Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED

By

See Keng Lek

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)
COMMUNICATIONS AND NETWORKING
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: _____Development of Fertilizer Automation in IoT-Based __________

____ Agriculture Monitoring Framework using Node-RED __________

Academic Session: ___June 2024__________

I ____________________SEE KENG LEK_________________________

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ _________________________

(Author's signature) (Supervisor's signature)

Address:

___91, Jalan Haji Baki,________

____84000, Muar, Johor______

Goh Hock Guan

Supervisor's name

Date: _____10/09/2024________ **Date**: 11/9/2024

FACULTY OF INFORMATION AND COMMUNICATION AND TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: ___10/09/2024___

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that SEE KENG LEK (ID No: 20ACB01973) has completed this final year project/ dissertation/ thesis* entitled "Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED" under the supervision of Ts. Dr. Goh Hock Guan (Supervisor) from the Department of Computer and Communication Technology, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*SEE KENG LEK*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ___________________________

Name : _______See Keng Lek___

Date : ________10/9/2024________

ACKNOWLEDGEMENTS

I would like to extend my heartfelt gratitude to my esteemed supervisor, Ts Dr. Goh Hock Guan, for entrusting me with the invaluable opportunity to embark on this project. His unwavering support and guidance have been instrumental in shaping my journey, offering profound insights, and imparting invaluable wisdom throughout the developmental process. Ts Dr. Goh's discerning feedback and constructive suggestions have not only enriched my understanding but have also propelled me towards greater heights of achievement. To him, I extend a million thanks for his mentorship and dedication.

Furthermore, I wish to express my profound appreciation to my circle of friends, whose unwavering patience and steadfast support have been a constant source of strength and encouragement during challenging times. Their presence has illuminated even the darkest of moments, reminding me of the importance of camaraderie and solidarity in the pursuit of shared goals. To each of them, I extend my heartfelt gratitude for their friendship and unwavering belief in my endeavors.

Last but certainly not least, I am deeply indebted to my beloved parents and family members for their boundless love, unwavering support, and unyielding encouragement throughout this journey. Their enduring faith in my abilities has been the cornerstone of my success, providing me with the motivation and inspiration to overcome obstacles and strive for excellence. To my parents and family, I offer my deepest gratitude for their immeasurable sacrifices and unwavering belief in my potential.

In conclusion, I am immensely grateful to each and every individual who has played a role, big or small, in shaping my path and contributing to my growth and development. Their collective support and encouragement have been the driving force behind my achievements, and for that, I am eternally thankful.

ABSTRACT

Amidst the backdrop of a rapidly expanding global population and mounting concerns surrounding food security, it becomes imperative to revolutionize traditional farming practices while maintaining environmental sustainability. This study employs cutting-edge technology, notably the Internet of Things (IoT), to devise an innovative solution tailored for indoor crop cultivation, focusing on water-intensive crops such as strawberries and cabbages. This specialized system meticulously monitors critical variables such as water quality, ensuring optimal conditions for plant growth, while also automating nutrient delivery processes. Additionally, the system incorporates water recycling mechanisms, thus promoting eco-conscious farming practices. Farmers stand to benefit significantly from this advanced framework, as it enables increased crop yield, resource conservation, and environmental stewardship. By disseminating our findings, we aspire to catalyze widespread adoption of smart agricultural technologies, fostering a more sustainable and environmentally friendly approach to farming worldwide.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CO_2	Carbon Dioxide
GPIO	General-Purpose Input Output
AC	Alternating Current
DC	Direct Current
OS	Operating System
RAM	Random Access Memory
ROM	Read Only Memory
GUI	Graphical User Interface
NFT	Nutrient Film Technique
SDLC	Software Development Life Cycle
EC	Electrical Conductivity
TDS	Total Dissolved Solids
MQTT	Message Queuing Telemetry Transport

CHAPTER 1

Project Background

In this chapter, we present the background and motivation of our research, our contributions to the field, and the outline of the thesis. Our research project is motivated by the urgent need to modernize agricultural practices in the face of mounting global challenges. With a growing global population, food security has become a paramount concern. Recent estimates place Malaysia's population at 33.2 million, reflecting a 1.6 percent increase in the first quarter of 2023. This demographic trend highlights the pressure to meet escalating food demands and underscores the critical importance of stable and sufficient food supplies. However, alongside the imperative of food security, there's an intrinsic link to environmental sustainability. Conventional farming practices have often contributed to soil degradation, water pollution, and excessive greenhouse gas emissions, disrupting ecosystems and depleting vital resources. Furthermore, technological advancements in IoT and automation offer a transformative potential to optimize resource utilization and enhance agricultural efficiency. Our research aims to bridge the gap in integrating real-time data-driven insights with automated actions, particularly in the context of fertigation, thereby empowering farmers to improve crop yields, curtail resource consumption, and reduce the environmental footprint of agriculture. Our contributions lie not only in addressing food security and sustainability concerns but also in harnessing cutting-edge technologies to usher agriculture into a more efficient, eco-conscious, and productive future.

