October University for Modern Science & Arts

Faculty of Engineering

Department of Computer Systems Engineering

HydroFarmIoT: IoT-Integrated Hydroponic Vertical Farming System

A Graduation Project

Submitted in Partial Fulfilment of the Requirements of

B.Sc. Degree in Computer Systems Engineering

Part II

Prepared By

Hussein Gharib Subhi Mahmoud Ahmed	213423
Ibrahim Attia Ibrahim Attia Gabr	210071

Supervised By

Dr. Ahmed Ayoub

2024/2025

Abstract

The rising demand for food, driven by global population growth and urbanization, is placing significant pressure on traditional farming methods, which depend on cultivable land and abundant water resources. As these resources become increasingly scarce due to urban expansion, climate change, and water scarcity, hydroponic systems offer a sustainable alternative, enabling soil-less agriculture with reduced water consumption. However, many existing systems lack automation and still require labour-intensive manual management. HydroFarmIoT: IoT-Integrated Hydroponic Vertical Farming System addresses these issues by utilizing IoT technology to automate the monitoring of key parameters such as pH, nutrient levels, temperature, and water level. Through sensors and actuators in a vertical farming setup, the system autonomously maintains pH level conditions. This project aims to enable year-round cultivation in urban environments, providing a scalable and sustainable solution for addressing food security challenges in urbanized regions.

Acknowledgement

We would like to express our deepest gratitude to all those who supported and guided us throughout the preparation of this work.

First and foremost, we extend our heartfelt thanks to our esteemed supervisor, **Dr. Ahmed Ayoub**, for his constant support, invaluable advice, and expert guidance. His encouragement and insights have been instrumental in shaping our understanding and ensuring the success of this research.

We are also profoundly grateful to the **Faculty of Engineering** and the **Head Of Computer Systems Engineering Department Dr Samer** for providing us with the resources and environment needed to carry out this project. Their commitment to fostering innovation and academic excellence has greatly inspired us.

Finally, we extend our thanks to our families and friends, whose patience and encouragement provided us with the motivation to persevere during challenging times.

To everyone who contributed to this project in any way, we offer our sincere gratitude. This work would not have been possible without your support.

Table of Contents

List of Abbreviations

CEA	Controlled Environment Agriculture
EC	Electrical Conductivity
IOT	Internet Of Things
LED	Light Emitting Diode
NFT	Nutrient Film Technique
pH	potential of hydrogen
TDS	Total Dissolved Solids
WSN	Wireless Sensor Networks

List of Figures

Chapter 1 Introduction

1.1. Field Survey

Introduction

On October 8th, 2024, our visit to Al-Hajjaj Farm was conducted to gather crucial insights for the graduation project titled 'HydroFarmIoT,' which focuses on smart farming techniques, especially hydroponics. This project aims to design and implement an IoT-integrated vertical farming system utilizing hydroponic methods to enhance crop production in a sustainable and resource-efficient manner. The visit provided hands-on observations of various farming techniques, including hydroponics, aquaponics, and sandponics, which are all advanced methods employed in modern smart farming.

Types of Smart Farming

Sandponic: A hybrid farming technique using sand as a growing medium, offering better control over water and nutrients. It's ideal for areas with poor soil, delivering nutrients directly to plant roots.

Aquaponics: Combines fish farming with hydroponics, where fish waste fertilizes plants, and plants purify the water. It is highly water-efficient and minimizes the need for chemical inputs.

Hydroponics: A soil-free system using nutrient-rich water for plant growth, allowing for better control, faster yields, and reduced pests. It is highly suitable for urban, water-efficient farming.

How Hydroponic Work

Hydroponics involves growing plants in a water-based solution mixed with essential nutrients like nitrogen, phosphorus, and potassium. These nutrients are typically absorbed by plants from soil, but in hydroponics, they are delivered directly to the roots through the water. The plants are supported by an inert growing medium such as coconut coir, perlite, or rock wool, which helps stabilize the roots while allowing water and nutrients to flow through. The roots are either submerged in the nutrient solution or kept in a moist environment to continuously absorb the necessary nutrients.

Hydroponic systems are often housed in controlled environments, such as greenhouses, where factors like light, temperature, and humidity can be optimized for plant growth. The system is highly resource-efficient, using less water than traditional soil-based farming and producing faster plant growth. The integration of IoT sensors in systems like HydroFarmIoT allows for real-time monitoring and control, further enhancing the efficiency of the hydroponic system.

Real Samples for Expansion and Home Production

During the visit, a model of green Batavia plants grown in hydroponic tubes was presented. These plants were 15 days old, and the system utilized 10 cm-wide tubes as seen in **Figure 1.1**. However, one batch of seeds showed signs of poor growth due to seed rot as seen in **Figure 1.2**, emphasizing the importance of using viable seeds for successful cultivation. Proper seed selection is critical to achieving optimal growth and yield in hydroponic systems.

Figure 1-1:Green Batavia plants grown in hydroponic tubes,15 days old, system utilized 10 cm-wide tubes. (Source: Authors)

Another sample demonstrated the use of floating foam boards for growing eggplants and tomatoes as seen in **Figure 1.3**. This technique allows the plants to float on a nutrient solution while the roots extend down into the water. However, the method has limitations for heavier fruits, as the foam board may not be able to support the weight, potentially causing it to sink. This highlights the need for careful consideration of plant types when designing hydroponic systems for specific crops.

Figure 1-2: Batch of seeds showed signs of poor growth due to seed rot. (Source: Authors)

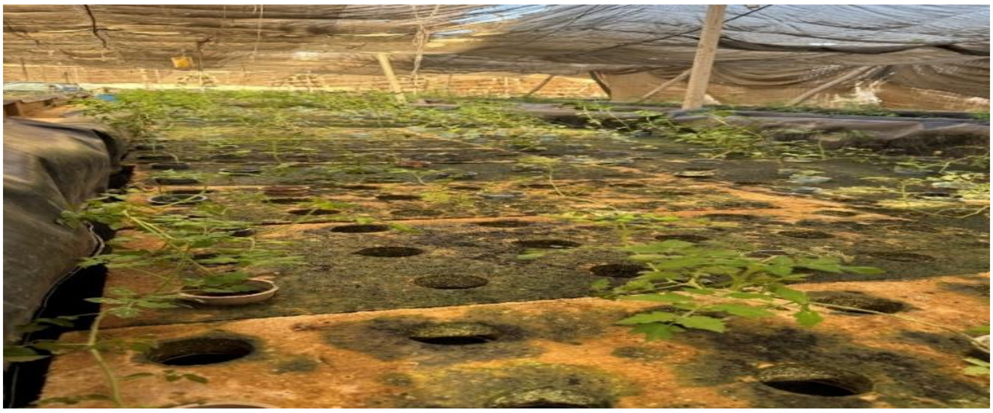

Figure 1-3: Floating foam boards for growing eggplants and tomatoes. (Source: Authors)

For small-scale, home-based farming, miniaturized hydroponic models were shown as seen in **Figure 1.4**, which can be placed on balconies or rooftops. These models use motors to lift the nutrient solution from tanks, cycling it through the system to maintain continuous water flow. Such systems are ideal for urban dwellers who want to grow their own vegetables in limited spaces.

Figure 1-4: Small-scale, home-based farming, miniaturized hydroponic models. (Source: Authors)

Important Tools Used

Several tools and equipment are essential for the proper functioning of a hydroponic system. The primary component is a tank that holds the nutrient solution as seen in **Figure 1.5**. The solution is recirculated through the system, with a typical system using approximately 12 meters of solution. Even with a complete 10-week growth cycle, only a fraction (1 to 2 meters) of the solution is consumed, emphasizing the water efficiency of hydroponic farming. The tank is equipped with a mesh system to prevent leakage.

Figure 1-5: Tank that holds the nutrient solution. (Source: Authors)

Figure 1-6: Molds used in various pipes in house the seeds. (Source: Authors)

Molds of different sizes are used to fit the various pipes in the system as seen in **figure 1.6**. The deeper the pipes, the better the support for the plant roots. Additionally, temperature control tools, such as water sprinklers and partial insulation against sunlight as seen in **figure 1.7**, are employed to maintain an optimal environment for plant growth. Effective temperature management ensures that plants remain healthy and continue to grow at an accelerated rate, regardless of external weather conditions.

Figure 1-7: Temperature control tools, such partial insulation against sunlight (left) and as water sprinklers (right) (Source: Authors)

Summary

The visit to Al-Hajjaj Farm provided practical insights into smart farming techniques, with a particular focus on hydroponics. Hydroponics offers a sustainable, water-efficient method of farming that eliminates the need for soil and provides plants with essential nutrients through a controlled water-based system. By integrating IoT technology, the HydroFarmIoT project aims to further optimize hydroponic farming by automating environmental controls and improving resource management. The observed models of hydroponic systems, along with the tools and techniques used, serve as a strong foundation for expanding the application of smart farming in urban and rural settings alike.

1.2. Problem definition

The matter investigated by hydroponic vertical smart farming relates to the increasing complexities associated with global food security, environmental sustainability, and resource scarcity, particularly in urban areas. Conventional agricultural practices encounter numerous constraints, which include [1]:

-Restricted Arable Land: The processes of urbanization and industrialization are diminishing the accessibility of fertile land for conventional agriculture, especially within densely populated urban environments.

-Water Scarcity: A substantial proportion of global water usage is attributable to agriculture, and in areas characterized by limited freshwater availability, traditional irrigation techniques frequently exhibit significant inefficiencies.

-Climate Dependence: Conventional agricultural methodologies are vulnerable to unpredictable climatic variations, shifts in climate patterns, and seasonal inconsistencies, leading to inconsistent agricultural yields.

-Suboptimal Resource Utilization: Soil-based agricultural systems frequently lead to nutrient depletion and ineffective utilization of resources such as water, fertilizers, and spatial dimensions. This inefficiency can induce environmental harm through phenomena such as soil erosion and excessive reliance on pesticides.

-Transportation: Fruits and vegetables sourced from remote agricultural regions often demand considerable transit efforts, thus amplifying greenhouse gas outputs and inflating food prices.

-Expanding Population and Urbanization: The global demographic is on the rise, particularly within urban locales, thereby amplifying the demand for fresh produce even as the availability of farmland continues to diminish.

1.3. State of the art

The current model in hydroponic vertical smart agriculture is marked by revolutionary developments that combine complex hydroponic strategies, vertical growing practices, and smart technologies, including the Internet of Things (IoT), artificial intelligence (AI), and automation, with the intention of advancing agricultural productivity and ecological sustainability. The following section articulates the major trends and innovations that represent the current state of the art:

1-IoT Integration and Automation

Hydroponic vertical smart agricultural frameworks deploy IoT sensors to precisely oversee vital parameters including temperature, humidity, and nutrient concentrations. These automated systems facilitate the dynamic adjustment of environmental conditions, thereby augmenting precision in resource utilization and minimizing labour requirements. Automation plays a crucial role in improving agricultural effectiveness through the provision of remote supervision and immediate assessment. [2]

2-Artificial Intelligence (AI) and Machine Learning

Artificial intelligence and machine learning techniques are employed to analyze data, forecast plant growth trajectories, and enhance resource distribution. These technological advancements contribute to enhanced crop yields and facilitate more informed decision-making processes. [3]

3-Energy-Efficient Lighting

Light Emitting Diode (LED) lighting systems are routinely utilized to provide optimal illumination for plant growth. These lighting arrangements emulate natural sunlight, providing targeted wavelengths tailored to various growth phases.

4-Vertical Farming Design

Current vertical farming methodologies maximize spatial efficiency through the cultivation of plants in vertically organized layers, thereby making them particularly advantageous for urban environments where land is scarce. Such configurations promote high-density agricultural practices, thereby augmenting productivity within constrained areas. [4]

5-Controlled Environment Agriculture (CEA)

Indoor vertical agricultural facilities are situated within climate-regulated environments, permitting continuous cultivation throughout the year irrespective of external climatic conditions.

6-Optimized Nutrient Delivery

Hydroponic frameworks administer nutrient-enriched aqueous solutions directly to the root systems of plants. These systems possess the capability to modify nutrient concentrations in real-time, ensuring that plants receive optimal nourishment throughout their growth phases, thereby enhancing overall operational efficiency.

7-Sustainable Water Management

Hydroponic agricultural practices utilize closed-loop water systems that recycle water, resulting in a substantial reduction in overall consumption levels. Water efficiency represents a critical advantage, with certain systems achieving up to 90% lower water usage compared to conventional agricultural methodologies.

8-Urban and Modular Farms

Urban vertical farms, encompassing container-based and rooftop agricultural systems, are increasingly gaining traction. These enterprises are specifically designed for urban settings, thereby diminishing the necessity for long-distance transportation of agricultural products.

1.4. Thesis Objectives

1. Develop an IoT-based hydroponic vertical farming system that monitors and controls key environmental parameters including water pH, temperature, and nutrient concentration.
2. Implement real-time monitoring and automated control of pH to maintain predefined conditions suitable for plant growth.
3. Desing a modular and scalable system architecture suitable for urban spaces, incorporating automation to reduce manual system interaction

1.5. Thesis structure

The dissertation concerning hydroponic vertical smart farming encompasses the subsequent framework:

• **Chapter 1: Introduction** – Presents the context of the project, highlights challenges in traditional agriculture, defines the research objectives, and outlines the structure of the thesis.

• **Chapter 2: Literature Review** – Reviews existing hydroponic and vertical farming systems, explores IoT applications in agriculture, and discusses previous scholarly work related to intelligent resource management.

• **Chapter 3: Proposed System Design** – Details the planning and design of the HydroFarmIoT system, including requirement analysis, system specifications, and relevant standards.

• **Chapter 4: Implementation** – Explains the hardware setup, software development, and system integration for the full implementation of the automated hydroponic system.

• **Chapter 5: Testing** – Presents the experimental setup and discusses the outcomes of hardware and software testing, including test cases and results.

- **Chapter 6: Cost Analysis** – Provides a comprehensive breakdown of the system's total cost and individual component pricing.

- **Chapter 7: Time Plan** – Describes the project timeline and the stages of development across the academic year.

- **Chapter 8: Conclusion and Future Work Plan** – Summarizes the main findings, contributions, and outlines potential directions for future improvements or industry applications.

Chapter 2 Literature Review

2.1. Optimizing Nutrient Levels with IoT

Introduction

The soilless cultivation method termed hydroponics has garnered interest in contemporary agriculture [5], especially in urban environments with constrained space and resources. The Nutrient Film Technique (NFT) is a popular hydroponic method for growing leafy greens like spinach. The performance of the system depends on maintaining the proper ratio of nutrients, pH, and environmental variables. In "Smart Hydroponic Systems: Optimizing Nutrient Levels with IoT Connectivity," Kulkarni et al. [6] describe an Internet of Things (IoT)-based approach to managing these parameters to maximize crop yield. The research aims to integrate Internet of Things technologies, including wireless sensor networks (WSN) and ESP32 microcontrollers, to develop a smart hydroponic system that offers real-time monitoring and data-driven decisions for fertilizer management.

