AquaGuard: IoT-Based Smart Aquaponics Monitoring System

A PROJECT REPORT

submitted by

K.S.L.YUKTHA (230701154)
KARTHIKHA SRE M (230701143)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report titled "**AQUAGUARD: IOT-BASED SMART AQUAPONICS MONITORING SYSTEM**" is the bonafide work of "**K.S.L.YUKTHA (230701154), KARTHIKHA SRE M (230701143)**" who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Ms. S. Ponmani M.E.,MBA,

SUPERVISOR

Assistant Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on ____________________

Internal Examiner **External Examiner**

ABSTRACT

Aquaponics is an integrated system that combines conventional aquaculture with hydroponics, creating a sustainable ecosystem where waste produced by aquatic animals supplies nutrients for plant growth, and plants, in turn, purify the water. However, maintaining this balance requires continuous monitoring of key parameters such as water level, water quality, and temperature. Traditional methods rely heavily on manual observation and intervention, which are not only time-consuming but also highly prone to human error.

To address these limitations, this project proposes **AquaGuard**, a smart IoT-based monitoring system designed to automate the tracking of essential parameters in an aquaponics environment. Using a suite of sensors, including a water level sensor, a soil moisture sensor repurposed for water conductivity detection, and a digital temperature sensor, AquaGuard ensures real-time monitoring. A Bluetooth module is integrated to send immediate alerts to a user's mobile device when any reading deviates from pre-set safe limits.

This real-time notification system allows users to respond quickly t unfavourable conditions, ensuring the optimal health of both fish and plants. Additionally, AquaGuard is built as a cost-effective and scalable solution suitable for educational institutions, small-scale farmers, and home users. By reducing manual efforts and introducing automation, AquaGuard aims to enhance the sustainability and productivity of aquaponics systems.

.

ACKNOWLEDGEMENT

First, we thank the almighty God for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan, B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project, and Vice-Chairman **Mr. Abhay Shankar Meganthan**, **B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college principal, **Dr.S.N.Murugesan M.E., PhD.,** and **Dr. P. Kumar M.E., Ph.D., Head of the Department of Computer Science and Engineering,** and our project guide **Ms. S. Ponmani M.E.,MBA,** for her encouragement and guiding us throughout the project. We would like to thank our parents, friends, all faculty members, and supporting staff for their direct and indirect involvement in the successful completion of the project for their encouragement and support.

CHAPTER 1

INTRODUCTION

Aquaponics presents a modern and sustainable approach to farming by integrating aquaculture and hydroponics into a single, symbiotic system. This innovative method not only conserves water but also reduces the need for chemical fertilizers, making it an environmentally friendly alternative to traditional agriculture. However, the maintenance of such systems requires close monitoring of environmental conditions to ensure the well-being of both aquatic life and plants.

The complexity of maintaining optimal conditions in an aquaponics system becomes a challenge, especially for small-scale farmers or educational users with limited technical knowledge. Factors such as water level, quality, and temperature need constant observation to prevent system failure or damage to the ecosystem. Manual monitoring introduces risks of delayed intervention and inconsistent data, which can compromise the entire system.

With recent advancements in Internet of Things (IoT) technologies, it has become feasible to create intelligent monitoring systems that are both affordable and reliable. IoT systems can automate data collection, trigger alerts, and even support decision-making processes. AquaGuard is designed with this in mind—a smart solution tailored to meet the needs of aquaponics enthusiasts who require a dependable monitoring tool.

This chapter outlines the context and need for a smart aquaponics monitoring system and introduces the role AquaGuard plays in revolutionizing traditional methods.

1.1 Motivation

The inspiration for AquaGuard comes from the increasing popularity of aquaponics in sustainable agriculture and the clear need for real-time monitoring tools in these systems. Small-scale farmers and educational setups often lack the resources and expertise required to manage aquaponics manually. Moreover, traditional practices are prone to human errors, delays, and inefficiencies that can lead to system imbalance, affecting both fish and plant health.