1.1 Motivation

In recent years, the field of agriculture has witnessed a transformative revolution through the widespread adoption of Internet of Things (IoT) technologies. These innovations have propelled agricultural practices into the digital age, provide farmers with tools to enhance productivity, optimize resource usage, and minimize environmental impact. IoT-enabled agriculture monitoring frameworks, in particular, have emerged as a pivotal component of this transformation. These frameworks have provided farmers with real-time data collection and analysis capabilities, and also provided them with the insights to make informed decisions for precise crop management.

The trend towards indoor plant growth underscores the need to adapt these technologies to meet the unique challenges and opportunities presented by controlled indoor environments. According to [1], indoor farming is a new method of growing vegetables and other plants under controlled environmental conditions. Indoor farming offers several advantages, including microclimate control, year-round production, reduce pesticide usage, water conservation, and greater crop diversity [1]. While controlled indoor environments offer various advantages, they necessitate meticulous management of factors like temperature, humidity, lighting, electrical conductivity and total dissolved solids. Robust IoT solutions are crucial for monitoring and addressing these intricacies, ensuring successful indoor farming practices with minimal environmental impact.

Fertilization remains a critical aspect of crop management, whether indoors or outdoors, requiring precise timing and accurate dosages for optimal yield and resource efficiency. Traditional fertilization methods often lack the finesse required to meet these demands, resulting in suboptimal outcomes and environmental concerns. For instance, farmers in Patzún implemented traditional fertilization in their plantation, using components such as mixed ashes, kitchen scraps, crop residues, weeds, leaf litter, and manure to produce compost [2]. However, previous studies by [3] & [4] have summarized that plants with oversufficient nutrients will increase the acceptability to pest populations.

The motivation of this research project emerges from the imperative to modernize agricultural practices in the face of mounting global challenges, particularly food security and environmental sustainability. As the global population continues to increase, placing pressure on food supplies. Recent estimates place the total population of Malaysia at 33.2 million, signifying a 1.6 percent increase compared to first quarter 2023, underscoring the pressure to meet escalating food demands [5]. This demographic trend is reflective of a broader global challenge, which is ensuring a stable and sufficient food supply to sustain the well-being of nations. Based on the research done by [6], the global hunger was measured by the prevalence of undernourishment, exhibited minimal changes from 2021 to 2022 but remained notably higher than before pandemic. In 2022, approximately 9.2 percent of world's population experienced hunger, a 7.9 percent higher compared to the year 2019.

Furthermore, moderate to severe food insecurity has affected a substantial portion of the global population in 2022, impacting 29.6 percent of people which is equivalent to 2.4 billion people, and 11.3 percent of people were experiencing severe food insecurity. Notably, food security tends to disproportionately burden women and individuals residing in rural areas. Alarmingly, more than 3.1 billion people worldwide, accounting for 42 percent of the population, were unable to afford nutritious diet in 2021, marking an increase of 134 million individuals compared to 2019.

In conjunction with addressing food security concerns, the modernization of agriculture is intrinsically linked to imperative of environmental sustainability. Environmental sustainability refers to the importance of preserving resources like clean air, water, and wildlife for future generations [7]. Additionally, the U.S. Environmental Protection Agency (EPA) further defines environmental sustainability as the concept of meeting today's needs without compromising the ability of future generations to meet their own requirements [8]. Conventional farming practices have frequently been culpable in causing soil degradation, water contamination, and the release of excessive greenhouse gases, all of which contribute to the depletion of vital natural resources and disrupt the delicate balance of ecosystems. Research conducted by [9] has pointed out a notable 0.9% increase in global carbon dioxide (CO_2)

emissions resulting from energy combustion and industrial process in 2022. This surge amount to an additional 321 Mt, propelling these emissions to a record-breaking high of 36.8 Gt. These findings are grounded in a comprehensive analysis by the IEA, which meticulously examines energy use, economic indicators, weather patterns, and the latest official national statistics to provide a region-by-region and fuel-by-fuel assessment of global CO_2 emissions.

As a result, this research project endeavors to bridge this gap by crafting a sophisticated IoT-based solution for precise and dynamic fertigation practice. By seamlessly uniting real-time data insights with automated fertigation actions, this research aims to equip farmers with the tools required to increase the crop yields, curtail resource consumption, and curtail the environmental footprint of agriculture. In doing so, we desire to not only address food security and environmental sustainability concerns but also harness cutting edge technologies to propel agriculture into a more efficient, eco-conscious, and productive future.

1.2 Problem Statement

Despite the advancements in IoT-based agriculture monitoring systems, there remains a gap in the integration of real-time data-driven decision-making processes with automated actions, specifically in the domain of fertigation, which combines fertilization and irrigation for efficient nutrient delivery to crops. Traditional fertilization methods often rely on manual observation or pre-defined schedules, which might not effectively adapt to dynamic changes in soil conditions, weather patterns, and plant growth stages. This can lead to suboptimal resource utilization, decreased crop quality, and increased environmental impact due to excessive fertilizer usage.