Hydroponic systems have variable such as pH, TDS, and EC that require manual monitoring. This is obviously time-consuming and error prone. Also, poor nutrient management might result in less plant growth or, in extreme situations, plant mortality. Through the Internet of Things, this study allows the automation of nutrient monitoring, enabling farmers to remotely monitor important indicators and quickly rectify problems. It is proven that this innovation would increase agricultural output and decrease resource waste, opening the door for more environmentally friendly farming methods.

Method

To monitor pH, TDS, and EC in a hydroponic system, wireless sensor networks are used for continuous monitoring of key environmental parameters. ESP32 microcontrollers manage a real-time data collecting system by connecting sensors to ensure continuous monitoring. After data is transmitted through Wi-Fi to a cloud platform, users can remotely view it on their smartphones.

pH Sensor: Specifically designed for hydroponics, a pH-sensitive electrode is utilized to monitor the pH level, crucial for plants to uptake nutrients. To maintain the balance of the nutritional solution, the sensor helps monitor pH changes all day long.

TDS Sensor: TDS sensors offer instant readings by gauging the concentration of dissolved nutrients in the water, indicating nutrient levels. These sensors help create ideal development conditions by allowing for timely adjustments.

The ESP32 microcontroller gathers data from the pH and TDS sensors and serves as the central hub. Sending data wirelessly through Wi-Fi to faraway devices improves the efficiency of monitoring.

This automated device streamlines the nutrient management process by removing the need for constant manual monitoring. **Figure 2.1** illustrates the setup of the pH and TDS detection circuit. This system improves hydroponic techniques using IoT technology, making maintenance easier and increasing productivity.

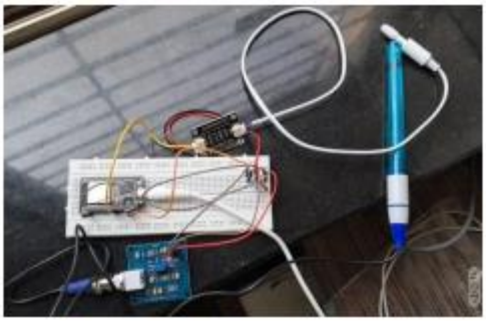

Figure 2-1: Proposed Circuit for pH and TDS sensing [6]

Implementation Process

The smart hydroponic system is centered on two main processes: collecting data and monitoring in real-time. The system is created to gather data constantly on different environmental factors that impact plant growth. The pH, TDS, and EC levels are constantly monitored by sensors to ensure the plants are in their best health.

Data Collection: Sensors are strategically positioned throughout the hydroponic setup to monitor pH levels and nutrient concentrations at various locations in the water circulation. The ESP32 microcontroller collects and processes data, then sends it to a cloud platform.

Cloud Connectivity: The ESP32 microcontroller sends data wirelessly through Wi-Fi to a server based in the cloud, allowing for immediate access to the information globally. Farmers can access a smartphone interface to monitor the present condition of the hydroponic system, which enables them to remotely modify the nutrient levels.

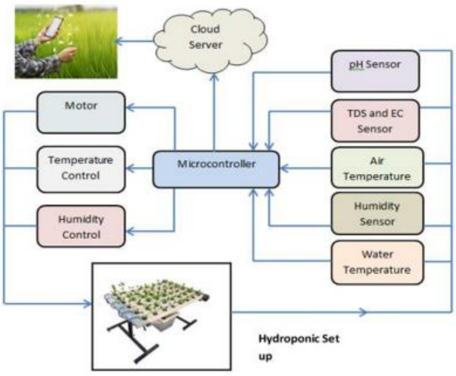

Figure 2-2: IoT System Architecture for Hydroponic Set Up [6]

The IoT architecture depicted in **Figure 2.2** showcases how the sensors interact with the ESP32 microcontroller and the process of storing and accessing data on the cloud platform. The system becomes truly "smart" by seamlessly integrating sensors and cloud computing, enabling precise control over nutrient management in the hydroponic setup.

Findings

The main results of the research emphasize how crucial it is to keep nutrient levels and environmental conditions in check for the best plant growth. The scientists carried out multiple experiments on spinach plants, examining how different TDS and pH levels impacted them.

TDS Levels: Plant growth was most optimal when the TDS levels were kept within a range of 1000 to 1200 ppm. At this stage, spinach plants displayed strong root growth and healthy green leaves, as depicted in **Figure 2.3**. Decreased TDS levels (800-900 ppm) led to reduced growth and lighter leaves, suggesting that the plants did not have enough nutrients to thrive.

Figure 2-3: Maximum height at pH=6, TDS=1100ppm [6]

The ideal pH for spinach growth was found to be 5.5 to 6.5. Plants that were cultivated within this pH range exhibited uniform growth in both their shoots and roots. Nevertheless, plant growth was notably impacted when the pH decreased below 5.0, leading to the emergence of brown spots on the leaves. **Figure 2.4** shows the root development seen under ideal pH and TDS conditions.

Figure 2-4: Root growth at pH=6, TDS= 1100ppm. [6]

These results highlight the significance of accurate management of nutrient levels and environmental factors in hydroponic setups. Through the utilization of IoT-driven monitoring, the system can sustain these levels autonomously, lessening the requirement for hands-on involvement and decreasing the chance of human mistakes.

Limitations

While the research provides a strong structure for intelligent hydroponic systems, there are restrictions that must be considered. Initially, the system was primarily evaluated in winter conditions, which implies that its performance in different seasons, especially summer, remains not completely clear. The paper concentrates solely on leafy vegetables such as spinach, lettuce, and cilantro, restricting its relevance to other plant varieties. In future research, it is important to broaden the variety of crops examined and investigate how well the system functions in different environmental conditions.

Summary

This study shows a notable progress in precision farming by combining IoT and wireless sensor networks in hydroponic agriculture. This study created a clever hydroponic system that allows for constant supervision of important factors such as pH, TDS, and EC to maintain ideal growth conditions for leafy greens. Automating nutrient management reduces manual intervention, minimizes human error, and enhances crop yield. The results indicate that the optimal conditions for spinach growth are keeping TDS levels within 1000 to 1200 ppm and maintaining a pH range of 5.5 to 6.5. Although it has its constraints, this study offers important perspectives on the future of IoT-powered farming and paves the way for additional progress in the sector.

2.2. IoT- Enabled system

Introduction

With the growth of urban areas and decreasing farmland, vertical farming is becoming a sustainable method for producing large amounts of food in small areas. Responding to this challenge, Harn Tung Ng et al. [7] introduce an IoT-enabled system that allows for real-time monitoring and control of environmental conditions in vertical farming. The system is created to oversee important factors like air temperature, humidity, and UV light exposure while offering users an easy-to-use interface for managing farming operations from a distance.

This IoT system uses microcontrollers and sensors to enhance environmental conditions in vertical farms, resulting in increased crop yields and operational efficiency. The article also emphasizes the advantages of automating vertical farming operations, including lower energy usage and greater scalability.

Method

The suggested IoT system relies on a combination of hardware and software elements to operate smoothly.

Computer hardware:

The DHT22 sensor is utilized to consistently monitor the temperature and humidity of the air in the vertical farming system.

The set-up combines an Arduino Nano with a NodeMCU ESP8266 Wi-Fi module to handle sensor information and send it to an online platform. These parts help with instant wireless data transmission and management of important farming tasks.

Relays and Multiplexers: A 4-channel relay module operating at 12 volts is utilized for controlling systems such as UV lighting. The relay enables effective changing of voltage needs for lighting and water systems.

Technology-related computer programs or applications.

The Arduino IDE is used for programming the system, while LabVIEW is utilized for developing the user interface. The information gathered is sent to the System Link IoT cloud, enabling users to oversee and manage environmental conditions via a web browser or mobile device.

This hardware and software combination enables accurate remote monitoring and control of environmental parameters. The proposed system's circuit connections are illustrated in **Figure 2.5.**

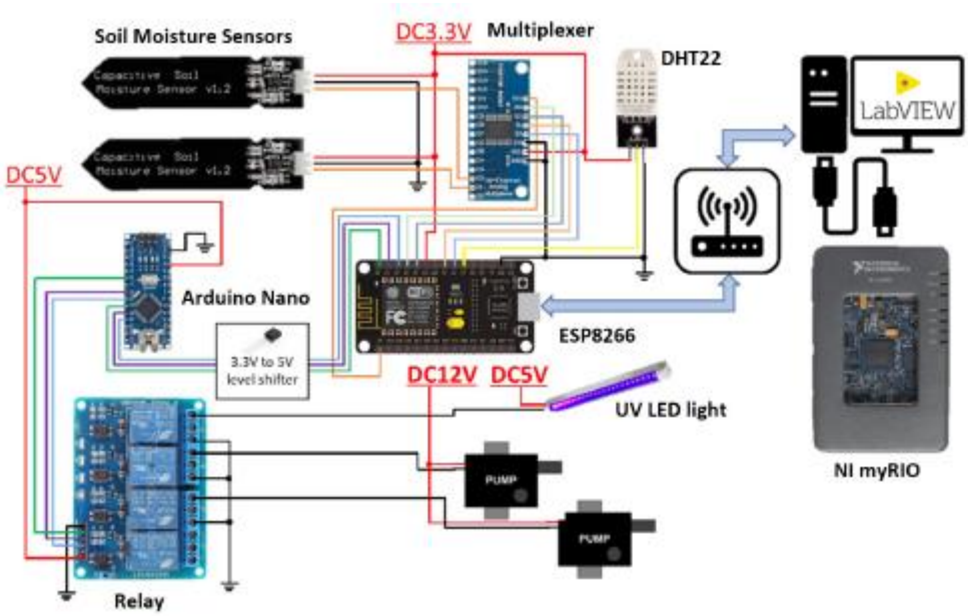

Figure 2-5: Detailed circuit connections of proposed system [7]

Implementation Process

The system functions by means of three main procedures.

Surveillance: The sensors consistently monitor air temp, humidity, and further environmental elements, with the microcontrollers analysing data and sending it to the cloud.

Cloud-Based Access: The sensor data is sent to the SystemLink cloud platform through the NodeMCU ESP8266 Wi-Fi module. Users can retrieve this information instantly from any device connected to the internet, enabling them to remotely monitor the conditions of vertical farms.

Regulation System: The system allows users to manage environmental conditions, like controlling UV lighting on/off, through the web interface. Instructions are sent from the interface to the microcontroller, which turns on or off components such as UV lights. **Figures 2.6 and 2.7** depict the program flow of the Arduino Nano and ESP8266 Wi-Fi module, demonstrating how the system guarantees uninterrupted operation and instant control.

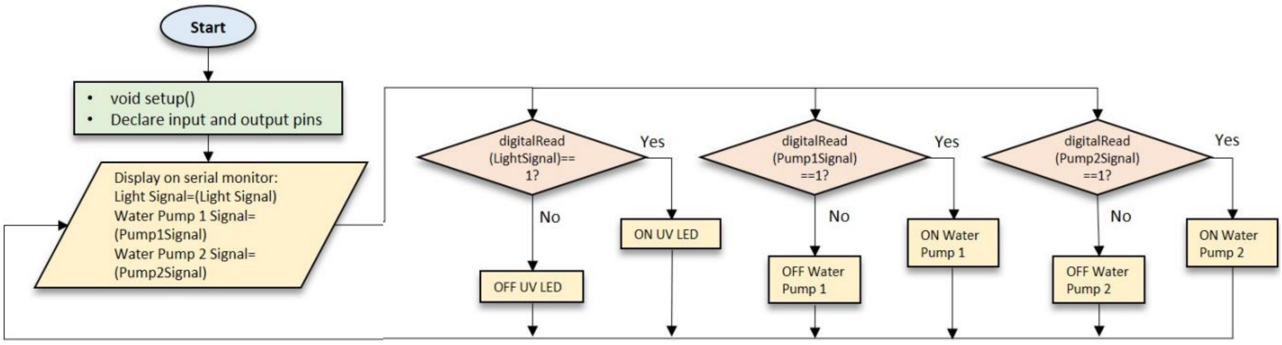

Figure 2-6: Program flowchart for Arduino Nano [7]

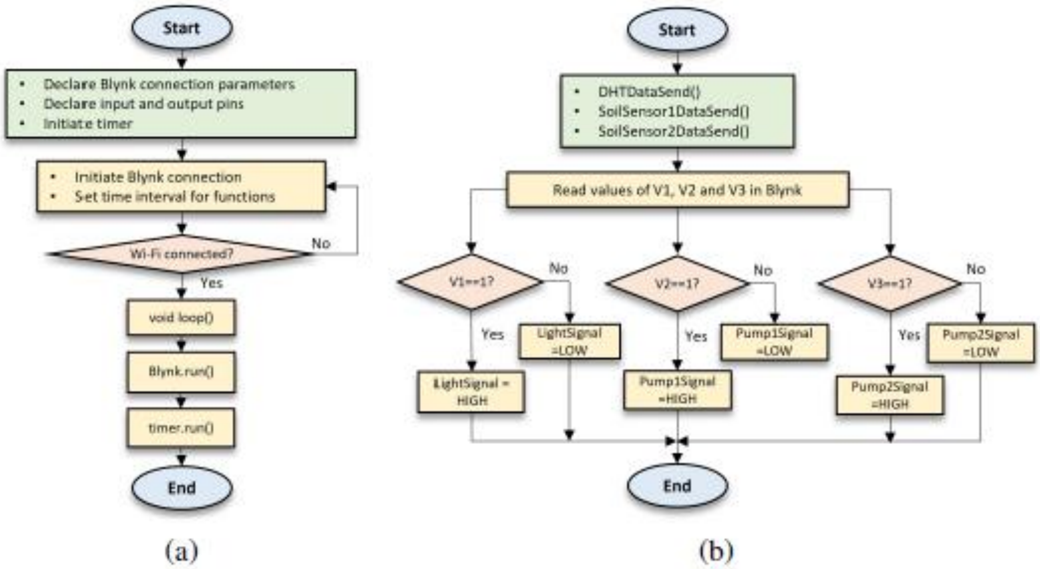

Figure 2-7: Program flowcharts of (a) ESP8266 and (b) Blynk.run () function. [7]

This system enhances environmental management in vertical farming and streamlines user interaction.

Findings

The model showed efficient monitoring and control functions for vertical farming settings. Some important discoveries are:

Monitoring data in real time: The system effectively recorded current temperature and humidity data, showing the readings on a web interface that refreshed every 10 seconds. The data could be viewed in numerical format as well as in charts, as illustrated in **Figure 2.8**.

Figure 2-8: The UI developed using G web development software. [7]

UV Light Management: The system effectively regulated UV lighting via the online interface, enabling users to set schedules for turning it on/off according to plant requirements. The system interface and hardware status LEDs reflected the activation of UV lights, as depicted in **Figure 2.9.**

(a) (b) (c)

Figure 2-9: User input data in (a) serial monitor, (b) UV lighting system in prototype, and (c) LED indicators for UV Lighting on the NI myRIO board and user interface. [7]

Scalability: Scalability was a key consideration in the design of the system. More sensors or actuators could be smoothly incorporated into the system, allowing it to be suitable for bigger agricultural activities. The system hosted on the cloud can also be integrated with artificial intelligence in the future for predicting farming management.

