends them to the relevant subscribers. The next section presents the design of PrioMQTT, a modified version of MQTT that supports prioritized low-latency communications at the application layer while maintaining compatibility with the standard MQTT protocol. This enhancement aims to improve the efficiency and responsiveness of MQTT in scenarios where message priority and latency are critical factors[26].

3.7.3 Arduino IDE

The Arduino Integrated Development Environment (IDE), also known as Arduino Software (IDE), is a comprehensive platform designed to facilitate the development of Arduino-based projects. It comprises various elements, including a text editor for writing and editing code, a message area to display important notifications and feedback, and a text console for monitoring program output and debugging messages.

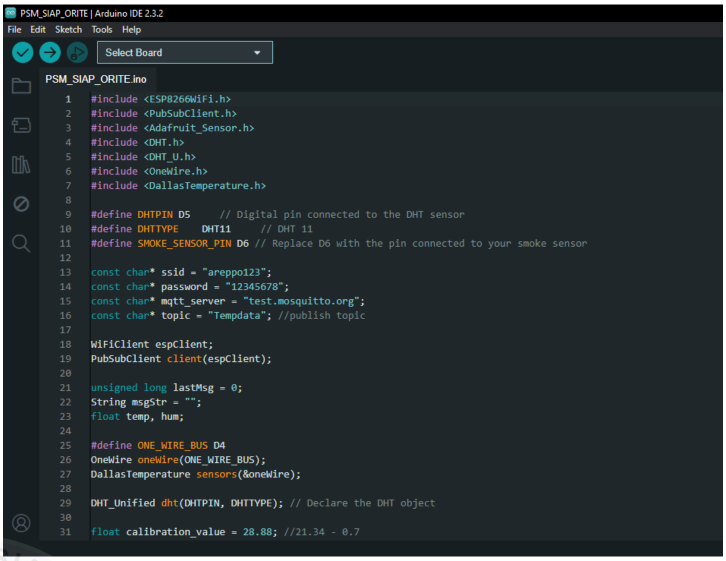


Figure 3.15: Arduino IDE interface

Additionally, the IDE features a toolbar with buttons that provide convenient access to standard functions such as compiling and uploading code, as well as a series of menus for accessing additional settings and functionalities. One of the key functions of the Arduino IDE is its ability to establish a connection with Arduino hardware, enabling users to upload programs to Arduino boards and communicate with them seamlessly[27]. This integration with Arduino hardware ensures a smooth and efficient development workflow, allowing users to write, test, and deploy code to their Arduino projects with ease.

3.7.4 Python

Python is a widely used programming language suitable for real-world applications. It is a high-level, dynamic, object-oriented, general-purpose language that operates through an interpreter. Python is designed to be easy to understand and use, making it popular among beginners and professionals alike. It has gained popularity as an introductory language, often replacing Java in educational settings.

As a dynamically typed language, Python offers flexibility and is more forgiving of errors, allowing programs to run until an issue is encountered. Python supports various programming styles, including structural and object-oriented, and can integrate modular components from other languages. For instance, you can write a module in C++ and import it into Python. This flexibility and simplicity make Python a versatile tool for engineers and developers.[28]

3.8 Working Flow of The Proposed System

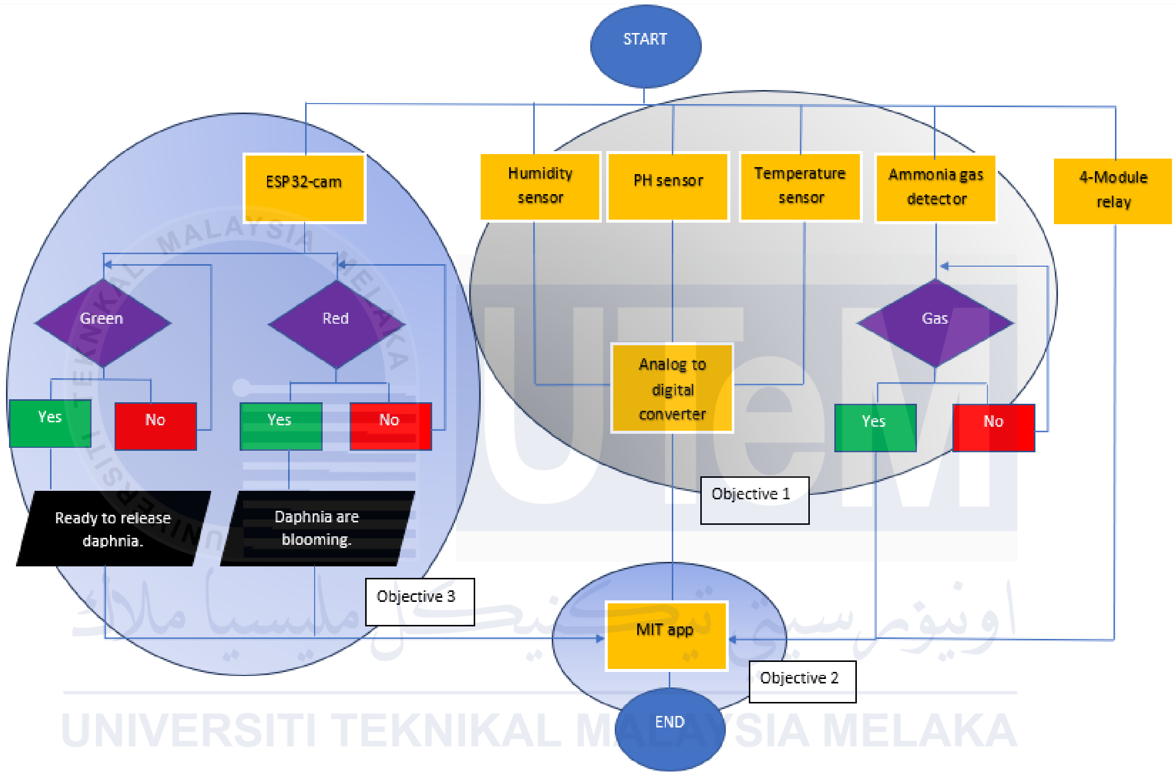


Figure 3.16: Flow chart for my proposed project

The Smart Hygiene Daphnia Magna Detection System comprises five crucial sensors for monitoring and controlling the aquatic environment. The first sensor is a temperature sensor, which determines the water's temperature and sends this data to the MIT App Inventor app for real-time monitoring. The second sensor is a pH sensor that measures the acidity or alkalinity of the water, also providing this information to the MIT App Inventor app to assess the general water quality. Additionally, the system includes an ammonia gas detection sensor to monitor the presence of ammonia in the water, a vital factor for Daphnia magna health. Data from this sensor is sent to the MIT App Inventor app, keeping users informed about potential dangers to the Daphnia magna population.

The fourth sensor is an ESP32 camera, capable of identifying red and green hues in the water. When the green color is detected, the system sends a warning to the Telegram app, indicating that conditions are ideal for releasing Daphnia magna. Conversely, when the red color is detected, it indicates that the Daphnia are flourishing. This real-time alert system provides an effective means of monitoring and responding to environmental conditions to maintain the well-being and optimal growth of Daphnia magna.

The fifth sensor is the DHT11 humidity sensor, which can detect humidity levels and surrounding temperature. This sensor provides critical information about the environmental conditions, ensuring that the overall habitat remains conducive for Daphnia magna. The data from the DHT11 sensor is also sent to the MIT App Inventor app for real-time monitoring.

Furthermore, the system incorporates a 4-relay module, allowing breeders to control four different outputs remotely through the MIT App Inventor app. This capability enhances the management and maintenance of the aquatic environment by enabling precise control over various aspects of the system.

Meanwhile, the device gathers information from sensors that measure water temperature, pH, ammonia, humidity, and color. The MIT App Inventor server functions as a central hub, receiving this data and updating the mobile application's dashboard for remote monitoring. This streamlined data flow ensures efficient real-time monitoring and control of the aquatic environment.