Through automation and real-time data monitoring, the AquaGuard system aims to simplify the process of aquaponics management. By leveraging cost-effective IoT components and Bluetooth-based alerts, AquaGuard offers a practical solution to bridge the gap between technological advancement and accessibility for everyday users.

1.2 Objectives

The primary objectives of this project are as follows:

1. To monitor the water level in an aquaponics tank in real-time and alert users when it falls below a safe threshold.
2. To estimate water quality using a soil moisture sensor repurposed for detecting changes in water conductivity.
3. To measure and track water temperature continuously to ensure optimal conditions for aquatic and plant life.
4. To send instant alerts via Bluetooth to a mobile device when any parameter exceeds its safe operating range.

5. To reduce manual monitoring through automation, thereby saving time and effort.
6. To design a scalable, low-cost, and low-power system that can be easily adopted by individuals, schools, or small-scale farms.

CHAPTER 2

LITERATURE REVIEW

Recent advances in smart agriculture and aquaculture have seen a surge in the development of IoT-based monitoring systems. These systems leverage sensors and wireless communication to automate real-time data collection and monitoring. The integration of technology into aquaponics has opened up new possibilities for enhancing productivity, reducing human error, and ensuring system stability.

Several research studies and practical implementations have explored the use of wireless sensor networks, cloud-based data analytics, and artificial intelligence in smart farming. These systems typically monitor water quality parameters such as pH, temperature, turbidity, and dissolved oxygen levels, providing farmers with insights and timely alerts. Despite their potential, many of these systems face limitations like high cost, dependence on stable internet connectivity, and complex setup processes, making them inaccessible to small-scale users.

This chapter presents a comparison between existing systems and the proposed AquaGuard solution, highlighting the challenges of current methods and the innovations introduced in this project.

2.1 Existing System

Various existing systems have been deployed in fish farming and integrated agriculture environments. Examples include cloud-based platforms that receive sensor data for analysis, and AI-powered digital twins that simulate farm environments to optimize operations. Notable systems like the IoT-Based Cloud Solution for Integrated Rice-Fish Farming rely on cloud servers for data transmission and decision-making, while Digital Twin-based systems offer high-fidelity virtual models for managing large-scale operations.

These systems often use multiple sensors to monitor parameters such as pH, turbidity, water level, and meteorological factors. However, the complexity of their implementation limits their use in small-scale or educational contexts. Moreover, their reliance on internet connectivity and expensive hardware components makes them unsuitable for resource-constrained environments.

2.1.1 Advantages of the existing system

- Enable real-time data collection and monitoring through remote access.
- Provide high-level analytics and AI-assisted decision-making in large-scale operations.
- Increase yield and system efficiency by optimizing environmental parameters.
- Allow cloud-based historical data storage for long-term trend analysis.

2.1.2 Drawbacks of the existing system

- Dependence on stable internet connectivity limits deployment in remote or rural areas.
- High costs associated with sensors, connectivity modules, and cloud subscriptions.
- Complex calibration and maintenance of sensors require technical expertise.
- Data privacy and security concerns when transmitting sensitive farm data to the cloud.

2.2 Proposed System

The proposed system, AquaGuard, offers a cost-effective, offline-capable solution designed specifically for small-scale or beginner aquaponics users. It uses readily available and budget-friendly components such as a water level sensor, soil moisture sensor (repurposed), temperature sensor, and Bluetooth module for wireless communication. Unlike cloud-based systems, AquaGuard sends real-time alerts directly to a nearby mobile device, ensuring faster response times and eliminating internet dependency.

The entire system is modular and scalable, allowing users to add more sensors or upgrade communication methods (e.g., to Wi-Fi or LoRa) in the future.

2.2.1 Advantages of the proposed system

- Operates without internet connectivity using Bluetooth for local alerts.
- Low cost and low power consumption make it accessible for hobbyists and students.
- Simplified installation and maintenance, even for users with minimal technical knowledge.
- Modular design supports future expansion or upgrades.