Limitations

Even though the system was successful, some constraints were identified.

Latency occurred in the system because of data transmission via the cloud, causing slight delays. While short, these delays could affect real-time control in important situations.

In the future, AI integration is expected to improve the system's monitoring and control capabilities by adding predictive features. Integrating AI could improve its ability to predict environmental changes and optimize farming conditions even more.

Summary

This IoT-enabled system provides an efficient and scalable solution for monitoring and controlling vertical farming environments. The system's real-time monitoring of air temperature, humidity, and UV lighting, combined with its cloud-based control interface, allows for precise environmental management. The flexibility of the system makes it adaptable to various farming scales, and its potential integration with AI opens opportunities for further advancements in smart agriculture.

Although the system faces some challenges, such as brief data transmission delays, its overall performance indicates a strong potential for improving the efficiency and scalability of vertical farming. **Figure 2.8** and **Figure 2.9** illustrate the practical applications of the system, making it a valuable tool for modern agriculture.

2.3. IoT- Enabled Smart Greenhouse System

Introduction

Hydroponic systems have become a viable alternative to traditional farming, arising in response to challenges such as shrinking arable land, population growth, and climate change. Austria et al.'s [8] paper delves into creating an IoT-enabled smart greenhouse system for hydroponic gardens. The goal of this system is to automatically supervise and manage important environmental variables like pH, light, temperature, and humidity. IoT technology integration enables mobile access for real-time data management in farming, enhancing efficiency and sustainability.

The greenhouse system in this study is designed to create a regulated environment for maximum plant growth, offering a viable option for urban agriculture and food security amid shifting environmental circumstances.

Method

The article outlines the hardware and software elements utilized in constructing the IoT-equipped intelligent greenhouse system.

Physical parts of a computer:

- Detectors: A range of detectors are integrated in the system, including a DHT11 to measure temperature and humidity, an LDR module to monitor light levels, a DS18B20 sensor to detect water temperature, and a pH4502C sensor for measuring water pH.

- Arduino Mega: The main processing unit responsible for managing the system and linking all the sensors. It uses an ESP8266 Wi-Fi module to establish real-time data transmission with the cloud.

- Relays and Submersible Pumps: Their purpose is to automatically control irrigation and other system operations using sensor data.

Elements of Software:

- ThingSpeak: Sensor data is stored and displayed using this cloud platform. With the use of MIT App Inventor, users can remotely monitor and control the greenhouse through an Android app.

- Monitoring System: The monitoring system oversees the limits established for temperature, humidity, pH, and light levels. When parameters go beyond or below certain limits, the system activates relays to control the environment by turning on the water pump or changing the lighting.

Figure 2.10 depicts the step-by-step process of building the project, outlining the overall system design which utilizes a variety of components to enable automated plant care and allow users to access real-time data, enhancing its usefulness for hydroponic gardening.

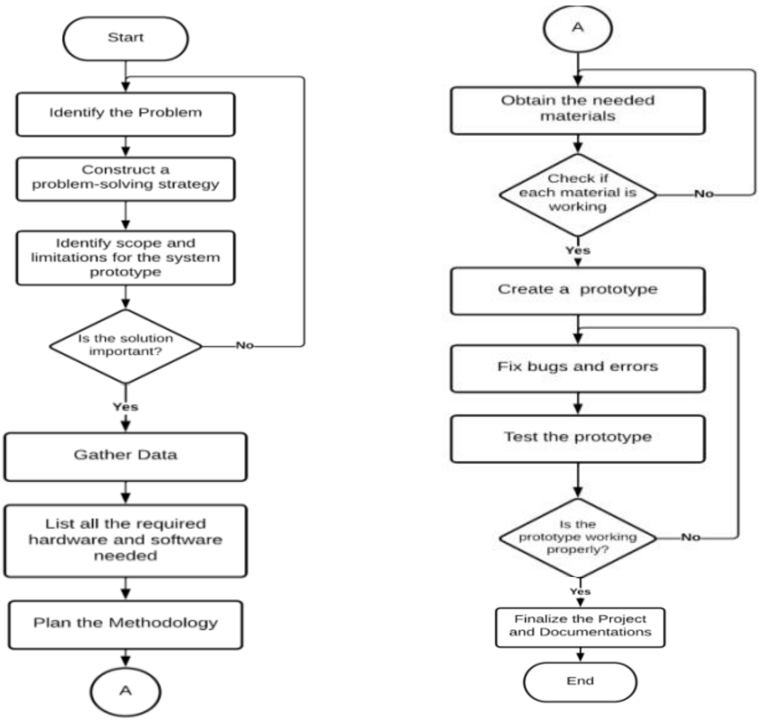

Figure 2-10: Project Construction Flowchart [8]

Implementation Process

The implementation process is broken down into three main phases:

1. Data Collection: Different sensors are used to observe immediate environmental conditions like temperature, humidity, water temperature, light intensity, and pH.

2. Integration with the cloud: Using the Arduino Mega and the ESP8266 Wi-Fi module, sensor data is sent to the ThingSpeak platform, which is based in the cloud. Users can track the greenhouse conditions from a mobile device, making it possible to control it from a distance.

3. Automation and Control: The system is created to automatically adjust to variations in environmental conditions. For example, if the water temperature or pH strays from the optimum range, the system triggers the appropriate relay to control the conditions. The control system flowchart in **Figure 2.11** is responsible for overseeing this automation process.

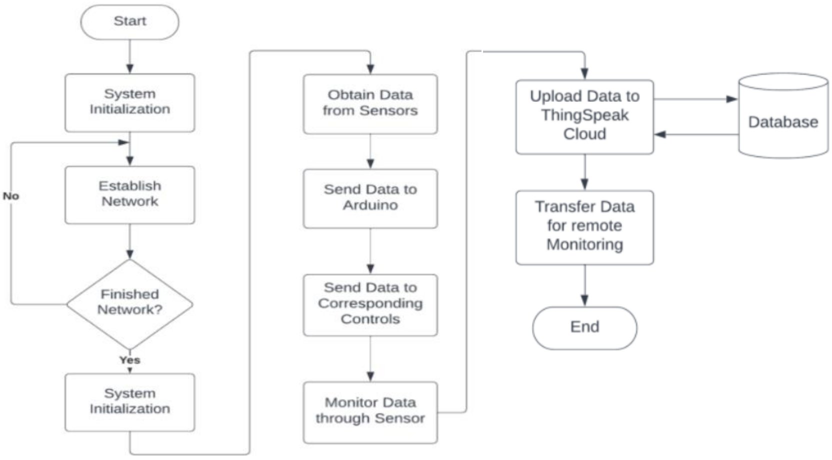

Figure 2-11: Control System Flowchart [8]

This method guarantees effective control of the hydroponic system with minimal human involvement, maximizing plant growth conditions inside the greenhouse.

Findings

The study's findings show that the system effectively preserves ideal environmental conditions for hydroponic farming.

- System Efficiency Monitoring: The system successfully tracked important parameters (temperature, humidity, pH, light, and water temperature) and sent the data to the cloud platform instantly. Data logs from ThingSpeak indicated a steady monitoring of environmental conditions, with recorded values staying within optimal ranges for the growth of hydroponic plants.

- Automation Reliability: The automation system reliably activated actions, like turning on submersible pumps, when pH or water temperature surpassed pre-set limits. Comparing the prototype's performance to commercial systems revealed a slight percentage variance, signifying its precision.

Mobile Application: Users could monitor greenhouse conditions remotely with the Android app created using MIT App Inventor. The application was discovered to be easy for users to navigate and successful in delivering current information on greenhouse conditions. The mobile app's performance and user-friendliness were assessed using the MARS framework, with users agreeing that the app functioned effectively.

Limitations

Although the system functioned effectively, some constraints were noted:

Sensor calibration: In Trial 3 of the sensor data collection, inaccurate readings were sometimes caused by wrong calibration of the pH4502C sensor during testing.

-Complex Wiring: The authors observed that the current wiring configuration is complex and suggest making it simpler in future versions for easier maintenance and installation.

-Energy Source: The system depends on AC power, and the research proposes incorporating solar panels to increase the system's sustainability, especially in isolated or off-grid locations.

Summary

The study successfully developed an IoT-enabled smart greenhouse system designed for hydroponic gardens. By automating the monitoring and control of environmental conditions such as pH, temperature, humidity, and light, the system significantly reduces the need for manual intervention, ensuring optimal plant growth. The real-time data access provided by the cloud-based platform and mobile application enhances user convenience and system efficiency.

Despite some limitations, such as occasional sensor inaccuracies and complex wiring, the system proves to be a valuable tool for modern hydroponic farming. Future improvements, such as the integration of solar power and enhanced sensor calibration, will further increase the system's reliability and sustainability.

2.4. Summary

The three reviewed papers all center on creating and utilizing IoT-enabled systems to improve controlled environment agriculture, especially in hydroponic and vertical farming. Every article describes a system that automates the supervision and regulation of important environmental factors like pH, temperature, humidity, light, and water quality. The systems use wireless sensor networks, microcontrollers such as Arduino, and cloud-based platforms like ThingSpeak to offer mobile app access for real-time data and remote control. These advancements seek to tackle the issues faced by conventional farming, such as restricted cultivable land, erratic weather patterns, and the growing worldwide need for food. [9]

A prevalent issue found in every study is the **lack of accuracy and occasional unreliability of some sensors**, especially those that measure pH levels. During multiple experiments, sensor calibration problems resulted in inaccurate data readings, posing a risk to plant development. Furthermore, the limitations of wiring complexity and dependence on external AC power were pointed out, which hinders the scalability and sustainability of these systems in rural or off-grid regions.

Our research intends to tackle these obstacles by creating a stronger IoT system that includes enhanced sensor calibration techniques, and better wiring. Through addressing these prevalent problems, our goal is to establish a smart greenhouse system that is more dependable, environmentally friendly, and adaptable, capable of aiding both small urban farms and large-scale agricultural enterprises in various climates.

Chapter 3 Proposed System Design

For our graduation project, we need to choose a development model that aligns with our objectives and the nature of the project. After considering different models, we've narrowed it down to three options: **Waterfall**, **Agile**, and **Incremental**.

Waterfall Model

The waterfall model is a linear and sequential approach where each phase (like planning, design, and implementation) is completed before moving to the next. This model works well when the requirements are clear and fixed from the start. However, since our project might require adjustments and feedback during development, the waterfall model may not be flexible enough for our needs.

Agile Model

Agile focuses on flexibility and iterative development. It allows us to build the project in smaller cycles (sprints), test it, and improve based on feedback. This is a great approach for projects like ours, where new ideas or changes might come up during development. However, it requires proper planning and constant communication to keep things on track.

Incremental Model

The incremental model combines the benefits of both waterfall and agile. We can divide the project into smaller modules, like sensors, actuators, and user interface, and develop them one at a time. This lets us deliver and test parts of the system early, reducing risks and ensuring steady progress.

Chosen Model

For our project, we'll use an **incremental model with agile practices**. This approach allows us to focus on one module at a time while staying flexible enough to adapt based on feedback from Dr. Ahmed Ayoub. It's the best fit for our project because it balances structure with adaptability, helping us manage risks and meet our goals efficiently.

3.1. Software Development Life Cycle Framework for HydroFarmIoT

1. Planning/Requirement Analysis

- **Goal**: Define the scope, objectives, and feasibility of the software system.
- **Tasks**:
    1. Identify the critical functionalities required for the hydroponic system (e.g., sensor data acquisition, actuator control, IoT integration).
    2. Analyse constraints such as budget, hardware capabilities, and timelines.
    3. Document the functional and non-functional requirements.

2. Requirements Definition

- **Goal**: Formalize the software requirements in a clear and detailed manner.
- **Tasks**:
    1. Specify the data flow between sensors, actuators, and the microcontroller.
    2. Define IoT requirements (e.g., data transmission intervals, cloud platform compatibility).
    3. Establish thresholds and control algorithms for automated adjustments.
    4. Ensure modularity for adding future components.

3. Design

- **Goal**: Create a blueprint for the software system, focusing on architecture and module interaction.
- **Tasks**:
    1. **System Architecture**:
        - Design how the software interacts with hardware (sensors, actuators, microcontroller) and external systems (IoT platform, user interface).
    2. **Flowchart**:
        - Develop flowcharts for sensor data processing, actuator control, and alert systems.

3. **Diagrams**
    - Use case, sequence, and activity diagrams to visualize system interactions.
4. **Database Design**:
    - Plan the data storage structure for logged data on the cloud platform.

4. Implementation

- **Goal**: Develop the software components and integrate them into the hardware system.
- **Tasks**:
    1. Write code for sensor data acquisition and actuator control using the Arduino IDE.
    2. Develop IoT integration using Wi-Fi modules ESP32 for cloud connectivity.
    3. Implement the logic for automated threshold-based responses.

5. Testing

- **Goal**: Verify that the software meets the defined requirements and functions as expected.
- **Tasks**:
    1. **Unit Testing**:
        - Test individual modules, such as pH data acquisition or water pump activation.
    2. **Integration Testing**:
        - Ensure smooth interaction between sensors, actuators, and IoT components.
    3. **System Testing**:
        - Test the entire system in a simulated hydroponic environment.

6. Deployment

- **Goal**: Install the system in the actual hydroponic setup for operation.
- **Tasks**:
  1. Load the finalized code onto the microcontroller.
  2. Ensure proper connections between sensors, actuators, and the microcontroller.
  3. Set up the IoT platform for remote monitoring and control.
  4. Provide user instructions for system operation.

7. Maintenance

- **Goal**: Ensure the software continues to function reliably and efficiently after deployment.
- **Tasks**:
  1. Monitor system performance and fix bugs as needed.
  2. Update the software to add new features or improve existing functionality.
  3. Regularly calibrate sensors and update thresholds for system adjustments.
  4. Ensure the IoT platform remains operational and secure.