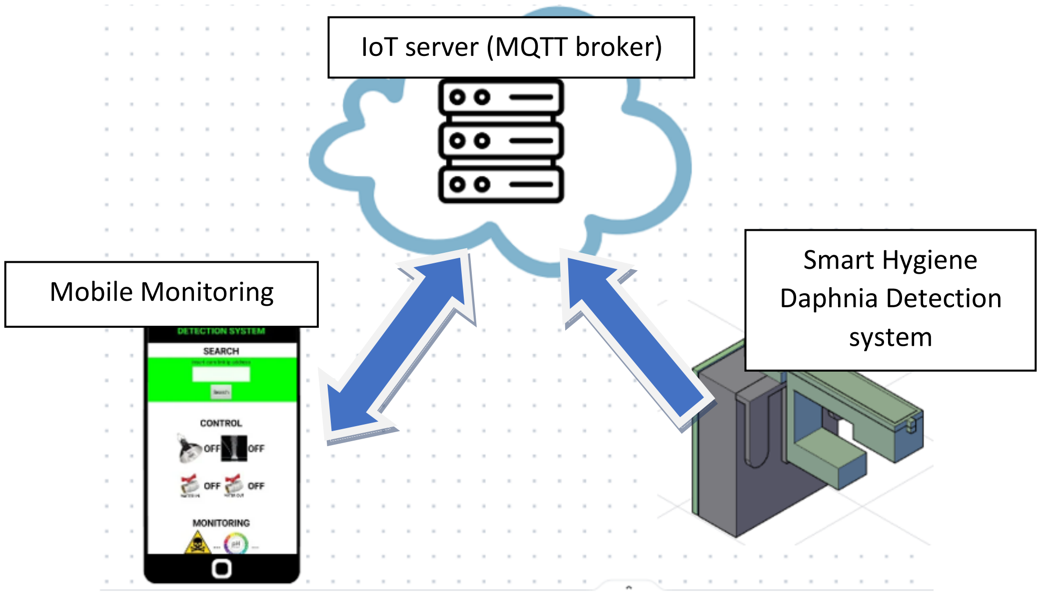


Figure 3.17: Shows how do proposed project work theoretically.

3.9 Device development step

First, all sensors are connected to the ESP8266 microcontroller as illustrated in figure 3.2. Once the connections are made, the appropriate code is uploaded to the ESP8266 using the Arduino IDE software. Following this, a separate piece of code is uploaded to the ESP32-CAM module to obtain its IP address, which will be incorporated into the Python script. This enables the ESP32-CAM to detect color changes effectively. The connection shown in figure 3.2 is specifically designed to provide a power supply to the ESP32-CAM, as it already contains an integrated microcontroller.

The system is designed to connect through MIT App Inventor via Wi-Fi, facilitating a comprehensive monitoring and control interface. The monitoring section, depicted in figure 4.2, displays real-time data from various sensors including humidity, water temperature, surrounding temperature, ammonia concentration, and pH levels. In addition, the control section enables the operation of four relay outputs that manage devices such as an air bubbler, heat lamp, solenoid valve, and water pump, as detailed in table 4. Lastly, the setup includes an online streaming feature from the ESP32-CAM, as shown in figure 4.1, providing continuous visual monitoring capabilities.

3.10 Summary

The operational flow of the Daphnia magna detection system for smart hygiene is described in the methodology chapter. The ESP8266 microcontroller is responsible for coordinating the functioning of all the sensors in the system. The ESP8266 controls the temperature, pH, gas, humidity, relay and esp32 camera data gathering, which starts the process. The MIT app inventor IoT server, which serves as a central hub for processing and receives this data after that. Remote monitoring of aquatic conditions is possible by the ESP32 microcontroller, which enables the smooth integration of real-time information onto the dashboard of the mobile application.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Result

4.1.1 Dashboard MIT app inventor

The result of the Daphnia magna detection system is displayed on the MIT App Inventor interface, which serves as the central dashboard for the system. This dashboard integrates all the necessary monitoring and control elements required for an effective and smart hygienic Daphnia detection system. It presents a comprehensive view of the environmental conditions, including temperature, pH, ammonia, and humidity levels. Additionally, it displays visual data from the ESP32 camera, providing real-time images and alerts.

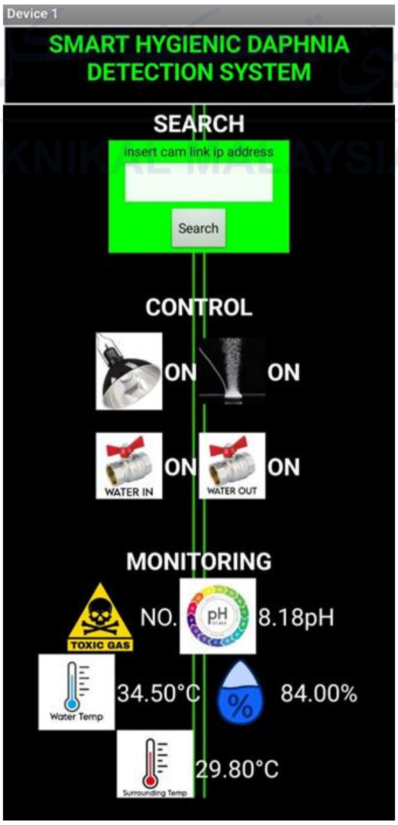


Figure 4.0: Dashboard MIT app inventor result

The dashboard is designed to be user-friendly, ensuring that all critical information is easily accessible and understandable. It allows users to monitor the conditions in real-time, with clear indicators and graphical representations of the data collected by the sensors. The dashboard also provides control functionalities, enabling users to adjust environmental parameters through the 4-relay module. This feature allows for precise management of devices such as heat lamp, air bubbler, solenoid valve and water pump to ensuring the aquatic environment remains optimal for the health and growth of Daphnia magna. MIT App Inventor dashboard offers a centralized, intuitive interface for monitoring and controlling the Daphnia magna detection system, making it an essential tool for maintaining a hygienic and effective aquatic environment.

4.1.2 Sensor monitoring system

For monitoring results, the sensor parameters initially appear on the Arduino IDE before being displayed on the MIT App Inventor dashboard. The dashboard is designed to update every 2 seconds if there are any changes in the parameters. This frequent update interval ensures that users always have access to the most current data, allowing for real-time monitoring of the aquatic environment.

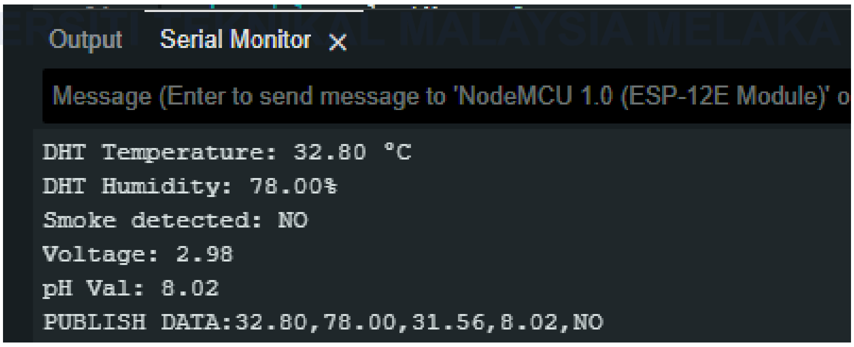


Figure 4.1: Arduino IDE serial monitor results.

The parameters monitored include temperature, pH, ammonia gas, and humidity levels, as well as visual data from the ESP32 camera. Each parameter is displayed in a clear and organized manner on the MIT App Inventor interface, making it easy for users to interpret the data quickly. Graphical representations and numerical values are used to provide a comprehensive view of the environmental conditions, helping users to identify any potential issues promptly.