The introduction of fertigation as a farming practice presents a promising solution to this challenge. Fertigation involves the application of fertilizer solutions through irrigation water, offering superior efficiency compared to traditional fertilization methods. By implementing fertigation practices, the agricultural sector can significantly reduce the fertilizer waste and mitigate land contamination, benefiting both farmers and environmental alike.

This refined problem statement succinctly addresses the central issue of integrating real-time data-driven decision-making with automated actions, specifically within the domain of fertigation, and emphasizes the importance of this integration in enhancing agricultural productivity while minimizing environmental impacts.

1.3 Project Objectives

The aim of the project is to develop a fertilizer system that capable of providing sufficient nutrients for crops grown in indoor farm while also filtering and recycling the hydroponics solution.

The first objective is to develop a system for accurate EC TDS reading by integrating the Raspberry Pi with sensors and actuators. The sensors attached to the Raspberry Pi will be measuring the conditions of the hydroponic containers and then the framework will collect, process, and generate output data. This will provide users with insights into EC TDS values and crop conditions, informing the precise control of actuators to manage the system effectively.

The second objective is to implement automation feature in the framework. Through automation, actuators like water pumps in the framework are allowed to work by themselves based on the data that collected from the sensors. By achieving high accuracy in data collection, automation will enable actuators to function more efficiently with minimal human interaction.

Lastly, the third objective is to develop an automated water filter system to facilitate the recycling of the entire water system within the indoor farm, thereby minimizing waste and promoting sustainability. This automated water filtration system will be designed to effectively remove impurities and containments from hydroponic solution, ensuring that the water remains clean and suitable for reuse.

1.4 Project Scope

The project will primarily focus on key components essential for building a fertilizer mixer for the IoT-based Agriculture Monitoring Framework, leveraging Raspberry Pi 3B+, Node-RED, sensors, actuators, InfluxDB and Telegram.

Firstly, the project will be concentrated on the value of EC TDS sensors that the lettuce and strawberries need. In order to grow these two plants, the EC TDS sensors are one of the most important elements. Integrated with the Raspberry Pi, EC TDS sensor, water pump and water level sensor will collect EC TDS value in the Mixed Solution Tank. Subsequently, Node-RED will capture this data and transmit it to the monitoring device via MQTT. Upon receipt, the data will be stored in the InfluxDB, the time-series database.

Simultaneously, data stored in InfluxDB will be extracted and showcased on the Node-RED dashboard for real-time visualization. Moreover, water level sensor data will govern the operation of the actuators, such as water pumps, responsible for delivering the mixed solution to lettuce or strawberries hydroponic tanks.

Furthermore, a Telegram chatbot will be developed and configured within Node-RED to enable users to monitor water tanks, and control actuators by issuing basic commands. Besides, the chatbot will also be configured to send alert message to the users upon detection of irregular data.

Once the framework has been built, the reliability and the performance of the fertilizer system will undergo thorough evaluation. The seeds of the lettuce and strawberries that has been germinated will be moved in the hydroponic container and placed inside the framework. Then, the growing process of the lettuce and strawberries will be observed, and it will be determined whether both lettuce and strawberries will grow healthily.

1.5 Contributions

To demonstrate the feasibility of an automated indoor farm, it is crucial for the agriculture sector to integrate IoT technologies into its existing systems. This integration involves establishing an optimal indoor farming environment to guarantee crop productivity, quality, nutrient levels, and water reusability. Additionally, this project will serve as a practical illustration of the functionality and effectiveness of the agriculture monitoring framework.

Firstly, the agriculture monitoring framework, seamlessly integrated with Raspberry Pi and an array of sensors, can generate invaluable data insights for farmers. Instead of relying on guesswork for decision-making, users will have access to real-time data representing precise environmental conditions. This data-driven framework significantly enhances operational efficiency. By collecting vital environmental parameters, the system compiles and displays all relevant data on a user-friendly dashboard, allowing for convenient farm monitoring.

Secondly, the incorporation of automation within the agriculture monitoring framework not only reduces labor but also ensures its precision. All manual actuators have been taken over by the automated system, relying on sensors to collect accurate data. This data-driven approach guarantees more accurate outputs. Farmers can redirect their focus towards crop care and management, as the system's decisions are based on real-time sensor data, alleviating their concerns about plant care.

Additionally, the implementation of a Telegram chatbot, operational around the clock, plays a pivotal role in effective farm management. The chatbot offers users the ability to send commands to the Raspberry Pi and obtain real-time farm conditions. Moreover, users can exercise control over actuators, regardless of their location or the time of day. Even when off-duty, farmers can rest assured, as the chatbot monitors the farm condition continuously. In the event of unusual data detection, the chatbot promptly sends alert messages to users, enabling them to make immediate decisions.

Furthermore, the project encompasses the development and integration of an automated fertilizer mixer, enhancing the system's capability to manage nutrient delivery effectively. By automating the fertilizer mixing process based on real-time sensor data, the system ensures precise nutrient dosages, optimizing crop growth and resource utilization. This approach prevents over-fertilization, ensuring the right nutrients are delivered to the crops.