3.2. Requirement Analysis

Class Diagram

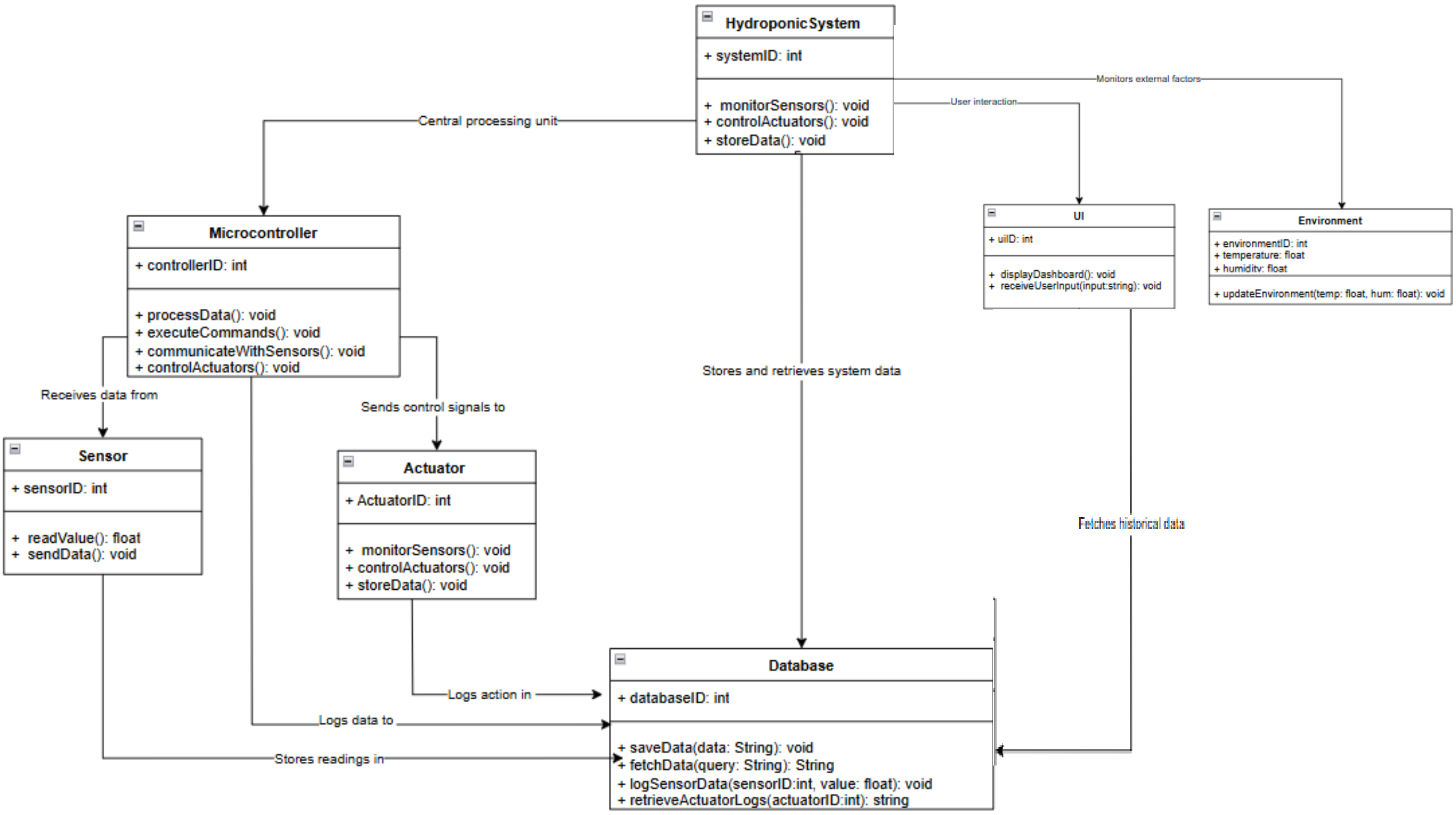

Figure 3-1: Class diagram of Hydroponic system.

This class diagram shows the main parts of the **Automatic Hydroponic System** and how they connect. Here's a breakdown of each class and what they do.

1. HydroponicSystem

This is the core of the system. It ties everything together and keeps everything running smoothly.

- **Attributes:**
  - systemId: A unique ID for the system.
- **Methods:**
  - monitorSensors(): Checks the data coming from the sensors.
  - controlActuators(): Sends commands to the actuators.
  - storeData(): Saves important data in the database.

Relationships:

- Directly connected to:

    - **Microcontroller**: Does all the processing and communication with hardware.
    - **Database**: Saves and retrieves system data.
    - **UI**: Allows the user to interact with the system.
    - **Environment**: Tracks things like temperature.

2. Microcontroller

The microcontroller is like the brain of the system. It handles sensor data and makes decisions for the actuators.

- **Attributes:**
    - controllerId: A unique ID for the microcontroller.
- **Methods:**
    - processData(): Handles data from the sensors.
    - executeCommands(): Sends instructions to the actuators.
    - communicateWithSensors(): Talks to the sensors and gathers data.
    - controlActuators(): Controls the actuators based on the data.

Relationships:

- Connected to:
    - **Sensor**: Gets data from the sensors.
    - **Actuator**: Tells the actuators what to do.
    - **Database**: Logs all the data it processes.

3. Database

The database is where all the data is stored and managed. It keeps track of everything the system does.

- **Attributes:**
    - databaseId: A unique ID for the database.
- **Methods:**
    - saveData(data: String): Saves data from the system.
    - fetchData(query: String): Retrieves data when needed.
    - logSensorData(sensorId: int, value: float): Logs the readings from the sensors.
    - retrieveActuatorLogs(actuatorId: int): Gets logs of what the actuators have done.

Relationships:

- Works with:
  - **HydroponicSystem**: Stores and retrieves system data.
  - **Microcontroller**: Saves processed data from sensors and actuators.

4. UI (User Interface)

The UI is how the user interacts with the system. It shows what's happening and lets users control things.

- **Attributes:**
  - uiId: A unique ID for the user interface.
- **Methods:**
  - displayDashboard(): Shows the current system status and data.
  - receiveUserInput(input: String): Takes commands from the user.

Relationships:

- Connected to:
  - **HydroponicSystem**: Displays system data and takes user commands.
  - **Database**: Pulls stored data for display.

5. Environment

This class represents the external conditions that the system monitors, like temperature and humidity.

- **Attributes:**
  - environmentId: A unique ID for the environment.
  - temperature: The current temperature.
- **Methods:**
  - updateEnvironment(temp: float, hum: float): Updates the temperature.

Relationships:

- Linked to:
  - **HydroponicSystem**: Provides real-time data about the external environment.

Detailed block diagram

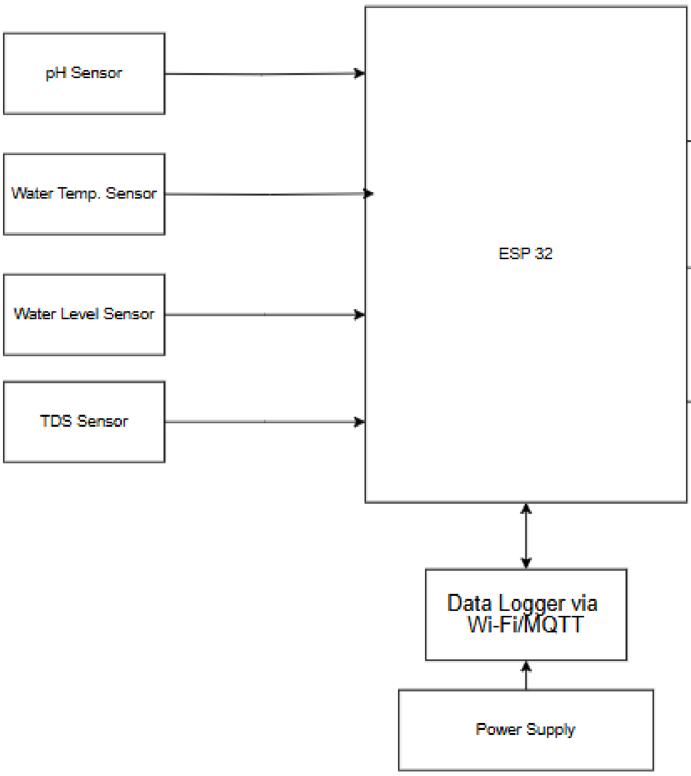

Figure 3-2: Block Diagram for IoT and Actuators of Hydroponic system

The system's functionality is represented by the block diagram below in **Figure 3.1**:

1. **Sensors**
    - Various sensors are used to monitor critical environmental and nutrient conditions:
        - **pH Sensor**: Tracks the acidity or alkalinity of the solution.
        - **Water Temperature Sensor**: Monitors the water's temperature for optimal plant growth.
        - **Water Level Sensor**: Detects the water level in the reservoir tank to prevent it from running dry.
        - **TDS Sensor**: Measure the nutrient concentration in water.
2. **Microcontroller**
    - Acts as the central processing unit of the system. It collects data from all sensors and makes decisions based on predefined conditions.
    - Controls solenoid valves through relay modules to regulate pH levels.
    - Publishes sensor readings over Wi-Fi using the MQTT protocol for remote data logging.

3. **Actuators**
    - **Solenoid Valve (Acidic) + Relay Driver**: Dispenses acidic solution into the water tank when the pH level is too high.
    - **Solenoid Valve (Alkaline) + Relay Driver**: Dispenses alkaline solution into the water tank when the pH level is too low.
4. **Data Logger (via Wi-Fi/MQTT)**
    - Transmits sensor data to an external logging system or database using MQTT over Wi-Fi
    - Enables historical data tracking for analysis and troubleshooting.
5. **Node-RED Dashboard**
    - A graphical interface built using Node-RED displays real-time sensor values and system status.
    - Allows remote user access and system interaction via a web browser.
6. **Power Supply**
    - Provides the necessary energy to operate the microcontroller, sensors, and actuators.

Summary:

This system integrates essential environmental and nutrient sensors with an ESP32 microcontroller to automate hydroponic water quality management. It controls pH levels using solenoid valves driven by relays and monitors nutrient strength with a TDS sensor. Real-time data is logged wirelessly via MQTT and visualized through a Node-RED dashboard, providing a smart, automated, and remotely accessible hydroponic farming solution.

Context Diagram

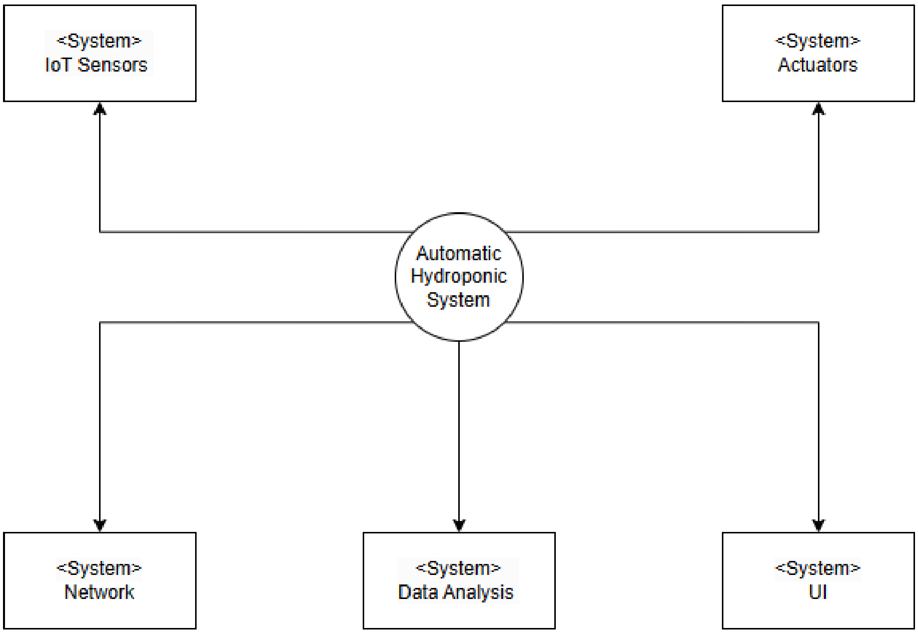

Figure 3-3: context diagram shows the structure of an Automatic Hydroponic System

This context diagram in **Figure 3.2** shows the structure of an **Automatic Hydroponic System** and how its different parts work together to create an efficient hydroponic farming setup. The central system connects and controls all the components, ensuring they communicate and function properly.

1. **IoT Sensors**
   - These sensors measure important factors like pH, temperature, tds, and water level in the hydroponic system. They send real-time data to the main system for monitoring.
2. **Actuators**
   - Actuators are devices that adjust things like pH level. Based on the sensor data, the system tells the actuators what to do to maintain the best conditions for plant growth.
3. **Network**
   - The network ensures that the sensors, actuators, and the main system are all connected and can share information without interruptions.

4. **Data Analysis**
    - This part analyzes the data from the sensors and figures out what actions need to be taken. For example, if the pH is too high, it will recommend adjustments to fix it automatically.
5. **User Interface (UI)**
    - The UI allows users to see the system's data and control it remotely. It shows things like sensor readings and system performance while also letting users make changes if needed.

How It All Works:

The **Automatic Hydroponic System** uses these components to create a smart, automated farming setup. The sensors collect data, the system analyses it, and the actuators make the necessary adjustments. The network keeps everything connected, and the UI makes it easy for users to interact with the system. This setup helps reduce manual work, improve efficiency, and ensures plants grow in the best conditions possible year-round.

Workflow Diagram

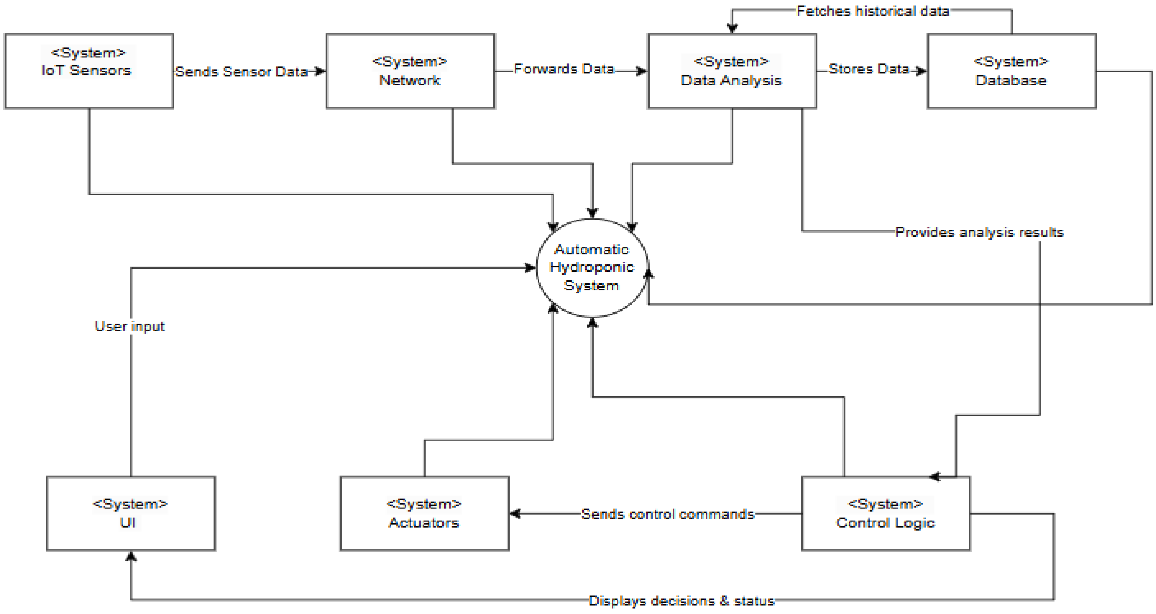

Figure 3-4: Workflow and components of an Automatic Hydroponic System.

This diagram in **Figure 3.3** illustrates the workflow and components of an **Automatic Hydroponic System**. Each component has a specific role, and they all interact to ensure efficient monitoring and control of the hydroponic environment. Here's how the system works:

1. **IoT Sensors**
   - These sensors collect real-time data on critical parameters like pH, temperature, and nutrient levels. The data is sent to the network for processing.
2. **Network**
   - The network transmits sensor data to other components of the system. It ensures seamless communication between the sensors, analysis unit, and control logic.
3. **Data Analysis**
   - This component analyses the incoming data to identify trends or irregularities. It uses real-time data and historical information fetched from the database. The results are forwarded to the control logic and the main system.