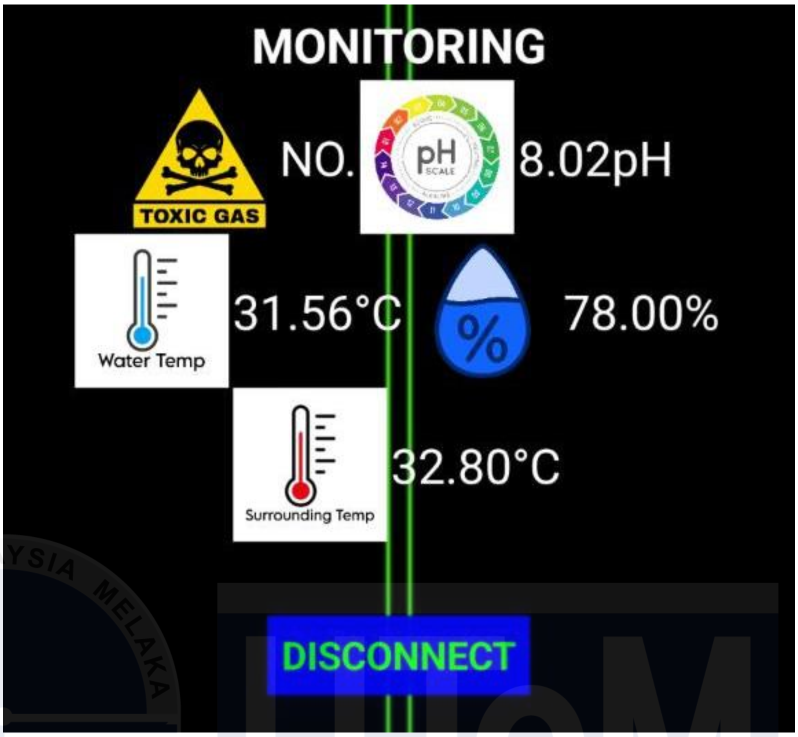


Figure 4.2: MIT app dashboard result same as serial monitor.

Because of these features, the MIT App Inventor dashboard enables comprehensive analysis of the PH and water temperature in 5 liters of water over a month, correlating it with temperature at 1 PM and pH levels. The real-time monitoring and frequent updates allow for accurate tracking of how these environmental factors impact the Daphnia population to optimize conditions for their growth and health.

4.1.3 Relay control system

For control results, the MIT App Inventor interface focuses on four types of controllable outputs a heat lamp, an air bubbler, a solenoid valve, and a water pump. The hardware setup for this system includes these devices connected to a 4-relay module, which is managed by the ESP8266 microcontroller. Within the app, users have access to on and off buttons for each device, allowing for precise control over the aquatic environment. The heat lamp regulates the water temperature, and the air bubbler ensures adequate oxygenation. The solenoid valve manages water flow into the system, while the water pump removes dirty water from the habitat.

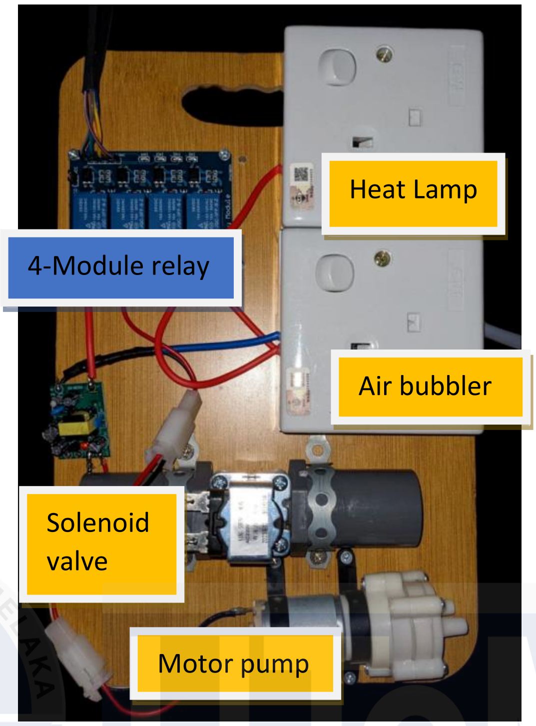
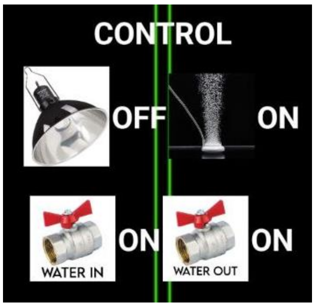
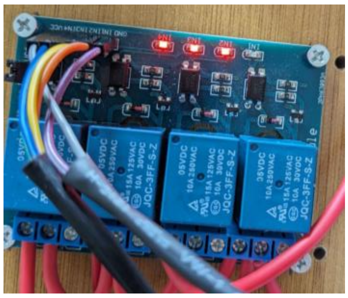
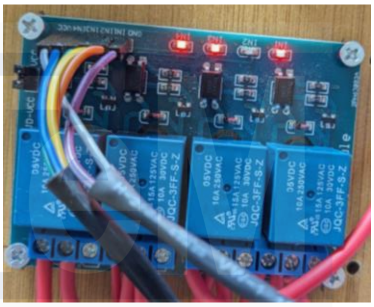
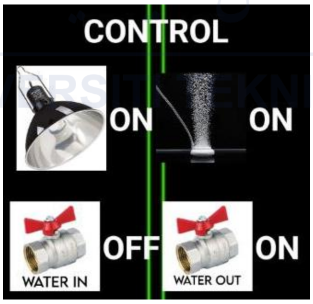
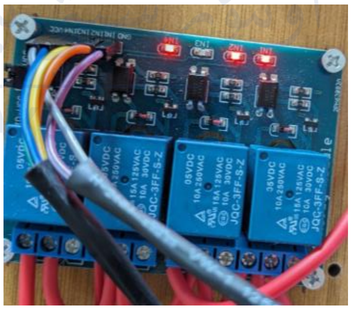
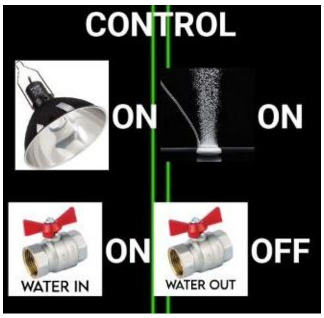
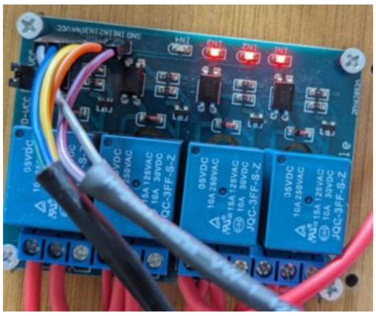


Figure 4.3: Relay output configuration

When a user activates a device through the app, the corresponding relay is triggered, indicated by a red LED light on the relay board, signaling that the device is on. This visual confirmation ensures that users can easily verify the operational status of each component directly from the relay module. The integration of these control features into the MIT App Inventor interface provides a comprehensive and user-friendly way to manage and adjust the conditions within the Daphnia magna habitat. This setup ensures that optimal environmental conditions are maintained, promoting the health and growth of the Daphnia population. The ability to remotely control these devices adds a significant level of convenience and efficiency, making the system highly effective for managing aquatic environments.

Table 4.0: MIT control dashboard vs relay condition

<table>
<tr><th>MIT app interface</th><th>Relay condition</th></tr>
<tr><td>
</td><td></td></tr>
<tr><td>
</td><td></td></tr>
<tr><td>
</td><td></td></tr>
<tr><td>
</td><td></td></tr>
</table>

4.1.4 ESP32 Camera Module Result

For the ESP32 camera results, users need to enter the camera's IP address before starting the video stream. The streaming output appears as two combined videos showing the same footage but with different detection overlays. The first video includes a blue box to detect green Chlorella water, indicating favorable conditions for Daphnia magna. The second video highlights a red box to detect red light, signifying blooming Daphnia. This dual-video setup with color-coded detection boxes provides a clear visual representation of the water conditions and Daphnia health, enhancing the overall monitoring capabilities of the system.