Additionally, the project addresses the crucial aspect of water management by introducing a water recycling system. This system efficiently reuses water within closed-loop circuits, minimizing waste and promoting resource sustainability. By combining these elements within the IoT-based Agriculture Monitoring Framework, the project aims to empower farmers with a comprehensive and automated solution for improved crop health, resource efficiency, and sustainable agriculture practices.

1.6 Report Organization

The report is structured into five chapters, commencing with the Introduction. This section addresses problem statements, project objectives, motivation, project scope, and expected contributions. Following this, the Literature Review in Chapter 2 explores the hardware and software technologies relevant to the project, along with an examination of four existing systems related to the project's context.

In Chapter 3, titled System Methodology, the development model employed in the project is discussed, along with a clear delineation of the hardware and software utilized in constructing the system. Moving forward, Chapter 4, System Design, elaborates on the design aspects of the system, including functional modules and system flows.

Finally, Chapter 5, Conclusion & Recommendation, draws conclusions based on the project's progress and provides recommendations for enhancing the system's functionality.

CHAPTER 2

Literature Review

2.1 Review of the Technologies

2.1.1 Hardware Platform

In this project, the primary hardware component will be a microcontroller or microprocessor board responsible for controlling the entire system, including data collection from sensors and actuator control.

- **Raspberry Pi 3B+**

  The Raspberry Pi is a widely-used microcontroller board favored by developers exploring IoT applications. Initially popular among students due to its affordability and user-friendly interface, its accessibility has increased its adoption. However, the COVID-19 pandemic has led to a rise in its price due to resource shortages affecting production.

Figure 2.1.1.1 Raspberry Pi 3B+

Based on figure 2.1.1.1, the Raspberry Pi 3B+ features 40 General-Purpose Input Output (GPIO) pins compatible with various sensors and actuators. Powered by the Broadcom BCM2837B0 Cortex-A53 (ARMv8) 64-bit SoC running at 1.4 GHz, it also includes 1GB LPDDR2 SDRAM to meet program requirements. Supporting Linux-based operating systems, it offers the capability to run multiple programs simultaneously. Additionally, it features dual-band wireless LAN and a fast Ethernet port for internet connectivity, with power supplied via a 5V/2.5A DC input through a Micro USB port.

- **Arduino Uno Rev3**

Figure 2.1.1.2 Arduino UNO

Illustrated in figure 2.1.1.2, The Arduino Uno, based on the Atmega328P microcontroller, is another popular choice among developers and students due to its affordability and user-friendly nature. As shown in Figure 2.1.1.2, it boasts 14 GPIO pins and 6 analog inputs for sensor connections. Operating at 5V via a power jack, it can be easily configured using a USB cable to a computer and an AC to DC adapter.

- **ESP8266**

Figure 2.1.1.3 ESP8266

The ESP8266 is a low-cost microcontroller featuring Wi-Fi capabilities, produced by Espressif Systems. Equipped with the Tensilica Xtensa® 32-bit LX106 RISC, it supports Real-Time Operating Systems (RTOS) and operates at clock frequencies between 80 to 160 MHz. As depicted in figure 2.1.1.3, it offers a total of 30 pins, with 17 GPIO pins supporting various peripheral duties such as UART, I2C, and SPI. Like the Arduino Uno, it can be programmed using the Arduino IDE platform via a USB cable connection to a desktop.

- **ESP32**

Introducing the ESP32, a versatile and budget-friendly microcontroller with built-in Wi-Fi capabilities, engineered by Espressif Systems, headquartered in China. The ESP32 boasts a robust Tensilica Xtensa® 32-bit LX6 RISC processor, providing ample processing power for a variety of applications. Its Real-Time Operating System (RTOS) support allows for efficient multitasking, enabling the microcontroller to operate seamlessly at clock frequencies ranging from 80 to 240 MHz.

Figure 2.1.1.4 ESP32

The ESP32 comes equipped with an advanced Wi-Fi and Bluetooth transceiver that supports 802.11b/g/n and Bluetooth v4.2 standards. This enables seamless connectivity to Wi-Fi networks and Bluetooth devices, facilitating internet access and IoT functionality. With the ESP32, developers can create connected devices capable of communicating and interacting with other devices and services over the internet, making it an ideal choice for IoT projects and applications.

In terms of pin configuration, the ESP32 offers a generous total of 38 pins, including 34 GPIO (General Purpose Input/Output) pins. These GPIO pins can be configured to perform various functions, including UART, I2C, SPI, PWM, and more, providing flexibility and versatility for a wide range of projects and applications.

Similar to its predecessor, the ESP8266, the ESP32 can be programmed using the Arduino Integrated Development Environment (IDE), simplifying the development process for beginners and experienced developers alike. By connecting the ESP32 to a desktop or laptop via USB cable, developers can leverage the familiar Arduino programming environment to write and upload code to the microcontroller.

With its powerful processing capabilities, integrated Wi-Fi connectivity, and extensive GPIO pin options, the ESP32 offers an ideal platform for developing innovative IoT projects, smart home devices, wearable gadgets, and much more. Whether you're a hobbyist, student, or professional developer, the ESP32 provides a cost-effective solution for bringing your ideas to life in the rapidly evolving world of connected devices.