4. **Database**
    - The database stores historical data collected by the sensors. It allows the system to compare current conditions with past data to make informed decisions.
5. **Control Logic**
    - This is the decision-making unit. It processes the analysis results and determines the necessary adjustments to maintain optimal growing conditions. Commands are then sent to the actuators.
6. **Actuators**
    - Based on instructions from the control logic, the actuators adjust environmental factors like pH level.
7. **User Interface (UI)**
    - The UI provides users with a way to interact with the system. Users can input commands, monitor system status, and view the decisions made by the control logic.

Workflow:

The **Automatic Hydroponic System** begins with sensors gathering data, which is sent via the network for analysis. The data analysis component uses historical records from the database to provide insights. Control logic then acts on these insights, sending commands to actuators to optimize the system. Users can monitor and manage the entire process through the UI. This setup ensures efficient and automated management of the hydroponic environment.

Activity Diagram

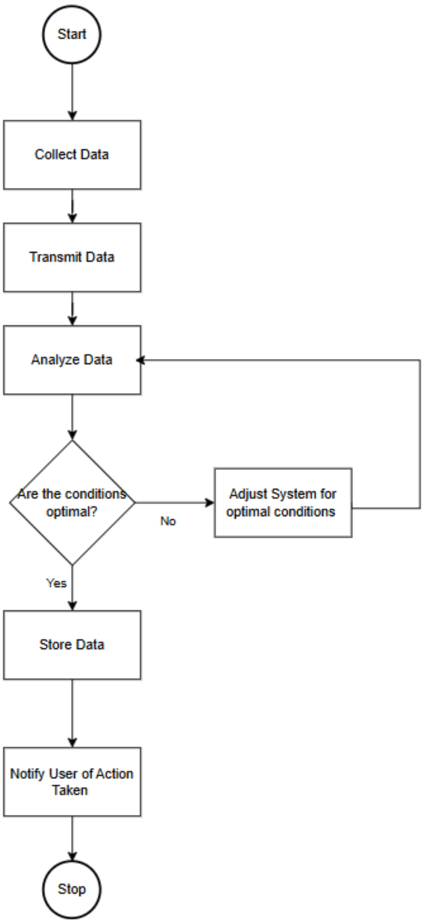

Figure 3-5: Activity Diagram of an Automatic Hydroponic System.

The **Automatic Hydroponic System** is designed to automate the monitoring and management of essential parameters required for optimal plant growth in a hydroponic setup. The system integrates sensors, actuators, and IoT features to ensure that the hydroponic environment remains within desired conditions.

Key Functions

1. **Data Collection**:
    - Sensors continuously monitor parameters such as pH, nutrient levels, water temperature, and water level.
    - Data is captured in real-time to ensure accurate decision-making.

2. **Data Transmission**:
    - Collected data is transmitted to a microcontroller or central processing unit for further analysis.
    - Wireless communication can be utilized to send data to a cloud server for remote monitoring.
3. **Data Analysis**:
    - The system compares the collected data against predefined optimal thresholds for each parameter.
    - A decision is made whether the conditions are optimal or require adjustments.
4. **Condition Adjustment**:
    - If conditions deviate from optimal ranges, the system activates the necessary actuators:
        - **Solenoid Valve (Acidic) + Relay Driver**: Dispenses acidic solution into the water tank when the pH level is too high.
        - **Solenoid Valve (Alkaline) + Relay Driver**: Dispenses alkaline solution into the water tank when the pH level is too low.
    - The system loops back to re-analyse the updated conditions.
5. **Data Storage and Notification**:
    - Data is stored locally or in the cloud for tracking system performance and historical analysis.
    - The user is notified of the system's status, including any adjustments made or alerts triggered.

Workflow Description

1. The system starts by **collecting data** from various sensors.
2. The data is **transmitted** to the processing unit for analysis.
3. The system **analyses** the data to check whether conditions are within optimal ranges.
    - If **optimal**, the system **stores the data** and **notifies the user** of the status.
    - If **not optimal**, the system **adjusts parameters** by activating relevant actuators.

4. After adjustments, the system loops back to re-analyse the conditions, ensuring a continuous feedback cycle.

Benefits of the System

- Ensures optimal growing conditions for plants, improving growth rates and yields.
- Reduces manual labour by automating monitoring and adjustments.
- Provides real-time data and notifications, enhancing user control and decision-making.

Enables historical data tracking for better understanding and optimization of the system over time.

Use Case Diagram

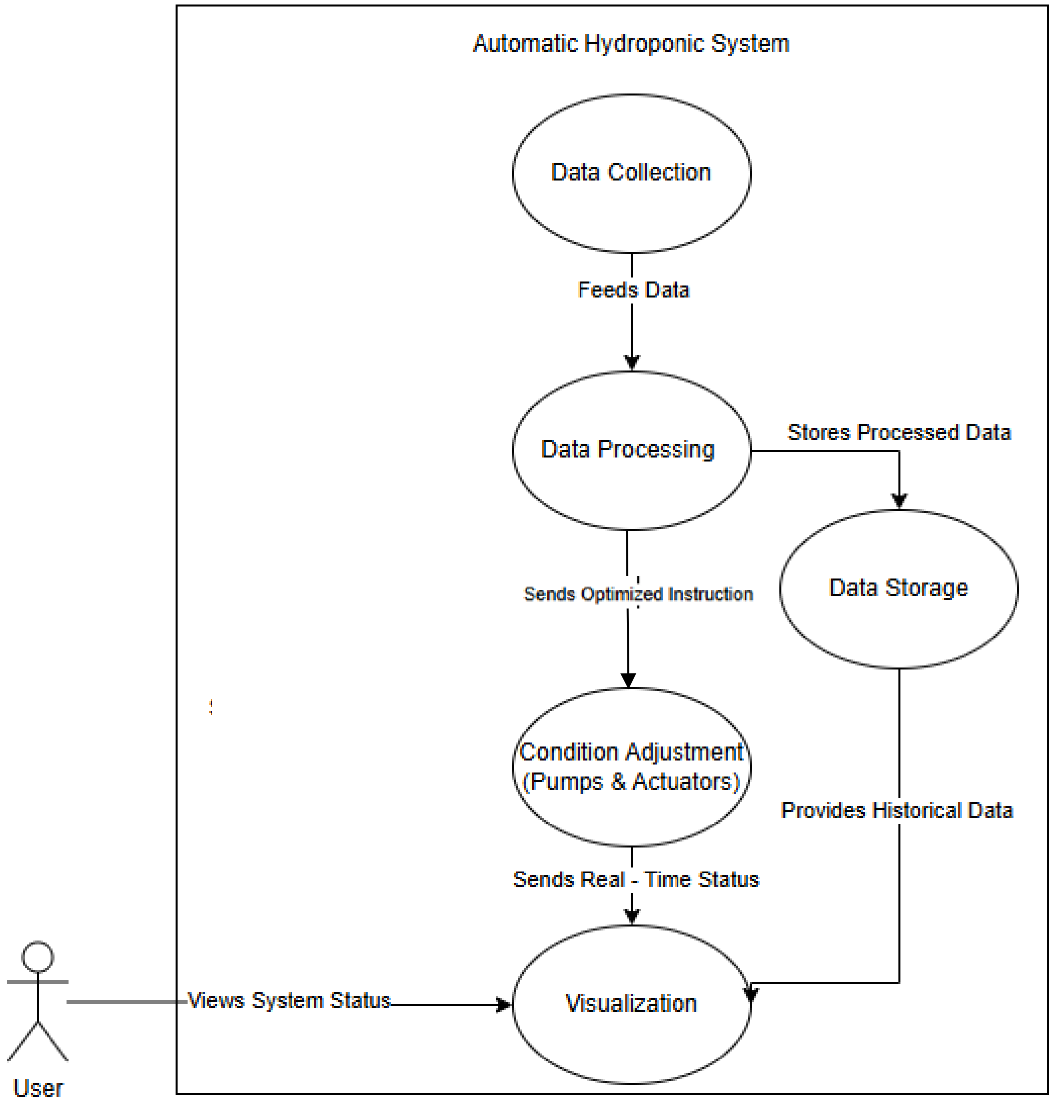

Figure 3-6: Use Case Diagram for the Automatic Hydroponic System.

The **Use Case Diagram** for the Automatic Hydroponic System illustrates the system's primary functionalities, their interactions, and how users engage with the system. It focuses on ensuring a clear and logical representation of the workflow, emphasizing the integration of AI to enhance decision-making and system efficiency.

Actors

- **User**: The sole external actor interacting with the system. The user can monitor the system's status and view data visualizations through a dedicated interface. This keeps the diagram clean and adheres to the principle of minimizing actor complexity.

System Overview

The system is represented as a rectangle containing five interconnected use cases:

1. **Data Collection**:
    - The starting point of the system’s workflow.
    - Sensors gather environmental data such as pH, temperature, nutrient levels, and dissolved oxygen.
    - Data flows to the next stage for analysis.
2. **Data Processing**:
    - Raw sensor data is processed and organized for meaningful analysis.
    - Outputs processed data to both the conditioned adjustment and the data storage unit.
3. **Condition Adjustment (using actuators)**:
    - Actuators such as pumps and valves are activated based on Node- RED generated instructions.
    - Maintains optimal growing conditions for plants.
    - Feeds real-time status updates to the visualization module.
4. **Visualization**:
    - Presents processed data, real-time system status, and historical trends to the user.
    - Enables the user to monitor and understand the system's performance effectively.

Key Interactions

- **User to Visualization**:
    - The user interacts with the system exclusively through the visualization module to view system status and data insights.
- **Data Collection to Data Processing**:
    - Sensor data is forwarded for cleaning, filtering, and structuring.

- **Data Processing to Data Storage and Visualization**:
    - Historical data is stored and made available for user access and monitoring.

Benefits of the Use Case Diagram Design

1. **Clarity**:
    - Simplifies interactions by focusing on core functionalities and a single user actor.
2. **Modularity**:
    - Shows logical connections between use cases, making the workflow easy to understand.
3. **User-Friendly**:
    - The visualization module ensures the user can access comprehensive insights without direct interaction with the system's technical components.

This diagram effectively balances simplicity and functionality, presenting a clear overview of the Automatic Hydroponic System.

3.3. Functional and Non-Functional Requirement

Functional Requirements

1. **Monitoring Capabilities**:
    - Monitor pH, nutrient levels, water temperature, and water level using respective sensors.
2. **Automated Control**:
    - Adjust pH level using solenoid valves and the water pump using smart timer.
3. **Data Visualization**:
    - Display real-time sensor readings on node RED Dashboard

4. **Data Logging**:
    - Sensor readings are transmitted via MQTT and logged through the Node-RED dashboard.
5. **IoT Integration**:
    - Send data to the cloud for remote monitoring and control via a mobile or web interface.
6. **Alert System**:
    - Trigger visual or audible alarms when parameters deviate from defined thresholds.

Non-Functional Requirements

1. **Reliability**:
    - Ensure continuous operation with minimal downtime.
2. **Scalability**:
    - Allow for adding more sensors or actuators as needed.
3. **Power Efficiency**:
    - Optimize power usage to support extended operation, especially in resource-constrained settings.
4. **User Interface**:
    - Ensure the interface is intuitive and easy to understand, with clear dashboard displays and alerts.
5. **Environmental Resistance**:
    - Components should withstand high humidity and water exposure inherent in hydroponic systems.
6. **Data Security**:
    - Protect transmitted and stored data from unauthorized access or tampering.

3.4. System Specifications

1. Hardware Requirements

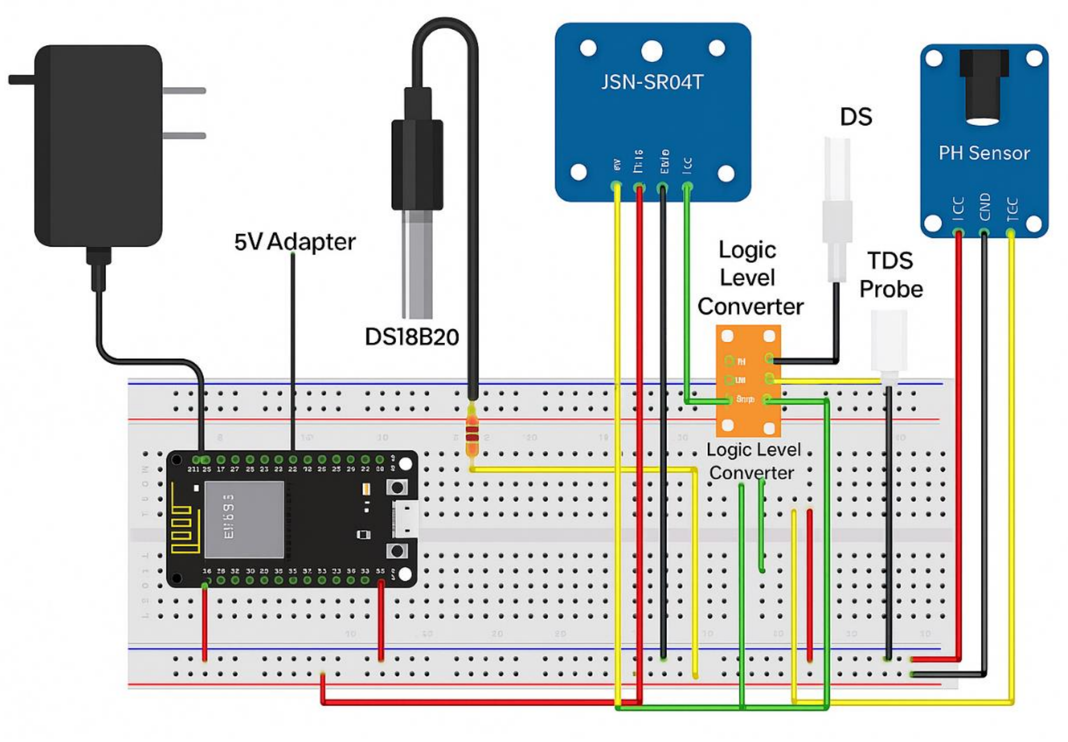

Figure 3-7: Basic Hydroponic Schematic

Schematic Explanation:

This schematic illustrates the full prototype setup of the Automatic Hydroponic System using an ESP32 microcontroller, multiple analog/digital sensors, and a regulated 5V power supply. The design prioritizes real-time environmental monitoring with reliable signal conditioning, accurate readings, and safe logic-level communication.

1. ESP32 Microcontroller

- Serves as the central controller responsible for reading sensor data and transmitting it over Wi-Fi to the Node-RED dashboard.
- Powered via the 5V adapter connected to the breadboard's positive rail.
- Receives signals from multiple sensors through analog and digital GPIOs.