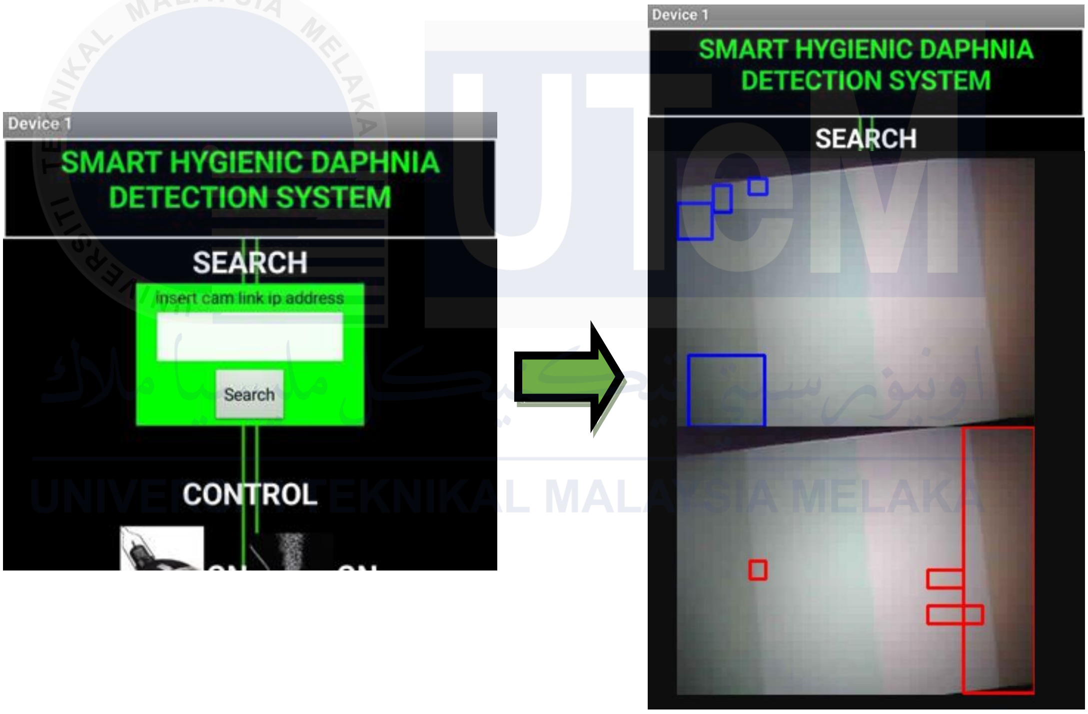


Figure 4.4: MIT app camera search result.

When the camera detects these specific conditions, it triggers the ESP8266 microcontroller to send a message to Telegram, informing the breeder about the status. This integration ensures that the breeder is promptly notified of significant changes in the aquatic environment, enabling timely interventions to maintain optimal conditions for the Daphnia magna population. The system's ability to provide real-time updates and alerts enhances the overall management and maintenance of the habitat, promoting the health and growth of the Daphnia magna.

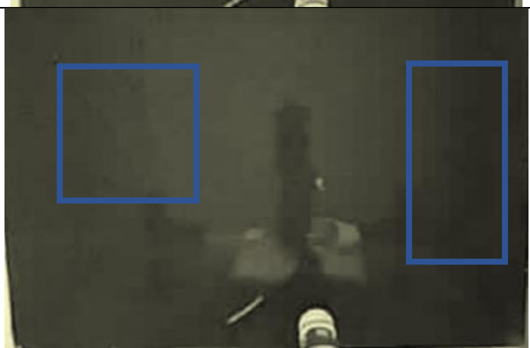
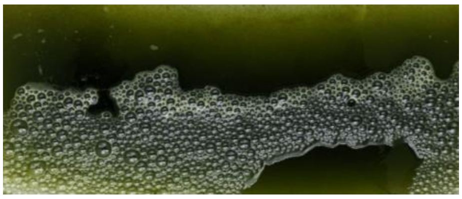
4.2 Analysis

4.2.1 ESP32 camera colour detecting analysis.

An analysis yields of cultured Chlorella green water over a seven-day period, comparing results obtained from an ESP32-CAM with colour detection to those observed through direct visual inspection. The study involves streaming data from the ESP32-CAM, which is Python language programmed to detect the presence of green coloration indicative of Chlorella growth. For the initial four days, the camera does not detect any significant green colour. However, from day five onward, the ESP32-CAM starts to identify the green hue associated with Chlorella. Upon detection, the system is designed to send a notification via Telegram. This approach allows for continuous, automated monitoring of the Chlorella culture, providing real-time updates and reducing the reliance on manual observation. The use of the ESP32-CAM for colour detection enhances the accuracy and efficiency of the monitoring process, offering a reliable alternative to subjective visual assessment.

Table 4.1: ESP32 cam view vs real life view

Days	Esp32 Camera View	Real life view
1		
2		

3		
4		
5		
6		
7		

4.2.2 Cultured chlorella green water analysis

The second analysis involves measuring the Daphnia population over a 30-day period, using a highly sensitive mini scale to determine their weight. The method consists of two main steps. First, the rake, which is used to collect the Daphnia, is weighed on the mini scale. Next, the rake is weighed again, but this time with the Daphnia on it. By subtracting the weight of the empty rake from the combined weight of the rake and Daphnia, the net weight of the Daphnia is obtained. This process ensures precise measurement of the Daphnia population, allowing for accurate tracking of their growth and density over the month-long study. The use of a sensitive mini scale is crucial for detecting even slight variations in weight, thus providing reliable data for the analysis.

Figure 4.5: Weighing daphnia magna using sensitive mini scale

The method confirms that the cultured hygienic Chlorella green water provides a safe environment for breeding Daphnia magna. Each week, excess Daphnia are removed to prevent overcrowding, ensuring there is sufficient space for reproduction and reducing the risk of mortality due to overcrowding. This approach maintains an optimal population density, promoting healthy growth and sustainability of the Daphnia magna population in the cultured environment. Regular monitoring and adjustment help create a balanced ecosystem, supporting continuous and effective breeding of Daphnia magna.

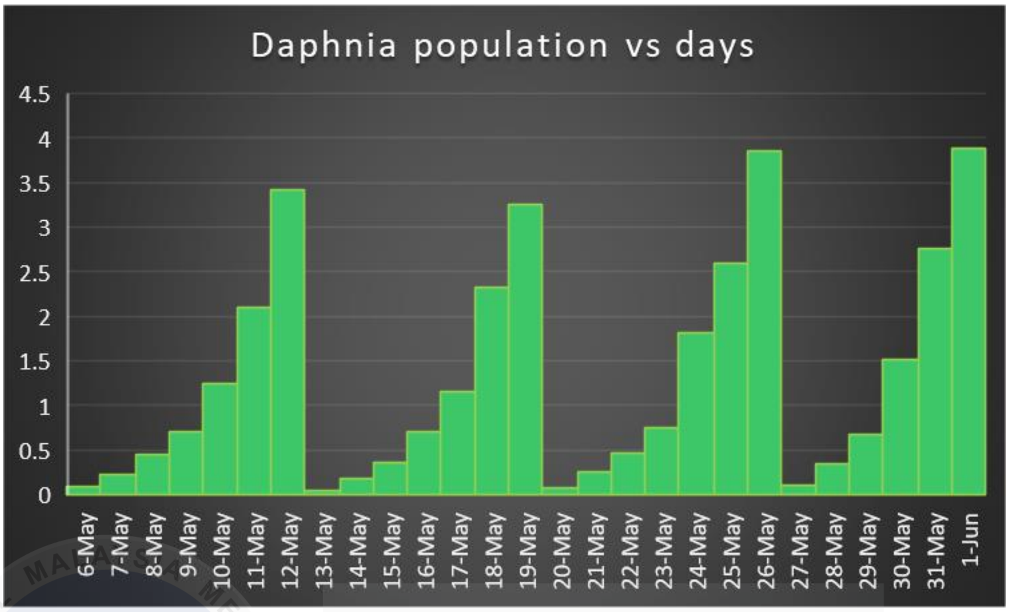


Figure 4.6: Daphnia magna population vs 30 days graph

4.2.3 Water & environment temperature sensor analysis

The analysis involved collecting hourly water temperature data using a combined sensor system consisting of a DS18B20 temperature sensor and a mounted temperature sensor in DHT11 humidity sensor. Water temperature sensor will be dipped into the chlorella solution while DHT11 sensor will be hanging near the device. This study aimed to investigate the durability and endurance of Daphnia under fluctuating temperature conditions. The results indicated a significant negative impact on the Daphnia population when exposed to direct sunlight.