2.1.2 Firmware/OS

- **Raspberry Pi OS (Raspbian)**

To enable the Raspberry Pi hardware functionality, the Raspberry Pi OS image must be flashed onto an SD card and installed in the hardware. The latest version of the OS optimizes the Linux OS for Raspberry Pi boards, providing a user-friendly interface for easy interaction with the microcontroller board.

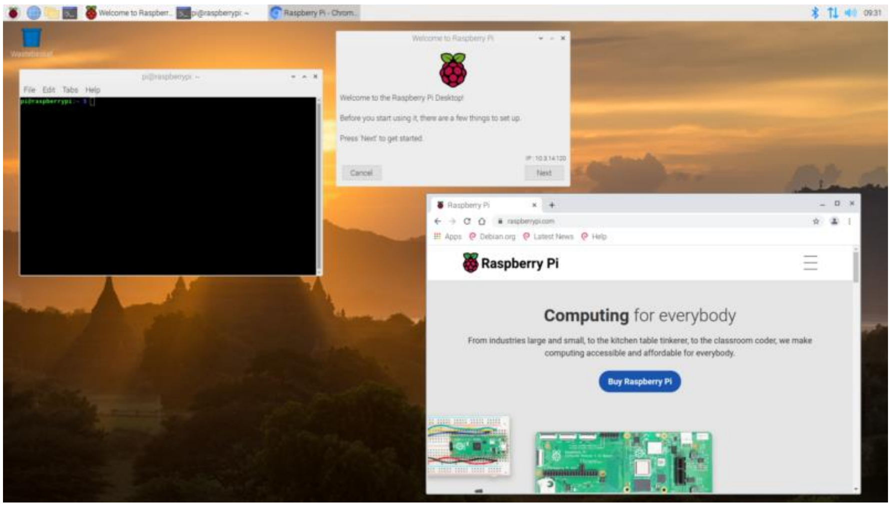


Figure 2.1.2.1 GUI of Raspberry Pi OS (Raspbian)

As shown in the figure 2.1.2.1, the OS furnishes a convenient desktop environment comprising application menus, a web browser, a file manager, and a terminal. Additionally, the desktop displays time and Wi-Fi settings to streamline user operations.

- **Arduino IDE**

In contrast to Raspberry Pi, the Arduino Uno does not operate on an OS; it is a simple board enabling users to program the ATmega328P microcontroller chip. However, a small amount of code, known as a bootloader, is preloaded into the chip's flash memory. The Arduino Uno relies on this bootloader to allow software on the user's PC to upload program code to the flash memory.

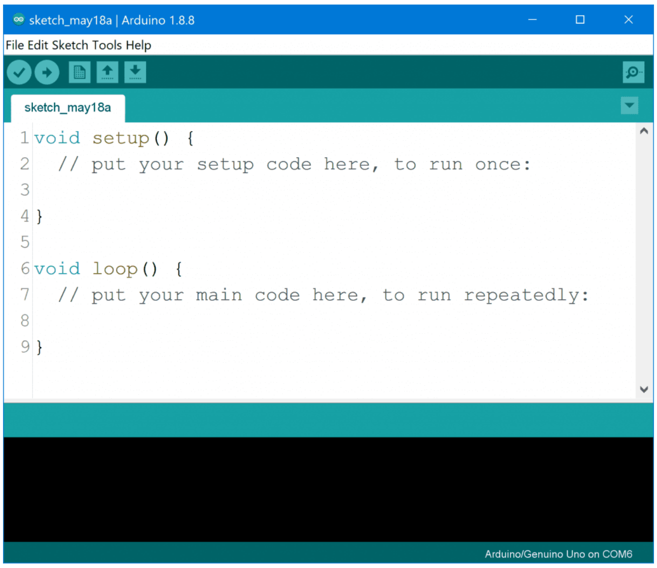


Figure 2.1.2.2 Arduino IDE Application

As shown in figure 2.1.2.2, the Arduino IDE application provides a straightforward and user-friendly interface. Users can write code in the sketch file and utilize available library resources in the software. Once the code for the entire program is written, users can upload the content from the sketch file to the flash memory of the Arduino Uno for desired output.

2.1.3 Database

- **MySQL**

MySQL, also known as Structured Query Language (SQL), is a programming language primarily used for data management in Relational Database Management Systems (RDBMS). Widely utilized for querying relational databases, MySQL organizes data into tables comprising rows and columns, with various tables linked in different ways. It allows users to access records by entering a single command to retrieve extensive data.

- **InfluxDB**

  InfluxDB is a time-series database optimized for storing and retrieving time-series data monitored or tracked over time. Indexed with time order, data points are arranged in sequence within the database. Designed to accommodate and store large volumes of time-series data, InfluxDB includes measurements for data monitoring, tracking, and refinement. It can be utilized for data analysis, sensor data reading, or stock exchange data management.

2.1.4 Programming Language

- **Python**

  Python is a versatile programming language used for high-level programming, offering an object-oriented approach for building small or large-scale projects with simple coding techniques. Unlike other programming languages, Python utilizes significant indentation to enhance code readability, facilitating the production of clear and logical code for ease of understanding.