2. Power Supply

- A regulated 5V adapter is used to power the entire breadboard setup.
- Power is distributed through the red (+) and blue (−) rails for clean and consistent voltage delivery.
- Components that require 3.3V (like the logic level converter and ESP32) are managed accordingly.

3. DS18B20 – Water Temperature Sensor

- A waterproof digital sensor that communicates over the 1-Wire protocol.
- Connected to GPIO pin (e.g., GPIO27) of the ESP32.
- A 4.7kΩ pull-up resistor is placed between the data line and VCC (3.3V) to ensure stable operation.
- Used to measure the nutrient solution's temperature, which also influences pH sensor accuracy.

4. pH Sensor Module

- Includes a BNC probe connected to a signal-conditioning board.
- Powered via the 5V rail and connected to the ESP32 through an analog GPIO (e.g., GPIO35).
- Measures the acidity or alkalinity of the hydroponic solution.
- Requires careful calibration with buffer solutions for accurate readings.

5. TDS Sensor (Total Dissolved Solids)

- A white TDS probe is attached to a small signal module.
- Connected to the 5V rail, with signal routed to an analog input on the ESP32 (e.g., GPIO34).
- Measures electrical conductivity to estimate nutrient concentrations in the water.
- Shares GND with other components for consistent voltage reference.

6. Ultrasonic Water Level Sensor (JSN-SR04T)

- Used to measure water level in the nutrient reservoir using ultrasonic pulses.
- Powered via the 5V rail but communicates using 5V logic signals, which are not directly compatible with the ESP32's 3.3V inputs.

7. Logic Level Converter

- A bi-directional logic level shifter is introduced to step down the 5V logic from the JSN-SR04T's Echo and Trigger pins to safe 3.3V levels for the ESP32.
- High Voltage (HV) side is powered by 5V, while the Low Voltage (LV) side is powered by 3.3V.
- Shared GND ensures stable logic conversion.
- Trigger and Echo pins from the JSN-SR04T are connected to the HV side, while corresponding LV lines are connected to GPIO pins on the ESP32 (e.g., GPIO25 for Trigger and GPIO26 for Echo).

Wiring and Electrical Notes

- All sensors share a **common GND** for consistent reference voltage.
- The **color-coded wiring** (Red = VCC, Black = GND, Yellow/Green = Signal) improves readability and troubleshooting.
- Each sensor was tested individually before integration to ensure signal clarity and avoid cross-talk, particularly between analog devices.

Summary

This schematic provides a clear overview of the sensor network architecture used in the hydroponic system. It combines multiple sensing modalities to monitor essential environmental factors while leveraging the ESP32's wireless capabilities for seamless data visualization and control. The addition of the logic level converter ensures safe integration of 5V logic components without compromising microcontroller performance or stability.

The hardware requirements include all physical components necessary to monitor, control, and automate the hydroponic system. [10]

Hardware Specification

Sensors:

1. **NPK Sensor**: Measures nutrient levels in the water.
2. **pH Sensor**: Tracks the acidity or alkalinity of the water.
3. **Water Temperature Sensor**: Monitors the water's temperature to ensure optimal plant health.
4. **Water Level Sensor**: Detects low water levels to prevent system dry-out.
5. **Dissolved Oxygen Sensor**: Ensures sufficient oxygen levels in the nutrient solution.
6. **Humidity Sensor**: Measures ambient humidity for environmental monitoring.
7. **Light Sensor**: Tracks light intensity to ensure adequate exposure for photosynthesis.

Actuators:

1. **Air Pump**: Aerate the water to maintain dissolved oxygen levels.
2. **Water Pump**: Circulates nutrient solutions in the hydroponic system.
3. **Nutrient Pumps A & B**: Dispense precise amounts of nutrients (macro and micro) into the water.
4. **Solenoid Valve**: Controls water flow in and out of the system.

Controller:

1. **Microcontroller**:
    - Arduino-based controller with sufficient GPIO pins for sensor and actuator integration.
    - Must support Wi-Fi for IoT connectivity.

Display and Storage:

1. **LCD Screen**: A 16x2 or 20x4 display for real-time data visualization.

2. **MicroSD Card Module**: For logging sensor data for analysis and troubleshooting.

Power Supply:

1. **12V DC Power Source**: Provides reliable power to all system components, including pumps and sensors.
2. **Voltage Regulator**: Ensures stable operation of sensitive components like the microcontroller and sensors.

2. Software Requirements

The software requirements cover the development tools and platforms needed to operate the system.

Development Tools:

1. **Programming Language**:
   - C++ for Arduino programming.
2. **Integrated Development Environment (IDE)**:
   - Arduino IDE for writing and uploading code to the microcontroller.
3. **IoT Platform**:
   - Blynk
4. **Data Logging**:
   - SD Card Library for microcontroller-based storage.

Algorithms:

1. Real-time monitoring and logging of sensor data.
2. Threshold-based actuator control to adjust environmental parameters automatically.
3. IoT communication for cloud data transmission and remote access.

3. Operational Requirements

These requirements ensure smooth deployment and operation of the system.

1. **Environmental Conditions**:
   - The system must function effectively in high humidity and moderate temperature conditions typical of hydroponic setups.
2. **Connectivity**:
   - Stable Wi-Fi connectivity is required for IoT functionality.
3. **Ease of Maintenance**:
   - The system must allow for easy calibration of sensors and replacement of actuators or other components.
4. **Safety**:
   - All electrical components must be insulated to prevent short circuits in a water-rich environment.
5. **Scalability**:
   - The system should support the addition of more sensors or actuators if needed.

3.5. Industry Standard

To ensure the reliability, scalability, and compatibility of the HydroFarmIoT system, several industry standards were adhered to during its design and implementation. These standards guide the system's compliance with best practices in hardware, software, communication protocols, and agricultural technology.

1. Hardware Standards

- **ISO 11783 (ISOBUS):** This standard ensures compatibility between agricultural machinery and electronics. Although primarily used in large-scale farming, its principles were adapted for seamless integration between sensors, actuators, and controllers in HydroFarmIoT.

- **IEC 61010:** Applicable to electronic measuring and control devices, this safety standard ensures that all sensors and electrical components operate safely under hydroponic conditions.

- **RoHS Compliance:** All components used in HydroFarmIoT, including the microcontroller, sensors, and actuators, meet the Restriction of Hazardous Substances Directive to minimize environmental impact.

2. Communication Standards

- **IEEE 802.11 (Wi-Fi):** Used for wireless communication, enabling IoT connectivity between the microcontroller and cloud platform.
- **MQTT Protocol:** A lightweight messaging protocol ideal for IoT applications. It ensures efficient communication between the HydroFarmIoT system and user interface while consuming minimal bandwidth.
- **HTTPS (Hypertext Transfer Protocol Secure):** Used for secure data transmission between the IoT system and cloud services, protecting user data and ensuring confidentiality.

3. Software Standards

- **ISO/IEC 12207:** This software lifecycle process standard was referenced to structure the development, deployment, and maintenance of the HydroFarmIoT software, ensuring quality and consistency.
- **OWASP IoT Security Guidelines:** These guidelines were implemented to protect the system from potential cybersecurity threats, ensuring secure operation and data management.

4. Agricultural Standards

- **ISO 22000:** Though primarily a food safety management standard, its principles were considered to design a system that supports sustainable and safe agricultural practices.
- **GlobalG.A.P. Standards:** HydroFarmIoT aligns with these standards for sustainable water management and nutrient optimization in hydroponic farming.

5. Environmental and Sustainability Standards

- **ISO 14001:** This environmental management standard was referenced to minimize the ecological footprint of the HydroFarmIoT system, particularly its water and energy consumption.
- **LEED Certification Guidelines:** While HydroFarmIoT does not directly seek certification, its vertical farming design follows the principles of resource efficiency and urban sustainability highlighted in these guidelines.

Summary

By adhering to these standards, the HydroFarmIoT system is not only robust and efficient but also compliant with the industry's best practices. This ensures it can be scaled, maintained, and adapted for future advancements while promoting sustainable and innovative agricultural solutions.

Chapter 4 Implementation

This chapter provides a detailed explanation of the implementation of the proposed Automatic Hydroponic System, covering hardware integration, software development, and overall system integration.

4.1 Hardware Implementation

The hardware implementation of the system consists of various components that work together to automate and monitor the hydroponic environment. The main components include:

4.1.1 Microcontroller (ESP32)

The **ESP32** microcontroller is the core processing unit of the system, responsible for reading sensor values, controlling actuators, and communicating with the dashboard and database. The ESP32 was chosen due to its built-in Wi-Fi capability, low power consumption, and sufficient GPIO pins for sensor and actuator connections.

4.1.2 Sensors

To monitor the critical parameters of the hydroponic environment, the following sensors were integrated into the system. Each sensor is connected to the ESP32, which reads data periodically and transmits it for real-time monitoring:

Sensor	Purpose	Model/Type	Operating Voltage	Interface	Range	Accuracy
pH Sensor	Measures the acidity/alkalinity of the solution	Analog pH Sensor Kit	5V	Analog	0 – 14 pH	±0.1 pH (after calibration)
Water Temp Sensor	Monitors water temperature	DS18B20 (Waterproof)	3.0 – 5.5V	Digital (1-Wire)	-55°C to +125°C	±0.5°C

Sensor	Purpose	Model/Type	Operating Voltage	Interface	Range	Accuracy
TDS Sensor	Measures total dissolved solids (nutrient levels)	TDS Sensor V1.0	3.3 – 5V	Analog	0 – 1000 ppm	±10%
Water Level Sensor	Detects the water level in the tank	JSN-SR04T Ultrasonic	5V	Digital (PWM)	20 – 600 cm (±3 mm error)	±3 mm
Relay Module	Controls high-voltage components (Valves)	2-Channel 5V Relay Module	5V	Digital (IN1/IN2)	10A @ 250VAC / 15A @ 125VAC	Opto-isolated, logic-level control
LCD Display	Displays real-time readings	16x2 LCD with I2C Module	5V	I2C	2 Rows × 16 Characters	I2C address configurable

Table 4.1: Details the key hardware components used, including their functions, models, voltage requirements, interfaces, and specifications.

These sensors enable continuous real-time monitoring of water quality and system conditions. Issues such as low nutrient levels (e.g., early readings of ~200 ppm) were detected and corrected using these sensor readings, which were crucial in maintaining the proper growth environment.

4.1.3 Actuators

The system automates nutrients and water flow using the following actuators:

- **Water Pump:** Circulates water throughout the hydroponic setup.
- **Solenoid Valves (Acidic & Alkaline solution):** Controls solution flow automation for optimal pH level.

The actuators are triggered based on predefined conditions set within the system's control logic.

4.1.4 Power Supply

The system operates on a **12V/5V power supply**, with voltage regulators ensuring each component receives the appropriate power level.

4.2 Software Implementation

The software implementation consists of embedded programming for the ESP32, dashboard development using Node-RED, and database setup for data logging.

4.2.1 Embedded Programming (ESP32 & Arduino IDE)

The ESP32 is programmed using **Arduino IDE** with C++ code to:

- Read sensor values at regular intervals.
- Process sensor data and determine actuator responses.
- Communicate with Node-RED via Wi-Fi using MQTT or HTTP protocols.

4.2.2 Node-RED Dashboard

A **custom dashboard** was developed using **Node-RED** to:

- Display real-time sensor readings.
- Provide manual control options for actuators.
- Log historical data for performance analysis.

The dashboard is designed to be intuitive, providing a clear visualization of the hydroponic system’s status.

4.2.3 Database Integration

To store and analyze sensor data over time, InfluxDB is used. The database is responsible for:

- Logging sensor readings at predefined intervals.
- Storing actuator activity logs for troubleshooting and optimization.
- Allowing retrieval of historical data for long-term analysis.

4.3 System Integration

The final step in implementation is ensuring that all hardware and software components work seamlessly together. The integration follows this workflow:

1. **Sensors collect environmental data** and send readings to the ESP32.
2. **The ESP32 processes the data** and determines if any actuator needs to be triggered.
3. **The ESP32 sends control signals** to the solenoid valve.
4. **The Node-RED dashboard receives live data** from the ESP32, displaying it in real time.
5. **The database logs sensor and actuator data** for long-term monitoring.
6. **Users can manually adjust settings** via the Node-RED dashboard.

This architecture ensures a robust and automated hydroponic system that optimizes water and nutrient delivery based on real-time conditions.

4.4 Summary

This chapter detailed the hardware and software implementation of the Automatic Hydroponic System. The system is designed to efficiently monitor and control nutrient delivery using ESP32**-based architecture**, **various sensors and actuators**, and **a real-time dashboard**. System integration ensures that data flows smoothly between hardware components and the user interface, creating an automated and intelligent hydroponic farming solution. Future work includes full database implementation and final hardware testing for complete system validation.

Chapter 5 Testing

5.1 Testing Setup

The testing environment for the proposed Automatic Hydroponic System was carefully prepared to evaluate the functionality of both the hardware and software components. The hardware setup consisted of an ESP32 microcontroller, a selection of environmental sensors (pH, Total Dissolved Solids (TDS), water temperature, and water level sensors), and actuators (a submersible water pump, and two solenoid valves labeled Acidic and Alkaline used for pH regulation).

All components were connected using a breadboard and jumper wires for prototyping. A stable power supply was used to ensure reliable operation during the testing phase. The sensors were calibrated individually prior to testing to improve accuracy.

The ESP32 was programmed using the Arduino IDE, and communication between the microcontroller and the control interface was established over Wi-Fi. The system interfaced with a custom-designed Node-RED dashboard, which provided real-time data visualization and control functionality. The dashboard allowed users to monitor sensor values and manually or automatically control actuators based on predefined thresholds.

Additionally, the system was linked to a local database, to store sensor readings and log system events for analysis and traceability. Testing was performed in a controlled indoor environment designed to simulate typical conditions found in a small-scale hydroponic farming system.

5.2 Testing Separate Hardware and Software Components

The testing of the Automatic Hydroponic System was conducted at various stages to ensure that both hardware and software components met the requirements and functioned as expected. This process was divided into three primary categories: Unit Testing, Integration Testing and ,System Testing.

5.2.1 Unit Testing

Unit testing focused on validating the functionality of each hardware and software component in isolation:

- **pH Sensor**: Tested with standard buffer solutions (pH 4.0, 7.0, and 10.0) to verify accuracy and consistency of readings.
- **TDS Sensor**: Calibrated using water samples with known nutrient concentrations to confirm sensor reliability.
- **Water Temperature Sensor**: Submerged in water at varying temperatures to check real-time data accuracy.
- **Water Level Sensor**: Tested by gradually filling a container and observing sensor responses at different levels.
- **ESP32 Microcontroller**: Independently programmed and tested for stable sensor readings, data transmission, and response to control signals.
- **Node-RED Dashboard**: Each UI component (gauge, chart, switch) was tested to ensure it correctly reflected data and sent control commands.
- **LCD Display**: Ensure that the data appearing on the lcd was accurate and in synch with the dashboard.
- **Solenoid Valve:** Ensure that the valves are activated upon command from the esp32 via the relay module.