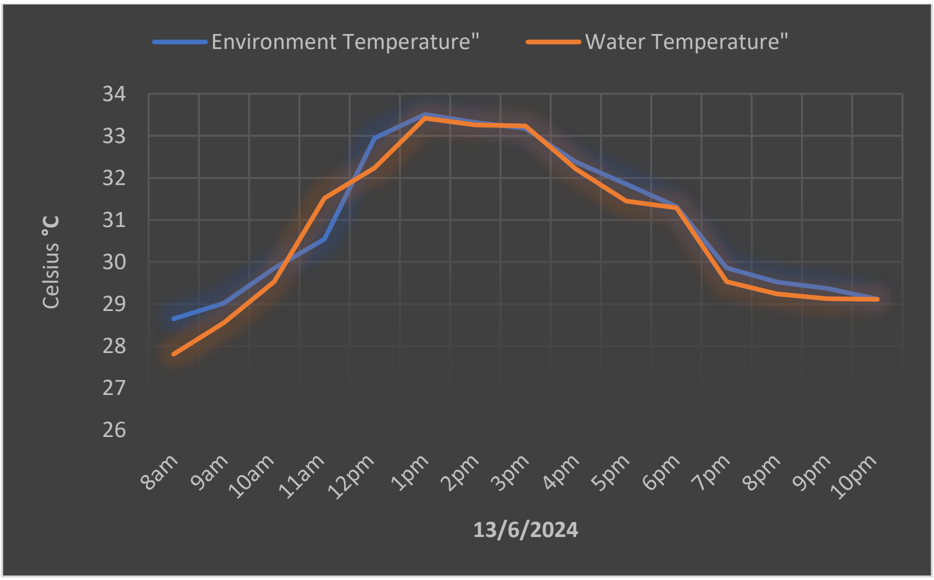


Figure 4.7: Environment and water temperature difference

Specifically, nearly 90% of the Daphnia died when placed in an aquarium subjected to direct sunlight. This high mortality rate underscores the importance of maintaining a stable and controlled temperature environment to ensure the survival and health of the Daphnia population. The data gathered from the sensors provide valuable insights into the critical temperature thresholds and environmental conditions necessary for sustaining Daphnia in cultured settings.

4.2.4 PH sensor analysis

The analysis of pH levels was conducted over a month, using a pH sensor to measure the potential hydrogen levels in the cultured Chlorella green water. It was observed that the matured Chlorella green water exhibited a higher pH value compared to distilled water, likely due to chemical reactions within the culture. The experiment results showed a gradual decline in pH levels over time, indicating a trend.

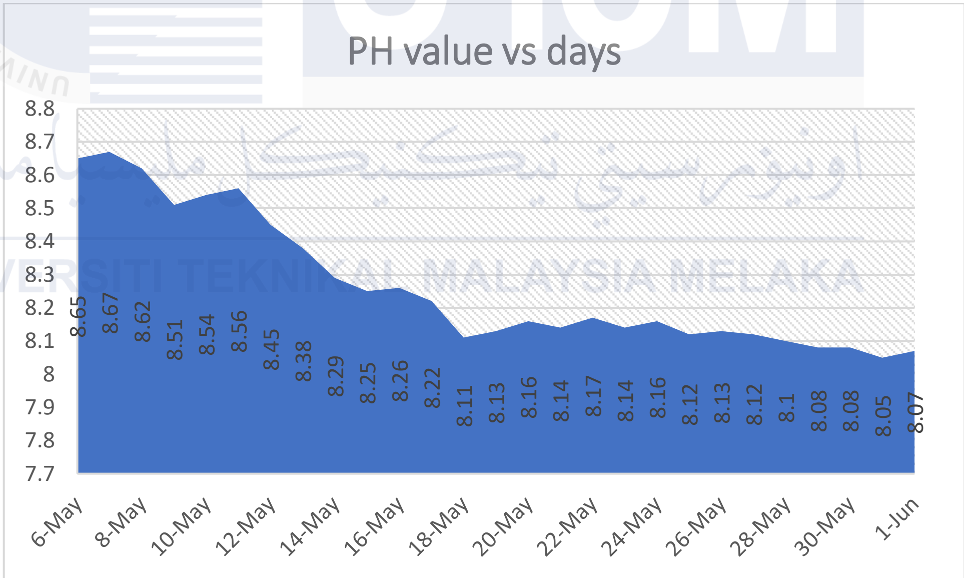


Figure 4.8: Solution PH value vs 30 days

This trend suggests that Daphnia magna acts as an effective water filter. By the end of the experiment, the water in the tank had become noticeably clearer, demonstrating the Daphnia's role in improving water quality. The data gathered from the pH sensor highlight the relationship between Daphnia activity and water filtration, providing evidence of their potential benefits in maintaining a balanced and healthy aquatic environment.

Figure 4.9: Filtered chlorella green water after daphnia occupied for 30 days.