  Moreover, Python offers numerous libraries that aid developers in project development. By importing these libraries before writing the script, users gain access to functions readily available for use. For embedded devices like Raspberry Pi, Python libraries enable convenient programming for interfacing with sensors and actuators.

- **Node.js**

  Node.js is an open-source JavaScript runtime environment that executes code outside of a web browser. Widely adopted by developers and major companies, Node.js supports cross-platform development and allows writing server-side scripting and APIs. Ideal for backend server and database development, Node.js runs as a standalone application, making it suitable for real-time and streaming web applications.

  Furthermore, the Node-RED programming tool is built on Node.js, leveraging its lightweight runtime and event-driven model. Providing a browser-based editor, Node-RED facilitates easy system deployment by allowing users to wire flows and connect nodes, making it an ideal programming tool for hardware like Raspberry Pi deployed at the network edge.

2.1.5 Summary of the Existing System/ Applications

Hardware platform

Name	Firmware/OS	Strength	Weakness
Raspberry Pi 3B+	Raspberry Pi OS	- Fast CPU performance - Supports Linux OS - Consists of 40 GPIO pins - Supports many programming languages	- High power consumption
Arduino Uno Rev3	Arduino IDE	- Lower power consumption - Supports analogue input - Low cost	- Low CPU performance - Not support Linux OS - Consists of fewer pins - Only support C/C++
ESP8266	Arduino IDE	- Low cost - Wi-Fi enabled - Supports analogue input	- Low CPU performance - Not support Linux OS
ESP32	Arduino IDE	- Low cost - Wi-Fi & Bluetooth enabled - Supports analogue input	- Low CPU performance - Not support Linux OS

Table 2.1.5.1 Summary of Hardware Platform

Database

Description	Strength	Weakness
MySQL	- Store relational data - Fast delivering - Support heavy load	- Complex query optimization - Performance drop when large data is stored
InfluxDB	- Store time-series data - Support real-time analytics - Simple to store and manage	- Limited scaling - Support single series

Table 2.1.5.2 Summary of Database

Programming Language

Description	Strength	Weakness
Python	- High-level language - Used for developing large project	- Need to write a lot codes to run the system
Node.js	- Require minimum effort of code to run the system - Fast understandable	- Not suitable for big project

Table 2.1.5.3 Summary of Programming Language

2.2 Review of Existing Work

The field of agriculture has witnessed a significant transformation over the years, driven by the rapid adoption of cutting-edge technologies, particularly those related to the Internet of Things (IoT). These technological advancements have ushered in a new era of digital agriculture, offering farmers powerful tools to optimize resource utilization, enhance productivity, and mitigate environmental impacts. One of the key developments in this field is the emergence of IoT agricultural monitoring systems. These systems have revolutionized the way farmers collect and analyze data, providing real-time insights into crucial factors affecting crop growth and overall farm management.

While IoT-based solutions have made notable strides in agriculture, a critical area that warrants further investigation is the automated mixing of fertilizers tailored to the specific nutrient requirements of plants in hydroponic systems. Hydroponic farming, characterized by soil-less cultivation, offers numerous advantages in terms of resource efficiency and crop yield but demands precise control over nutrient delivery. This control can be achieved through automated fertilizer mixing systems driven by IoT technology. The significance of this research lies in its potential to address the challenge of ensuring precise and real-time nutrient management in hydroponics, thereby contributing to increased crop yields and sustainable farming practices.

The following literature review will be focused on discussing the previous research on IoT-based Automated Fertilizer Mixers, the comparison with other current fertilizer mixers, limitations of the previous studies, and the proposed solutions towards enhancing the existing framework.

2.2.1 Previous works on Automated Fertilizer Mixer

2.2.1.1 Hydroponic nutrient mixing system based on STM32

According to [10], the authors focused on the development of sophisticated systems designed to address the unique nutrient requirements of individual plants in hydroponic environments. The authors have introduced a system that allows for the individualized mixing of hydroponic nutrient solution based on specific plant nutrient recipes. This level of customization is crucial because different plant species, growth stages, and environmental conditions can necessitate distinct nutrient formulations. The microcontroller SMT32 and embedded device Raspberry Pi are used in this project.