5.2.2 Integration Testing

After confirming the functionality of individual components, they were gradually integrated and tested together. The goal was to ensure proper communication between the ESP32, sensors, actuators, and the Node-RED interface:

- Sensor data was successfully transmitted from ESP32 to Node-RED via Wi-Fi and displayed in real time.
- Control commands from Node-RED triggered the appropriate actuators (e.g., Solenoid ON/OFF).
- Logical operations were tested, such as activating a nutrient pump when pH dropped below a threshold.

- Multiple sensor readings were sent simultaneously to verify synchronization and stability.

5.2.3 System Testing

System testing was conducted with the complete setup running under simulated hydroponic conditions. The system was observed for several hours, and the following aspects were tested:

- Automatic and manual control of actuators from the dashboard.
- Real-time updates and responsiveness of all sensors.
- Stability of data logging and Wi-Fi connection.
- System behavior under varying environmental conditions (e.g., changes in light or water level).

These tests validated the performance of the entire system as a cohesive unit, ensuring it could perform as intended in a real-world environment.

5.2.5 Test Cases

1. ESP32 Functionality Test

- **Test Case ID**: TC001
- **Description**: Verify that the ESP32 successfully powers on, runs code, and communicates with the serial monitor.
- **Preconditions**: ESP32 is connected via USB and properly configured in the IDE.
- **Test Steps**:
    1. Connect the ESP32 to the computer.
    2. Upload a simple sketch (e.g., print a message or blink an LED)
    3. Open the Serial Monitor to observe the output.
- **Expected Result**: Serial output is visible or the LED blinks, confirming the ESP32 is functioning correctly.

2. Actuator Operation

- **Test Case ID**: TC002
- **Description**: Verify actuators respond to commands from the controller.
- **Preconditions**: Actuators are connected and idle.
- **Test Steps**:
    1. Issue a command to the actuator (e.g., turn on solenoid valve).
    2. Observe the actuator's behaviour.
- **Expected Result**: Actuator performs the action correctly (e.g. Valve adjusts).

3. Data Logging and Retrieval

- **Test Case ID**: TC003
- **Description**: Verify that data is logged and can be retrieved correctly.
- **Preconditions**: System is running with sensors and actuators active.
- **Test Steps**:
    1. Allow the system to run for a predefined duration.
    2. Access the data log for the specific timeframe.
    3. Compare retrieved data with real-time observations.
- **Expected Result**: Logged data matches observed sensor and actuator values.

4. User Interface Functionality

- **Test Case ID**: TC005
- **Description**: Verify that the user interface displays real-time and historical data correctly.
- **Preconditions**: System is operational, and data is logged.
- **Test Steps**:
    1. Access the real-time monitoring page.
    2. Navigate to historical data visualization.
- **Expected Result**: Real-time data matches live conditions, and historical data matches logged values.

5. Data Feedback Loop

- **Test Case ID**: TC006
- **Description**: Verify that the data feedback loop adjusts sensor readings accurately.
- **Preconditions**: Sensors and data logger are active.
- **Test Steps**:
    1. Simulate abnormal readings.
    2. Observe the system's response and adjustments via the feedback loop.
- **Expected Result**: Sensor readings stabilize based on feedback from the data logger.

6. System Initialization

- **Test Case ID**: TC007
- **Description**: Verify that the system initializes correctly.
- **Preconditions**: System components are connected and powered off.
- **Test Steps**:
    1. Power on the system.
    2. Observe startup sequence and initial states of sensors, actuators, and user interface.
- **Expected Result**: System initializes without errors, and all components are in a ready state.

7. TDS Sensor Verification

- **Test Case ID:** TC07
- **Description:** Confirm accuracy and transmission of TDS sensor readings.
- **Preconditions:** TDS sensor is connected, and system is operational.
- **Test Steps:**
    1. Place TDS sensor in a solution with known ppm (e.g., 500 ppm).
    2. Check Node-RED dashboard reading.
- **Expected Result:** Dashboard displays ~500 ppm with small variance.

8. Water Temperature Sensor Verification

- **Test Case ID:** TC08
- **Description:** Validate DS18B20 temperature sensor readings.
- **Preconditions:** Sensor connected to digital pin; system powered.
- **Test Steps:**
    1. Place sensor in water at known temperature (e.g., 25°C).
    2. Compare dashboard reading to thermometer.
- **Expected Result:** Dashboard reading is within ±0.5°C of actual.

9. Ultrasonic Water Level Sensor Test

- **Test Case ID:** TC09
- **Description:** Verify water level readings using ultrasonic sensor.
- **Preconditions:** Sensor mounted at top of water tank.
- **Test Steps:**
    1. Fill tank to different levels (e.g., 10 cm, 20 cm, 30 cm).
    2. Observe Node-RED output.
- **Expected Result:** Displayed distance matches expected level within ±3 mm.

9. LCD Display Functionality

- **Test Case ID:** TC010
- **Description:** Validate that the LCD display shows real-time sensor data.
- **Preconditions:** LCD is connected to the ESP32 via l3C interface.
- **Test Steps:**
    1. Power on the system.
    2. Observe the LCD screen.
    3. Compare displayed values with the actual sensor reading.
- **Expected Result:** LCD Display updated sensor data clearly.

Below is a sample test case for verifying the pH sensor integration:

Project Name: Automatic Hydroponic System

Test Case Template

Test Case	**TC011**
Test Priority	High
Module Name	pH Sensor
Test Title	Verify pH sensor readings and transmission to Node-RED
Description	Validate the accuracy of pH sensor and real-time update on dashboard
Pre-conditions	ESP32 is powered and connected to Node-RED
Dependencies	pH sensor connected to analog input pin

Step	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Notes**
1	Insert pH 4 buffer	pH 4	Dashboard shows ~4	4.1	Pass	Slight offset
2	Insert pH 7 buffer	pH 7	Dashboard shows ~7	7.0	Pass	Accurate
3	Insert pH 10 buffer	pH 10	Dashboard shows ~10	10.1	Pass	Stable reading

Table 5.1:Sample test case for verifying pH sensor.

Post-conditions: pH sensor is validated, values correctly displayed and logged.

5.2.6 Issues faced when setting up the sensors.

During the sensor setup and initial testing phase of the HydroFarmIoT system, several challenges were encountered that affected data accuracy and overall system performance. These issues were gradually addressed through troubleshooting, hardware adjustments, and calibration.

1. Sensor Calibration and Stability:

Many of the sensors, including the pH sensor and TDS (Total Dissolved Solids) sensor, initially provided unstable or inconsistent readings. This was primarily due to:

- Incomplete calibration of the sensors.
- Electrical noise from shared power lines on the breadboard.
- Variability in water quality and sensor placement.

To address these issues, sensors were recalibrated multiple times, and connections were reinforced. The pH sensor especially required extra attention, including shielding from electromagnetic interference and isolating its analog input from other components.

2. Incorrect Nutrient Concentration Detection:

In the early stages of testing, the TDS sensor reported a nutrient concentration of around **200 ppm**, which was significantly below the recommended level of **700 ppm** for most leafy greens. This misreading led to an undernourished hydroponic solution and prompted concern over sensor accuracy.

Upon further inspection, the low reading was traced back to two main causes:

- The actual nutrient solution ratio was incorrect, leading to low concentration in the water sample.
- The sensor required recalibration to match the target ppm range accurately.

After adjusting the nutrient dosing and recalibrating the TDS sensor, the readings stabilized, and nutrient levels reached the desired 700 ppm range.

This highlighted the importance of both physical nutrient mixing and reliable sensor calibration.

3. Waterproofing and Placement Issues:
Some sensors, especially the ultrasonic water level sensor and the DIY pH probe, required careful positioning. The JSN-SR04T ultrasonic sensor, although waterproof, occasionally produced incorrect readings if angled improperly or placed too close to water splashes. The pH probe had to be immersed at the correct depth and isolated from water turbulence.

4. Logic Level Compatibility and Power Concerns:
Certain components operated at different logic levels (e.g., 5V sensors with a 3.3V ESP32 microcontroller), which led to unstable data transfer. A logic level converter was added to bridge this gap. In addition, ensuring that all sensors received consistent and regulated power, especially when using external power adapters was necessary to prevent fluctuations in sensor outputs.

These early challenges were crucial learning steps that improved the robustness of the system and ensured that later data collection and automation logic were based on accurate and reliable readings.

5.3 Results and Observations

The results of the testing phase provided clear evidence that the HydroFarmIoT system functioned as expected. Real-time monitoring, sensor accuracy, actuator control, and the overall user interface were validated through both controlled tests and live system behaviour. The following points present visual outcomes of the implementation, including the Node-RED dashboard, the physical sensor connections, and the plant growth over time.

1. Node-RED Dashboard Screenshot

A custom Node-RED dashboard was developed to serve as the system’s primary monitoring and control interface. It displays real-time sensor values for pH, TDS, water temperature, and water level.

The dashboard also includes gauges and historical charts to help users visualize environmental trends and system responses over time.

The image below shows a live snapshot of the dashboard during testing:

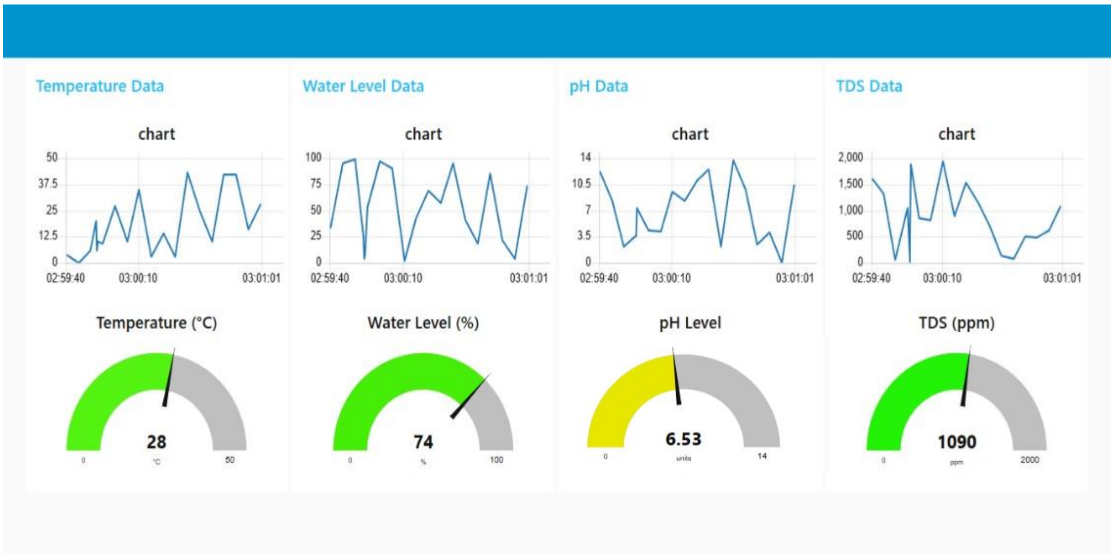

Figure 5-1: Node-RED dashboard showing real-time sensor data and actuator control options.

2. Hardware Setup and Sensor Connections

The hardware prototype of the HydroFarmIoT system was assembled on a breadboard for flexible testing and adjustments, but later on were installed on a PCB board. All sensors and actuators were wired to the ESP32 microcontroller using jumper wires. Key components included the pH sensor, TDS sensor, DS18B20 water temperature sensor,and ultrasonic water level sensor.

The figure below illustrates the complete wiring layout of the system during testing. This setup was used to validate data collection.

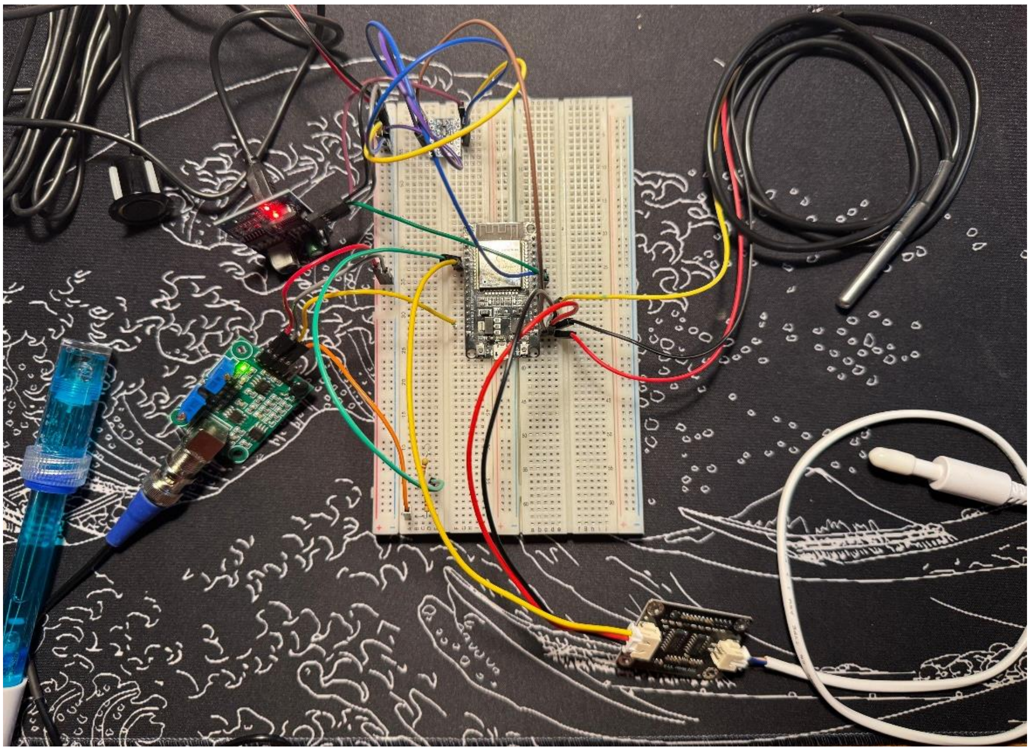

Figure 5-2:ESP32 microcontroller connected to sensors on a breadboard.

This physical arrangement enabled easy troubleshooting and quick reconfiguration when needed. During testing, care was taken to isolate analog inputs, reduce interference, and ensure all modules were powered appropriately.

3. Plant Growth Over Time

To evaluate the practical impact of the HydroFarmIoT system on plant development, a small-scale hydroponic tower was used to grow leafy greens over a period of three months. Throughout this period, the automated system maintained optimal environmental conditions based on sensor readings, ensuring proper pH, nutrient levels, and water availability.

The images below demonstrate the visible progression in plant growth over three consecutive months under the managed environment of the system.