4.3 AutoCAD 3d casing Drawing.

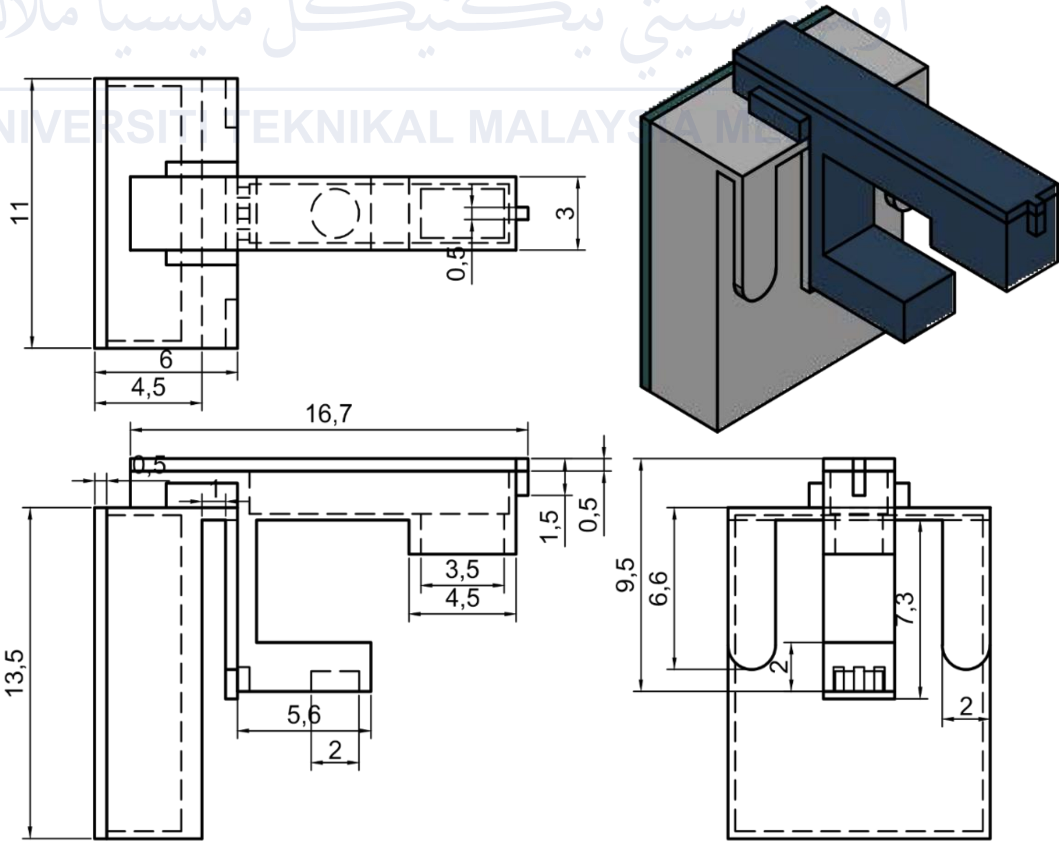


Figure 4.10: proposed project casing with dimensions illustrated by AutoCAD.

4.4 Final 3d printed casing result.

Figure 4.11: Latest product result.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, the Smart Hygiene Daphnia Detection System project for aquaculture effectively combines tools like a pH monitor, gas detector, humidity sensor, and temperature sensor to create the best environment for Daphnia growth. This semester, the focus is on setting up a control system with a 4-relay module to manage water flow, lighting, and oxygen bubbling in the breeding tanks. The project is making progress toward creating a system that helps breeders keep the tanks clean. Recommendations include setting up a data logging system to track sensor readings over time, creating a mobile app or website for remote monitoring and control, adding automated alerts like speaker notifications to inform breeders of any issues, improving the control algorithms for better management, adding backup sensors and power supplies for reliability, providing user training and documentation, adding more sensors such as ammonia and nitrate detectors for better water quality monitoring, and regularly updating the system based on user feedback and new technology to keep it effective and user-friendly.

Future work for the Smart Hygiene Daphnia Detection System project involves several key areas to enhance its functionality and reliability. Firstly, there are plans to expand the sensor array by incorporating sensors for ammonia, nitrate, and other water quality parameters to provide a more comprehensive overview of the tank environment. Implementing automated calibration and maintenance routines for the sensors will improve accuracy and reduce manual intervention. Enhancing the control system by refining the relay module algorithms will ensure more precise management of water flow, lighting, and oxygen levels. Another focus area is the development of robust backup systems, including redundant sensors and power supplies, to ensure continuous operation and reliability. Providing thorough user training and detailed documentation will be crucial for effective system operation and maintenance. Regularly updating the system based on user feedback and technological advancements will ensure it remains effective and user-friendly.

REFERENCES

[1] A. Ali, S. Jawad, A. H. Ali, and M. Winter, “Stability Analysis for the Phytoplankton-Zooplankton Model with Depletion of Dissolved Oxygen and Strong Allee Effects,” *Results in Engineering*, p. 102190, May 2024, doi: 10.1016/J.RINENG.2024.102190.

[2] N. N. Suh and R. A. Nyiawung, “Climate change dynamics and youth participation decisions in aquatic food systems: Case of the oyster sector in The Gambia, West Africa,” *Mar Policy*, vol. 156, p. 105804, Oct. 2023, doi: 10.1016/J.MARPOL.2023.105804.

[3] “Water on the Web | Understanding | Lake Ecology | The Food Web.” Accessed: Dec. 25, 2023. [Online]. Available: https://www.waterontheweb.org/under/lakeecology/11_foodweb.html

[4] “8. PERANGKAAN 2022 Infografik Perikanan”.

[5] “Fish & Seafood - Malaysia | Statista Market Forecast.” Accessed: Jan. 16, 2024. [Online]. Available: https://www.statista.com/outlook/cmo/food/fish-seafood/malaysia#demographics

[6] “9. LATEST_FINALIZE_MAKLUMAT PERIKANAN_2022”.

[7] D. Ebert, “Daphnia as a versatile model system in ecology and evolution,” *EvoDevo*, vol. 13, no. 1. BioMed Central Ltd, Dec. 01, 2022. doi: 10.1186/s13227-022-00199-0.

[8] D. Ebert, “Daphnia as a versatile model system in ecology and evolution,” *EvoDevo*, vol. 13, no. 1. BioMed Central Ltd, Dec. 01, 2022. doi: 10.1186/s13227-022-00199-0.

[9] H. Herman, Y. Andriani, A. Sahidin, T. Hidayat, and T. Herawati, “Culture of Daphnia sp. (crustacean - Cladocera): The effect of manure variation on the growth, natality, and mortality,” in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics Publishing, Apr. 2018. doi: 10.1088/1755-1315/137/1/012018.

[10] C. Martínez, A. T. Souza, and J. Kubecka, “Fish eradication in freshwater ecosystems by repeated fishing,” *IFAC-PapersOnLine*, vol. 55, no. 20, pp. 659–664, Jan. 2022, doi: 10.1016/J.IFACOL.2022.09.171.

[11] E. Borgia, “The Internet of Things vision: Key features, applications and open issues,” *Comput Commun*, vol. 54, pp. 1–31, Dec. 2014, doi: 10.1016/J.COMCOM.2014.09.008.

[12] N. Sadek, N. kamal, and D. Shehata, “Internet of Things based smart automated indoor hydroponics and aeroponics greenhouse in Egypt,” *Ain Shams Engineering Journal*, vol. 15, no. 2, p. 102341, Feb. 2024, doi: 10.1016/J.ASEJ.2023.102341.

[13] T. M. Edwards, H. J. Puglis, D. B. Kent, J. L. Durán, L. M. Bradshaw, and A. M. Farag, “Ammonia and aquatic ecosystems – A review of global sources, biogeochemical cycling, and effects on fish,” *Science of The Total Environment*, vol. 907, p. 167911, Jan. 2024, doi: 10.1016/J.SCITOTENV.2023.167911.

[14] P. Kanakaraja, P. Syam Sundar, N. Vaishnavi, S. Gopal Krishna Reddy, and G. Sai Manikanta, “IoT enabled advanced forest fire detecting and monitoring on Ubidots platform,” *Mater Today Proc*, vol. 46, pp. 3907–3914, Jan. 2021, doi: 10.1016/J.MATPR.2021.02.343.

[15] I. Ezzahoui, R. A. Abdelouahid, K. Taji, and A. Marzak, “Hydroponic and Aquaponic Farming: Comparative Study Based on Internet of things IoT technologies.,” *Procedia Comput Sci*, vol. 191, pp. 499–504, Jan. 2021, doi: 10.1016/J.PROCS.2021.07.064.

[16] K. K. Y. Shin, T. P. Ping, M. G. B. Ling, C. Chee Jiun, and N. A. B. Bolhassan, “SMART GROW – Low-cost automated hydroponic system for urban farming,” *HardwareX*, vol. 17, p. e00498, Mar. 2024, doi: 10.1016/J.OHX.2023.E00498.

[17] J. Doshi, T. Patel, and S. K. Bharti, “Smart Farming using IoT, a solution for optimally monitoring farming conditions,” *Procedia Comput Sci*, vol. 160, pp. 746–751, Jan. 2019, doi: 10.1016/J.PROCS.2019.11.016.

[18] A. Abu Sneineh and A. A. A. Shabaneh, “Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things,” *MethodsX*, vol. 11, p. 102401, Dec. 2023, doi: 10.1016/J.MEX.2023.102401.

[19] Andreas, C. R. Aldawira, H. W. Putra, N. Hanafiah, S. Surjarwo, and A. Wibisurya, “Door Security System for Home Monitoring Based on ESP32,” *Procedia Comput Sci*, vol. 157, pp. 673–682, Jan. 2019, doi: 10.1016/J.PROCS.2019.08.218.