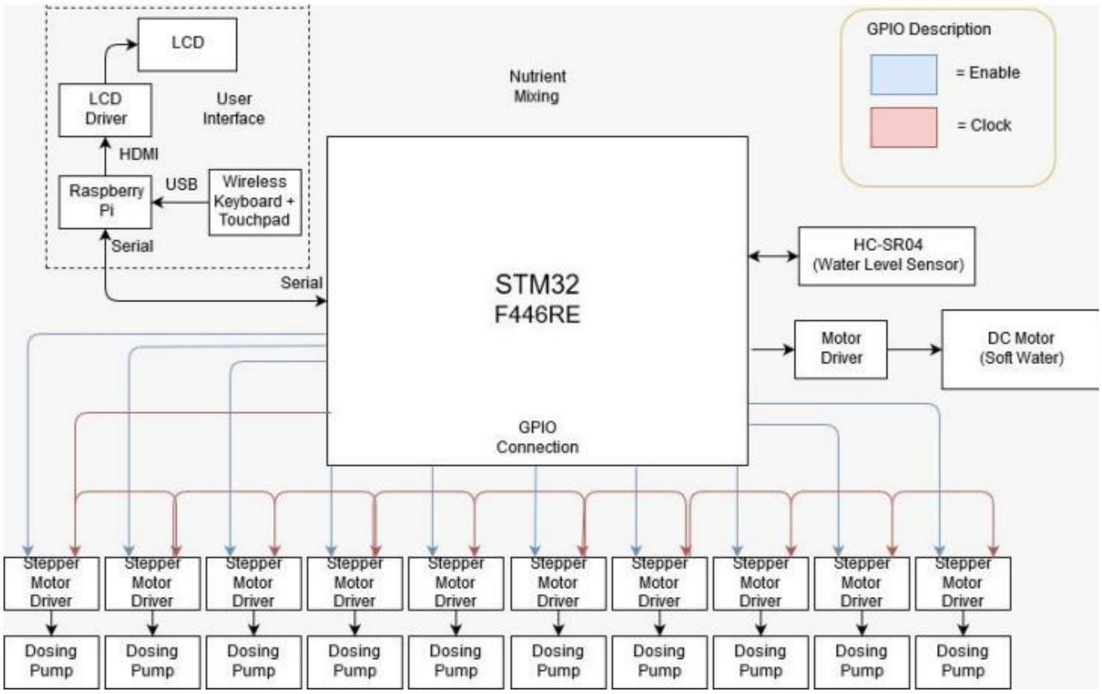


Figure 2.2.1.1.1 Block Diagram of STM32 based Hydroponic Nutrient Mixing System

In figure 2.2.1.1.1 has clearly shows that the system design is divided into 2 parts, which are the system user interface and nutrient mixing controller. The system user interface consists of a database to store plant recipes, fertilizer composition, and sending command to nutrient mixing system confirming to the plant recipe. On the other hand, the nutrient mixing controller, a STM32 F446RE microcontroller was connected with various sensors and motors, such as water level sensor, stepper motor driver, motor driver, DC motor, the microcontroller will be processing the command which sends from the user interface and control the dosage of each dosing pump motor.

The study which was done by Tamdil et al. has showcased the integration of Internet of Things (IoT) technology to enhance nutrient management in hydroponics. IoT-enabled sensors and controllers were used to monitor and control key parameters, such as electrical conductivity (EC) and pH levels, ensuring precise nutrient delivery to plants. Other than that, Tamdil et al. have also shown their research has highlighted the significance of delivering precise nutrient levels to plants. It emphasized that variations in nutrient absorption can directly impact plant growth, making accurate nutrient management a critical factor in hydroponic systems.

Apart from that, Tamdil et al. have also explored the solubility of different fertilizer components at varying temperatures, highlighting that higher temperatures can lead to better solubility. This insight is essential for optimizing nutrient solutions in hydroponics, where maintaining a consistent nutrient concentration is vital. The study detailed the use of dosing pumps equipped with stepper motor drivers to precisely deliver individual fertilizer concentrates. The collaboration process for these dosing pumps was addressed to ensure accurate volume measurements. Tandil et al. have also reported that the developed system underwent rigorous testing, with results indicating a maximum error rate of 9%. The accuracy of the water-filling system, controlled by a proportional controller, was also evaluated, yielding an error rate of approximately 2.784% for a 5-liter water volume.

2.2.1.2 Automatic hydroponic nutrient mixing for hydroponic NFT and fertigation.

The author in [11] underscores the pressing challenge of feeding a growing global population and advocates for the adoption of modern technology to devise innovative solutions. Notably, vertical farming is presented as a strategy to maximize crop yields in confined spaces by harnessing artificial light sources. In their study, Hartatono et al. (2021) introduce two distinct hydroponic systems: the circulated Nutrient Film Technique (NFT) system and the non-circulated Fertigation system. NFT, an advanced hydroponic technique known for its efficient nutrient recycling, reuses nutrient solutions, while fertigation integrates both fertilization and irrigation to cater to the nutritional needs of soilless crops [11]. Both methods are crucial responses to the challenge of providing essential nutrients to plants in hydroponic environments.

Hartatono et al. (2021) accentuate the paramount importance of precise nutrient management in hydroponic systems. Unlike traditional soil-based agriculture, where nutrients are naturally present in the soil, hydroponics necessitates meticulous control over nutrient levels. The study underscores that even minor fluctuations in nutrient absorption can exert a substantial influence on plant growth, underscoring the imperative of delivering nutrients with utmost accuracy.

Moreover, the research delves into system design, with a specific focus on the two hydroponic systems mentioned earlier: NFT and fertigation. These systems both employ vertical farming techniques and incorporate an array of features including nutrient mixing, temperature regulation, lighting control, and soil moisture sensing. Utilizing these components is essential to ensure optimal plant growth conditions.