CHAPTER 3

SYSTEM DESIGN

3.1 Development Environment

The AquaGuard system was developed using an Arduino-compatible microcontroller, allowing easy interfacing with sensors and modules. The Arduino IDE was chosen due to its simplicity and vast community support. For hardware interfacing, standard analog and digital sensors were used, along with a Bluetooth HC-05 module.

3.1.1 Hardware Requirements

- **Microcontroller: Arduino Uno or compatible board**
- **Water Level Sensor: Float-type or capacitive level sensor**
- **Soil Moisture Sensor: Repurposed to estimate water conductivity**
- **Temperature Sensor: DS18B20 digital temperature sensor**
- **Bluetooth Module: HC-05 for wireless data transmission**
- **Power Supply: USB-powered**
- **Connecting Wires & Breadboard: For circuit assembly**

Microcontroller

The microcontroller serves as the central processing unit of the system. An Arduino UNO or any compatible board is used to manage inputs from sensors and execute control logic for outputs, forming the core of the electronic system.

Water Level Sensor

A float-type or capacitive water level sensor is used to detect the presence and height of water within a container or tank. It enables the system to monitor and respond to water level changes effectively.

Soil Moisture Sensor

This sensor, though originally intended for measuring soil moisture, is repurposed here to estimate water conductivity. It helps assess water quality or contamination levels through electrical conductivity variations.

Temperature Sensor

The DS18B20 is a digital temperature sensor known for its accuracy and ease of use with Arduino. It measures the surrounding temperature, allowing the system to track environmental conditions.

Bluetooth Module

The HC-05 Bluetooth module facilitates wireless communication between the Arduino and a mobile device. It enables real-time data transmission and remote system monitoring or control.

Power Supply

The system can be powered either through a 9V battery or via a USB connection to ensure portability and flexibility in deployment.

Connecting Wires & Breadboard

Jumper wires and a breadboard are used to connect all components without soldering. This setup allows for quick prototyping, testing, and modification of the circuit.

3.1.1 Software Requirements

- **Arduino IDE**: For writing and uploading code to the microcontroller
- **Mobile Bluetooth Terminal App**: To receive and display alerts
- **Embedded C/C++**: Programming language for firmware development
- **Serial Monitor**: For debugging and real-time data checks