[20] M. I. Khan, K. Mukherjee, R. Shoukat, and H. Dong, “A review on pH sensitive materials for sensors and detection methods,” *Microsystem Technologies*, vol. 23, no. 10. Springer Verlag, pp. 4391–4404, Oct. 01, 2017. doi: 10.1007/s00542-017-3495-5.

[21] M. Fezari and A. Al Dahoud, “Exploring One-wire Temperature sensor ‘DS18B20’ with Microcontrollers.” [Online]. Available: https://www.researchgate.net/publication/330854061

[22] H. Dietz *et al.*, “ESP32-CAM as a programmable camera research platform,” in *IS and T International Symposium on Electronic Imaging Science and Technology*, Society for Imaging Science and Technology, 2022. doi: 10.2352/EI.2022.34.7.ISS-232.

[23] M. R. Islam, K. Oliullah, M. M. Kabir, M. Alom, and M. F. Mridha, “Machine learning enabled IoT system for soil nutrients monitoring and crop recommendation,” *J Agric Food Res*, vol. 14, p. 100880, Dec. 2023, doi: 10.1016/J.JAFR.2023.100880.

[24] S. Shafiril, A. Yusoff, and N. C. Yusoff, “The Development of an Automated System in Detecting Environmental Data for the Monitoring of Forest Activity,” *International Journal of Environmental Science and Development*, vol. 7, no. 7, pp. 532–536, 2016, doi: 10.18178/ijesd.2016.7.7.834.

[25] E. W. Patton, M. Tissenbaum, and F. Harunani, “MIT App Inventor: Objectives, Design, and Development,” in *Computational Thinking Education*, Springer Singapore, 2019, pp. 31–49. doi: 10.1007/978-981-13-6528-7_3.

[26] G. Patti, L. Leonardi, G. Testa, and L. Lo Bello, “PrioMQTT: A prioritized version of the MQTT protocol,” *Comput Commun*, vol. 220, pp. 43–51, Apr. 2024, doi: 10.1016/J.COMCOM.2024.03.018.

[27] P. N. Chandramohan *et al.*, “Development of Traffic Light Control System Using Arduino Uno Full Paper,” 2022. [Online]. Available: https://beam.pmu.edu.my

[28] K. R. Srinath, “Python-The Fastest Growing Programming Language,” *International Research Journal of Engineering and Technology*, 2017, [Online]. Available: www.irjet.net

APPENDICES

Arduino IDE

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define DHTPIN D5      // Digital pin connected to the DHT sensor
#define DHTTYPE    DHT11     // DHT 11
#define SMOKE_SENSOR_PIN D6 // Replace D6 with the pin connected to your smoke
sensor

const char* ssid = "areppo123";
const char* password = "12345678";
const char* mqtt_server = "test.mosquitto.org";
const char* topic = "Tempdata"; //publish topic

WiFiClient espClient;
PubSubClient client(espClient);

unsigned long lastMsg = 0;
String msgStr = "";
float temp, hum;

#define ONE_WIRE_BUS D4
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

DHT_Unified dht(DHTPIN, DHTTYPE); // Declare the DHT object

float calibration_value = 28.88; //21.34 - 0.7
int phval = 0;
unsigned long int avgval;
int buffer_arr[10], temp_val;
float ph_act;

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
```

```
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    digitalWrite(2, 0);
    delay(200);
    digitalWrite(2, 1);
    delay(200);
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();

  // Switch on the LED based on payload
  switch ((char)payload[0]) {
    case '0':
      digitalWrite(D0, HIGH);
      break;
    case '1':
      digitalWrite(D0, LOW);
      break;
    case '2':
      digitalWrite(D1, HIGH);
      break;
    case '3':
      digitalWrite(D1, LOW);
      break;
    case '4':
      digitalWrite(D2, HIGH);
      break;
    case '5':
      digitalWrite(D2, LOW);
      break;
    case '6':
      digitalWrite(D3, HIGH);
      break;
    case '7':
```

```
      digitalWrite(D3, LOW);
      break;
    default:
      break;
  }
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      client.subscribe("device/led");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);

  dht.begin();
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  dht.humidity().getSensor(&sensor);
  uint32_t delayMS = sensor.min_delay / 1000;

  pinMode(SMOKE_SENSOR_PIN, INPUT);

  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);

  sensors.begin(); // Start up Dallas Temperature sensor
}

void loop() {
  if (!client.connected()) {
```

```
    reconnect();
  }
  client.loop();

  unsigned long now = millis();
  if (now - lastMsg > 2000) {
    lastMsg = now;

    sensors.requestTemperatures(); // Request temperature from Dallas sensor
    float temperatureC = sensors.getTempCByIndex(0); // Get temperature in
Celsius from Dallas sensor

    sensors_event_t event;
    dht.temperature().getEvent(&event);
    if (!isnan(event.temperature)) {
      temp = event.temperature;
      Serial.print(F("DHT Temperature: "));
      Serial.print(temp);
      Serial.println(F(" °C"));
    } else {
      Serial.println(F("Error reading DHT temperature!"));
    }

    dht.humidity().getEvent(&event);
    if (!isnan(event.relative_humidity)) {
      hum = event.relative_humidity;
      Serial.print(F("DHT Humidity: "));
      Serial.print(hum);
      Serial.println(F("%"));
    } else {
      Serial.println(F("Error reading DHT humidity!"));
    }

    int smokeValue = digitalRead(SMOKE_SENSOR_PIN);

    if (smokeValue == HIGH) {
      Serial.println("Smoke detected: YES");
    } else {
      Serial.println("Smoke detected: NO");
    }

    for(int i = 0; i < 10; i++) {
      buffer_arr[i] = analogRead(A0); // Using analog pin A0 for pH sensing
      delay(30);
    }

    for(int i = 0; i < 9; i++) {
      for(int j = i + 1; j < 10; j++) {
```

```
        if(buffer_arr[i] > buffer_arr[j]) {
          temp_val = buffer_arr[i];
          buffer_arr[i] = buffer_arr[j];
          buffer_arr[j] = temp_val;
        }
      }
    }

    avgval = 0;
    for(int i = 2; i < 8; i++)
      avgval += buffer_arr[i];

    float volt = (float)avgval * 3.3 / 1023.0 / 6; // Adjusted ADC range to 0-
1023
    Serial.print("Voltage: ");
    Serial.println(volt);

    ph_act = -7 * volt + calibration_value;

    Serial.print("pH Val: ");
    Serial.println(ph_act);

    msgStr = String(temp) + "," + String(hum) + "," + String(temperatureC) +
"," + String(ph_act) + "," + (smokeValue == HIGH ? "YES" : "NO"); //
Concatenate all sensor readings
    byte arrSize = msgStr.length() + 1;
    char msg[arrSize];
    Serial.print("PUBLISH DATA:");
    Serial.println(msgStr);
    msgStr.toCharArray(msg, arrSize);
    client.publish(topic, msg);
    msgStr = "";
    delay(50);
  }
}
```

Python

```
from http.server import BaseHTTPRequestHandler, HTTPServer
import socketserver
import socket
from PIL import Image
import io
import cv2
import numpy as np
import urllib.request
import threading
import time


# WiFi Configuration
WIFI_SSID = "areppo123"
WIFI_PASS = "12345678"


# ESP32-CAM Server Configuration
ESP32_CAM_IP = "192.168.173.88"
ESP32_CAM_PORT = 80


# Global variables for color tracking
lower_color_1 = np.array([47, 24, 66])  # Lower bound for HSV color thresholding for color
1
```