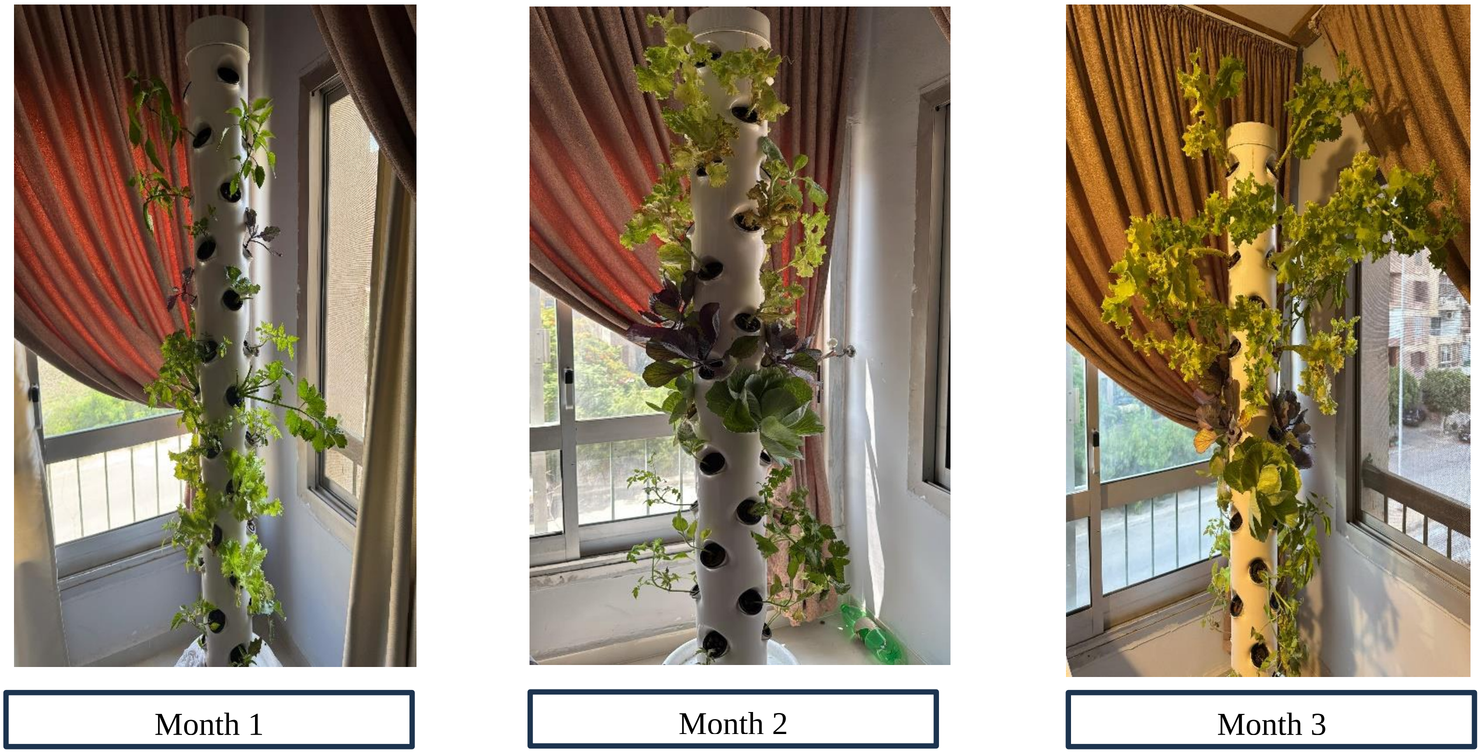

Figure 5-3: Monthly progression of plant growth under HydroFarmIoT system (Month 1, Month 2, Month 3)

This growth progression highlights the effectiveness of real-time environmental control. The plants showed improved size, color, and vitality over time, indicating that the system maintained suitable conditions consistently and reliably.

Summary

The results and observations presented in this section confirm the successful implementation of the HydroFarmIoT system. The Node-RED dashboard provided a responsive and user-friendly interface for monitoring and control. The hardware setup, including sensors and actuators connected to the ESP32, functioned reliably during testing.

Visual documentation of the plant growth over three months further demonstrated the system's practical value in maintaining optimal growing conditions. Together, these outcomes validate the system’s performance in both technical and real-world agricultural scenarios.

Chapter 6 Cost Analysis

In this chapter, we break down the cost of each hardware component.

Hardware Costs

Component	Quantity	Unit Price (EGP)	Supplier	Total Price (EGP)
Microcontroller (ESP32)	1	385	Circuits Electonics	385
pH Sensor	1	1.6k	Circuits Electonics	1.6k
pH Buffer solution	1	250	Amazon	250
Water Temperature Sensor	1	165	Circuits Electonics	165
Water Level Sensor	1	450	Circuits Electonics	450
Logic Level Converter	1	45	Circuits Electonics	45
TDS Sensor	1	850	Circuits Electonics	850
Miscellaneous (wires, connectors, enclosure)	1	440	Circuits Electonics	440
Solenoid valve	2	250	Circuits Electonics	500
2 Relay Module	1	75	Circuits Electonics	75
Power Adapter (5V & 12V)	2	90	Circuits Electonics	180
PCB Board	1	1700	Circuits Electonics	1700
LED	1	950	Amazon	950
LCD Display via I2C	1	170	Circuits Electonics	170
PVC vertical pipe **with 40** net pots and water pump	1	5k	Amazon	5k

Table 6.1:This table lists all hardware components used in the HydroFarmIoT system along with their quantities, unit prices, and total costs.

Total Hardware Cost: 12,760 EGP or 254.95 USD

Chapter 7 Time Plan

In this chapter, the main milestones and stages of the project are illustrated in the following table, the aid of the Gantt chart shown in **Figure 7.1**, also lists the task holder of each task.

Task No.	Task Description	Task Holder	Expected duration to be executed
1	Defining the Problem	Ibrahim & Hussein	1 week
2	Surveying Literature: methods of monitoring and controlling hydroponic systems	Ibrahim	2 weeks
3	Surveying Literature: Standard IoT systems and AI models	Hussein	2 weeks
4	Analyzing System and collecting requirements	Hussein	2 weeks
5	Preparing System requirements specifications	Ibrahim	2 weak
6	Designing front-end: wireframe of UI	Ibrahim & Hussein	2 weeks
7	Designing back-end: IoT H/W schematic and UML diagrams	Ibrahim & Hussein	2 weeks
8	Writing thesis and reports: chapters 1-2	Ibrahim & Hussein	1 week
9	Writing thesis and reports: chapters 3-5	Ibrahim & Hussein	2 weeks
10	Verifying thesis and preparing the presentation	Ibrahim & Hussein	1 week

Table 7.1: This table outlines the scheduled tasks, milestones, and their corresponding timeframes throughout the development of the HydroFarmIoT system.

Gantt Chart

Task		Week in Semester																
Description	Holder	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Problem Defintion	Hussein & Ibrahim																	
Surveying Litreature: Part 1	Ibrahim																	
Surveying Litreature: Part 2	Hussein																	
Writing thesis and reports: chapters 1-2	Hussein & Ibrahim																	
Writing thesis and reports: chapters 3-5	Hussein & Ibrahim																	
Analyzing System and collecting requirements	Hussein																	
Preparing System requirements specifications	Ibrahim																	
Designing front-end: wireframe of UI	Hussein & Ibrahim																	
Designing back-end: IoT H/W schematic and UML	Hussein & Ibrahim																	
Verifying thesis and preparing the presentation	Hussein & Ibrahim																	

Figure 7-1: Gantt Chart for Project timeline

Chapter 8 Conclusion and Future work plan

8.1 Conclusion

The Grad I phase of this project laid a strong theoretical foundation for the design and development of an Automatic Hydroponic System, focusing on system architecture, component selection, and the preliminary planning for integration. Moving into Grad II, the project progressed from theoretical models and initial concepts to a fully functional, real-world prototype.

The following summarizes the key outcomes of this project:

- **System Implementation and Hardware Integration:**
  The hardware implementation was completed by integrating sensors and actuators with an ESP32 microcontroller. The system included pH, TDS, water temperature, light, and water level sensors, as well as actuators such as water pumps and solenoid valves. A detailed schematic was developed and used to assemble the physical system. Hardware challenges such as unstable sensor readings and power management issues were successfully identified and resolved.

- **Software Development and IoT Integration:**
  Control algorithms were developed using Arduino C++ to manage sensor readings and control actuator behavior. A real-time monitoring dashboard was implemented on Node-RED, providing live data visualization and allowing both manual and automatic system control. Additionally, data logging functionality was incorporated through database integration, enabling historical tracking of system performance.

- **System Testing and Validation:**
  The complete system underwent rigorous testing under simulated and real-world conditions. These tests validated the accuracy and stability of sensor readings, the reliability of actuator responses, and the system's ability to maintain optimal environmental conditions. Operational parameters such as nutrient delivery intervals, pH balancing, and water flow rates were optimized based on test results.

- **Challenges and Lessons Learned:**
  Several important lessons were learned throughout the implementation phase. Sensor calibration in real-world environments proved more complex than expected and required careful adjustment. Maintaining network stability and managing power distribution were essential in ensuring continuous operation. Furthermore, the project emphasized the importance of balancing hardware limitations with effective software-based solutions.

The successful completion of this project demonstrated the practical feasibility of automating hydroponic farming systems using IoT technologies. The project delivered a fully operational prototype supported by a reliable monitoring and control platform, paving the way for future enhancements.

8.2 Grad II Work plan

The Grad II phase will focus on implementing the system in a real-world prototype and optimizing its performance. Below is the detailed work plan:

1. **Hardware Development:**
   - Complete the hardware implementation, integrating all sensors and actuators into the system.
   - Ensure proper power management for all components, including external power supplies for high-power devices like pumps and valves.
   - Address hardware challenges identified in Grad I, such as sensor noise and relay delays.
2. **Software Development:**
   - Develop and test the control algorithms for water flow, nutrient delivery, and environmental adjustments.
   - Implement data logging and real-time monitoring via IoT, allowing users to track system performance remotely.
   - Create a user interface for easier system interaction, such as a mobile app or web dashboard.
3. **Testing and Validation:**
   - Conduct thorough system testing under simulated and real-world conditions to evaluate performance, stability, and reliability.

- Optimize the system based on test results, focusing on reducing power consumption and increasing response time.
- Validate the system's ability to maintain optimal conditions for plant growth, such as pH balance, nutrient levels, and water flow.

4. **Scalability and Improvements:**
    - Explore ways to make the system scalable for larger hydroponic setups.
    - Research and integrate additional technologies, such as machine learning for predictive analysis and solar power for energy efficiency.
    - Incorporate user feedback to refine the system's usability.
5. **Final Documentation and Presentation:**
    - Prepare detailed documentation, including technical designs, test results, and system performance reports.
    - Develop a polished presentation for the final thesis defence. showcasing the system's features, benefits, and potential applications.

References

[1] S. Kaur and B. Dewan, “Hydroponics: A Review on Revolutionary Technology for Sustainable Agriculture.,” *Asian Journal of Agricultural and Horticultural Research,* vol. 10, no. 4, pp. 297-302, 2023.

[2] C. B. D. Kuncoro, M. B. Asyikin and A. Amaris, “Development of an Automation System for Nutrient Film Technique Hydroponic Environment,” *ISSAT,* vol. volume 207, pp. 437 - 443, 2021.

[3] S. D. Putra, H. E. F. Cahyadi, K. Anggriani and M. H. I. S. Jaya, “Development of Smart Hydroponics System Using AI-Based Nutrient Film Technique,” *International Journal of Agricultural Research and Technology,* vol. 9, no. 4, pp. 234-241, 2022.

[4] N. A. M. Hatipi, S. S. Sarnin, M. N. Md Tan, M. R. Dohad, N. F. Kamaruzaman, N. F. Naim and A. dris, “IoT-Based Smart Vertical Hydroponic System for Chili Plant,” *Journal of Advanced Research in Applied,* no. 2, pp. 124-133, 2025.

[5] M. Ali, V. Suresh, S. Sravani, S. Srinivasan and T. Logasundari, “IoT Based Automated Indoor Hydroponic Farming System,” *E3S Web of Conferences,* vol. vol. 547, p. p. 02002, 2024.

[6] P. V. Kulkarni, V. Gohokar and K. Kulkarni, “Smart Hydroponic Systems: Optimizing Nutrient Levels with IoT,” *JES,* vol. 20, no. 2, pp. 499 - 507, 2024.

[7] H. T. Ng, Z. K. T. K. Tham, N. A. Abdul Rahim, A. W. Rohim, W. W. Looi and N. S. Ahmad, “IoT-enabled system for monitoring and controlling vertical farming operations,” *IJRES,* vol. 12, no. 3, pp. 453 - 461, 2023.

[8] A. C. H. Austria, J. S. Fabros , K. R. G. Sumilang , J. Bernardino and A. C. Doctor , “Development of IoT Smart Greenhouse System for Hydroponic Gardens,” *IJCSR,* vol. 7, pp. 2111-2136, 2023.

[9] U. Thapa, N. N. Hansda, S. Kundu, A. Giri, D. Tamang and A. O. Rahaman, “Advancements in Hydroponic Systems: A Comprehensive Review,” *Archives of Current Research International,* vol. 24, no. 11, p. 317–328, 2024.

[10] B. Thapa, P. Bhandari, R. G.C, Y. Acharya and S. Phuyal, “Application of Hydroponic System,” *KEC Journal of Science and Engineering,* vol. 8, no. 1, pp. 33-37, 2024.

ملخص المشروع باللغة العرب

نظام زراعة مائية عمودية متكامل مع إنترنت الأشياء :HydroFarmIoT

تزايد الطلب على الغذاء نتيجة للنمو السكاني العالمي والتوسع العمراني يشكّل ضغطًا كبيرًا على أساليب الزراعة التقليدية التي تعتمد على الأراضي الصالحة للزراعة ووفرة الموارد المائية. ومع تزايد ندرة هذه الموارد بسبب التوسع الحضري وتغير المناخ ونقص المياه، تُعد أنظمة الزراعة المائية (الهيدروبونيك) بديلاً مستدامًا، حيث تتيح الزراعة دون تربة مع تقليل استهلاك المياه. ومع ذلك، تفتقر العديد من الأنظمة الحالية إلى الأتمتة ولا تزال تتطلب إدارة يدوية مرهقة. يأتي مشروع :HydroFarmIoT نظام الزراعة العمودية المائية المتكامل مع إنترنت الأشياء لمعالجة هذه التحديات من خلال توظيف تقنيات إنترنت الأشياء لأتمتة مراقبة المعايير الحيوية مثل درجة الحموضة(pH) ، ومستويات المغذيات، ودرجة الحرارة، ومستوى المياه، ومن خلال استخدام الحساسات والمشغّلات في بيئة زراعة عمودية، يقوم النظام بالحفاظ تلقائيًا على الظروف المثلى للنمو مع إمكانية الإدارة عن بُعد. يهدف هذا المشروع إلى تمكين الزراعة على مدار العام في البيئات الحضرية، وتقديم حل قابل للتوسّع ومستدام لمواجهة تحديات الأمن الغذائي في المناطق الحضرية.

جامعة أكتوبر للعلوم الحديثة والآداب

كلية الهندسة

قسم هندسة نظم الحاسبات

نظام زراعة مائية عمودية متكامل باستخدام إنترنت الأشياء :HydroFarmIoT

مشروع التخرج

مُقدَّم ضمن متطلبات درجة البكالوريوس في هندسة نظم الحاسبات

الجزء الثاني

إعداد

حسين غريب صبحي 213423

إبراهيم عطية 210071

إشراف

د. أحمد أيوب

2025-2024