CHAPTER 4

PROJECT DESCRIPTION

4.1 SYSTEM ARCHITECTURE

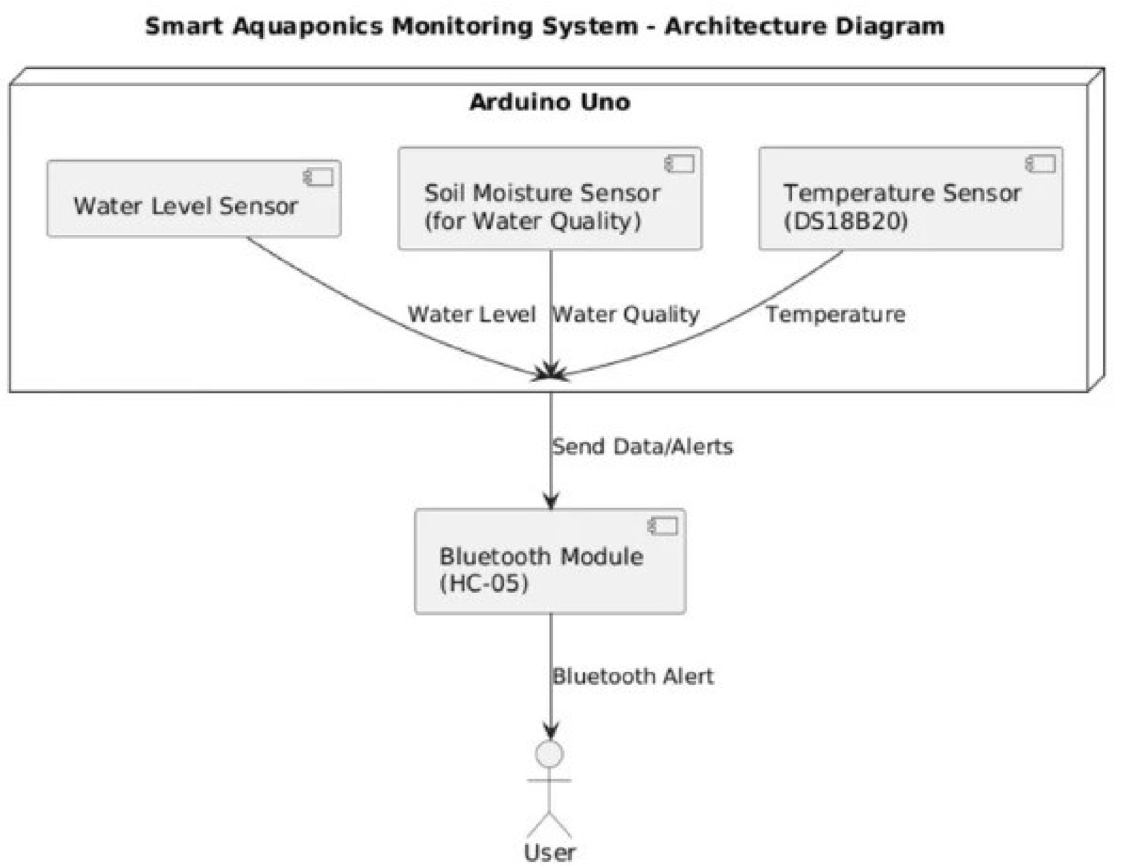

Fig 4.1 System Architecture

4.2 METHODOLOGY

1. Sensor Calibration

This initial phase involves manually testing each sensor to verify and adjust its readings. Calibration ensures that sensors such as the water level sensor, soil moisture (conductivity) sensor, and temperature sensor provide accurate and consistent data. For example, known water levels or temperatures are used to fine-tune sensor thresholds. This step is crucial to avoid false alerts and maintain system reliability.

2. System Integration

After calibration, all hardware components—including sensors, microcontroller (Arduino UNO), Bluetooth module, power supply, and indicators like LEDs—are assembled on a breadboard. Jumper wires are used to interconnect the components without soldering. The firmware, written in Embedded C/C++ using the Arduino IDE, is uploaded to the microcontroller to define the behavior of each component and manage data flow between them.

3. Testing Phase

Once integrated, the entire system is tested under controlled and simulated conditions. This includes changing water levels, introducing different conductivity levels, and adjusting temperature to observe if the system responds correctly. The alert mechanism is verified by checking whether correct signals are sent to the Bluetooth app and whether LEDs indicate system status appropriately.

4.Deployment

After successful testing, the system is installed in a small-scale aquaponics setup for field evaluation. This real-world application helps assess the performance of the system over time, in dynamic environmental conditions. The deployment also validates the practical usability of the wireless alerts and the robustness of the setup.

5. Data Logging

During both testing and deployment, data from the sensors are displayed in real-time on the Arduino Serial Monitor. This allows to review system behavior, debug potential issues, and keep a log of environmental data for further analysis.

.

CHAPTER 5

RESULTS AND DISCUSSION

The AquaGuard system was tested under various environmental conditions to validate its functionality. Simulations showed that the sensors effectively detected changes in water level, conductivity (used to approximate water quality), and temperature. Bluetooth alerts were received reliably within a 10-meter radius.