```
upper_color_1 = np.array([137, 193, 175])  # Upper bound for HSV color thresholding for color 1
tracking_color_1 = (255, 0, 0)  # Color for drawing the tracking rectangle for color 1


lower_color_2 = np.array([0, 0, 50])  # Lower bound for HSV color thresholding for color 2
upper_color_2 = np.array([42, 255, 255])  # Upper bound for HSV color thresholding for color 2
tracking_color_2 = (0, 0, 255)  # Color for drawing the tracking rectangle for color 2


class ESP32CamHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/result_video':
            # Send result video stream
            self.send_response(200)
            self.send_header('Content-type', 'multipart/x-mixed-replace; boundary=frame')
            self.end_headers()


            try:
                while True:
                    start_time = time.time()  # Record start time for frame processing


                    # Fetch original raw frame
                    raw_img_resp = urllib.request.urlopen("http://" + ESP32_CAM_IP + "/cam-lo.jpg", timeout=5)
                    raw_img_array = np.array(bytearray(raw_img_resp.read()), dtype=np.uint8)
```

```
raw_frame = cv2.imdecode(raw_img_array, -1)

# Perform color tracking for color 1
tracked_frame_1 = track_color(raw_frame.copy(), lower_color_1,
upper_color_1, tracking_color_1)

# Perform color tracking for color 2
tracked_frame_2 = track_color(raw_frame.copy(), lower_color_2,
upper_color_2, tracking_color_2)

# Combine frames vertically
combined_frame = np.concatenate((tracked_frame_1, tracked_frame_2), axis=0)

_, img_encoded = cv2.imencode('.jpg', combined_frame)
img_bytes = img_encoded.tobytes()

self.wfile.write(b'--frame\r\n')
self.send_header('Content-type', 'image/jpeg')
self.send_header('Content-length', len(img_bytes))
self.end_headers()
self.wfile.write(img_bytes)
self.wfile.write(b'\r\n')

# Calculate time taken for processing and add a delay to maintain frame rate
processing_time = time.time() - start_time
```

```
                if processing_time < 0.1:  # Adjust this value as needed
                    time.sleep(0.1 - processing_time)

        except Exception as e:
            print("Error:", e)

    def log_message(self, format, *args):
        return

def track_color(frame, lower_color, upper_color, tracking_color):
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    mask = cv2.inRange(hsv_frame, lower_color, upper_color)
    contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    for contour in contours:
        area = cv2.contourArea(contour)
        if area > 100:
            x, y, w, h = cv2.boundingRect(contour)
            cv2.rectangle(frame, (x, y), (x + w, y + h), tracking_color, 2)
    return frame

def start_server():
    server_address = ('', 8080)  # Listen on all available interfaces
    httpd = HTTPServer(server_address, ESP32CamHandler)
```

```
    print('Server running...')
    httpd.serve_forever()


if __name__ == "__main__":
    # Start server in a separate thread
    server_thread = threading.Thread(target=start_server)
    server_thread.start()


    # Connect to ESP32-CAM and set up color tracking
    print("Server IP address:", socket.gethostbyname(socket.gethostname()))
    print("Connecting to WiFi...")
    # Insert WiFi connection code here


    print("Connected to WiFi")
    print("Performing color tracking...")


    # You can adjust this part to configure the ESP32-CAM resolution and other settings
    # Example:
    # urllib.request.urlopen("http://" + ESP32_CAM_IP + "/control?var=framesize&val=3")


    print("Color tracking initialized")


    # Keep the main thread running
    while True:
```

```
    pass


#include <WebServer.h>

#include <WiFi.h>
#include <esp32cam.h>
#include <ArduinoJson.h> // Library for handling JSON data
#include <HTTPClient.h>  // Library for making HTTP requests

const char* WIFI_SSID = "areppo123";
const char* WIFI_PASS = "12345678";
const char* BOT_TOKEN = "6520762393:AAGmemzVSL7g-zrGox0B-g79F0aCPDsnHwE";
const char* CHAT_ID = "499875827";
const char* BASE_URL = "https://api.telegram.org/bot";
const long INTERVAL = 1800000; // 30 minutes in milliseconds

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);

bool notificationSent = false;

void serveJpg()
{
  auto frame = esp32cam::capture();
  if (frame == nullptr) {
    Serial.println("CAPTURE FAIL");
    server.send(503, "", "");
    return;
  }
  String result = "CAPTURE OK " + String(frame->getWidth()) + "x" +
String(frame->getHeight()) + " " + String(static_cast<int>(frame->size())) +
"b";
  Serial.println(result);

  // Color detection thresholds in HSV
  int greenHueLow = 40;
  int greenHueHigh = 70;
  int redHueLow = 0;
  int redHueHigh = 10;

  bool greenDetected = false;
  bool redDetected = false;

  for (size_t i = 0; i < frame->size(); i += 3) {
    int hue = frame->data()[i] * 2; // Scale to 0-360 range
```

```
    int sat = frame->data()[i + 1];
    int val = frame->data()[i + 2];

    if (hue >= greenHueLow && hue <= greenHueHigh && sat >= 40 && val >= 40) {
      greenDetected = true;
    }
    else if (hue >= redHueLow && hue <= redHueHigh && sat >= 100 && val >=
100) {
      redDetected = true;
    }
  }

  if (!notificationSent && (greenDetected || redDetected)) {
    String message = "";
    if (greenDetected) {
      message += "Your Green Water is Ready\n";
    }
    if (redDetected) {
      message += "Your Daphnia is Start Blooming\n";
    }
    sendTelegramMessage(message);
    notificationSent = true;
  }

  server.setContentLength(frame->size());
  server.send(200, "image/jpeg");
  WiFiClient client = server.client();
  frame->writeTo(client);
}

void handleJpgLo()
{
  if (!esp32cam::Camera.changeResolution(loRes)) {
    Serial.println("SET-LO-RES FAIL");
  }
  serveJpg();
}

void handleJpgHi()
{
  if (!esp32cam::Camera.changeResolution(hiRes)) {
    Serial.println("SET-HI-RES FAIL");
  }
  serveJpg();
}

void handleJpgMid()
{
```

```
  if (!esp32cam::Camera.changeResolution(midRes)) {
    Serial.println("SET-MID-RES FAIL");
  }
  serveJpg();
}

void sendTelegramMessage(String message) {
  HTTPClient http;
  String url = String(BASE_URL) + String(BOT_TOKEN) + "/sendMessage";
  String data = "chat_id=" + String(CHAT_ID) + "&text=" + message;

  http.begin(url);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");
  int httpResponseCode = http.POST(data);

  if(httpResponseCode>0){
    String response = http.getString();
    Serial.println(httpResponseCode);
    Serial.println(response);
  }
  else {
    Serial.print("Error in sending message. HTTP Error: ");
    Serial.println(httpResponseCode);
  }

  http.end();
}

void setup(){
  Serial.begin(115200);
  Serial.println();
  {
    using namespace esp32cam;
    Config cfg;
    cfg.setPins(pins::AiThinker);
    cfg.setResolution(hiRes);
    cfg.setBufferCount(2);
    cfg.setJpeg(80);

    bool ok = Camera.begin(cfg);
    Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");
  }
  WiFi.persistent(false);
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
```

```
  Serial.print("http://");
  Serial.println(WiFi.localIP());
  Serial.println("  /cam-lo.jpg");
  Serial.println("  /cam-hi.jpg");
  Serial.println("  /cam-mid.jpg");

  server.on("/cam-lo.jpg", handleJpgLo);
  server.on("/cam-hi.jpg", handleJpgHi);
  server.on("/cam-mid.jpg", handleJpgMid);

  server.begin();
}

void loop()
{
  server.handleClient();
  static unsigned long lastNotificationTime = 0;
  if (!notificationSent && (millis() - lastNotificationTime >= INTERVAL)) {
    lastNotificationTime = millis();
    String message = "Your Green Water is Ready\nYour Daphnia is Start 
Blooming";
    sendTelegramMessage(message); // Default message
    notificationSent = true;
  }
}
```