The results indicated that:

- The water level sensor triggered alerts when the tank dropped below 40%.
- The soil moisture sensor detected unusual conductivity, suggesting possible contamination.
- Temperature variations outside 20–30°C triggered appropriate warnings.
- Overall, the system performed consistently and achieved its primary objective of real-time monitoring with minimal manual intervention. Some limitations include potential interference in Bluetooth range and the need for periodic sensor cleaning.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

AquaGuard successfully demonstrates how IoT technologies can be harnessed to improve the monitoring and maintenance of aquaponics systems. By automating the detection of critical environmental parameters and delivering real-time alerts, it reduces manual workload and increases system reliability. Its simplicity, affordability, and scalability make it a valuable tool for educational and small-scale use cases.

6.2 Future Work

Future enhancements can include:

- Integration with cloud platforms for remote access and historical data analysis.
- Use of advanced sensors like pH, turbidity, and dissolved oxygen for more precise water quality monitoring.
- Replacement of Bluetooth with Wi-Fi or LoRa for extended communication range.
- Development of a mobile app for real-time visualization and data logging.

APPENDIX

SOFTWARE INSTALLATION

Arduino IDE

To run and mount code on the Arduino NANO, we need to first install the Arduino IDE. After running the code successfully, mount it.

Sample code

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SoftwareSerial.h>

#define ONE_WIRE_BUS 7          // Temperature sensor
#define LEVEL_SENSOR A1         // Water level sensor
#define QUALITY_SENSOR A0       // Water quality sensor (soil sensor)

SoftwareSerial BT(10, 11);     // Bluetooth module connected to D10 (RX), D11 (TX)

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

// Calibrated sensor range
const int sensorDry = 500;     // Analog value when sensor is dry (top)
const int sensorWet = 403;     // Analog value when sensor is fully submerged (bottom)

void setup() {
  Serial.begin(9600);          // For debugging
  BT.begin(9600);              // For Bluetooth
  sensors.begin();

  Serial.println("System Initialized");
  BT.println("Bluetooth Connected ✅");
}

void loop() {
  // Read temperature
```

```
sensors.requestTemperatures();
float tempC = sensors.getTempCByIndex(0);

// Read raw water level value
int rawLevel = analogRead(LEVEL_SENSOR);

// Map raw analog value to percentage (dry → wet)
int levelPercent = map(rawLevel, sensorDry, sensorWet, 0, 100);
levelPercent = constrain(levelPercent, 0, 100);  // Clamp to 0–100%

// Read water quality sensor value
int rawQuality = analogRead(QUALITY_SENSOR);

// ---- Debug Output ----
Serial.print("Raw Level: "); Serial.println(rawLevel);
Serial.print("Water Level: "); Serial.print(levelPercent); Serial.println(" %");
Serial.print("Temperature: "); Serial.print(tempC); Serial.println(" °C");
Serial.print("Water Quality: "); Serial.println(rawQuality);
Serial.println("--------------------------");

// ---- Bluetooth Output ----
BT.print("BT:TEMP:"); BT.print(tempC); BT.println("°C");
BT.print("BT:LEVEL:"); BT.print(levelPercent); BT.println("%");
BT.print("BT:QUALITY:"); BT.println(rawQuality);

if (tempC > 35 || tempC < 10) {
  BT.println("BT:WARNING:Temperature out of range!");
}

if (levelPercent < 30) {
  BT.println("BT:ALERT:Water level low!");
}

if (rawQuality < 300) {
  BT.println("BT:QUALITY_STATUS:Contaminated ❌");
} else if (rawQuality < 500) {
  BT.println("BT:QUALITY_STATUS:Moderate ⚠");
} else {
  BT.println("BT:QUALITY_STATUS:Clean ✅");
}

BT.println("-----");  delay(2000);}
```

REFERENCES

[1] IoT-Based Cloud Solution for Intelligent Integrated Rice-Fish Farming, SciSpace.

[2] Digital Twin-Based Intelligent Fish Farming, ScienceDirect.

[3] AJPO Journals – IoT Monitoring Systems in Fish Farming.

[4] Arduino Documentation – https://www.arduino.cc

[5] HC-05 Bluetooth Module Datasheet

[6] DS18B20 Temperature Sensor Datasheet