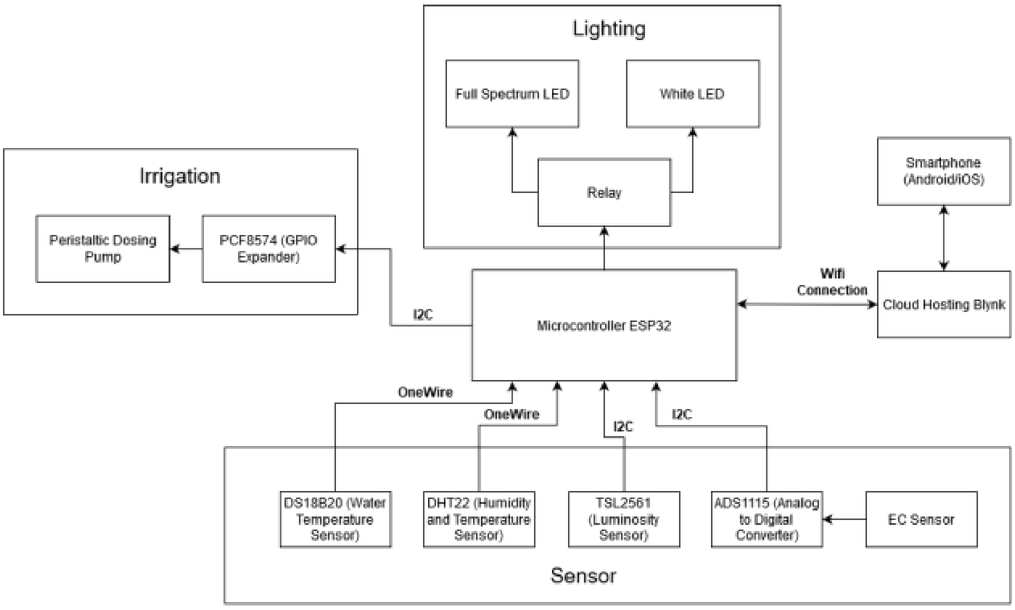


Figure 2.2.1.2.1 NFT System Block Diagram

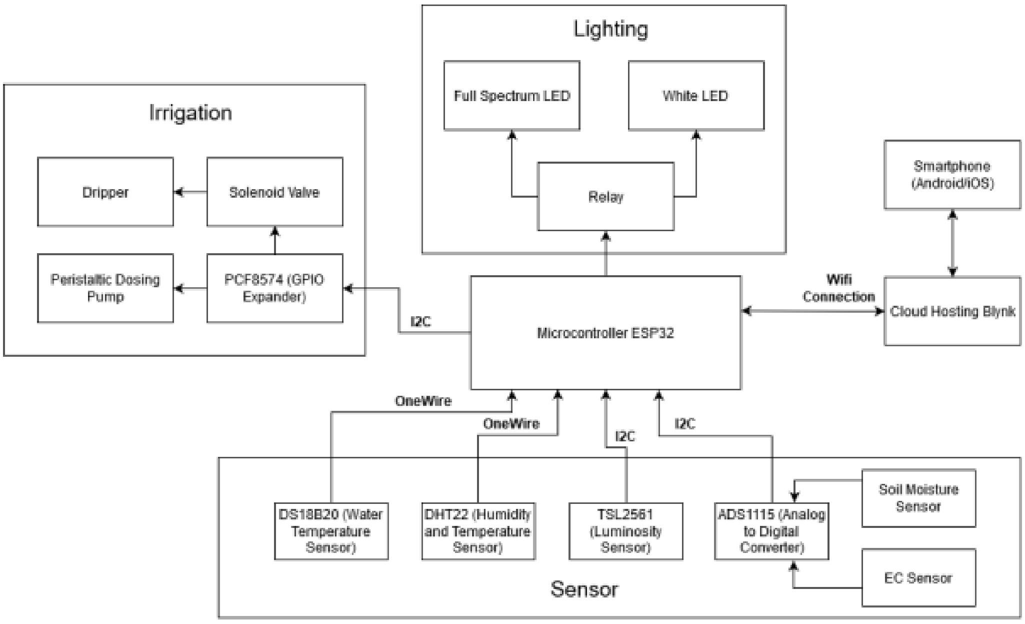


Figure 2.2.1.2.2 Fertigation System Block Diagram

Figure 2.2.1.2.1 depicts the NFT System Block Diagram, while Figure 2.2.1.2.2 illustrates the Fertigation System Block Diagram, providing a visual representation of the key components in each hydroponic system. Both systems integrate DHT22, DS18B20, and EC sensors to guarantee data accuracy. Additionally, the performance of the scheduler system is highlighted, with rigorous examination of its effectiveness in controlling lighting schedules. The automatic EC control system is also subject to scrutiny, assessing its capability to uphold nutrient levels within predefined parameters.

In summary, this research offers a comprehensive overview of the synergy between technology and agriculture within the realm of hydroponics. It prominently underscores the pivotal role of sensor-based technology in optimizing hydroponic systems, ultimately contributing to more efficient and sustainable agricultural practices.

2.2.1.3 Smart ecosystem for hydroponic land in the hydroponic farmers group guided by CSR PT. Otsuka Indonesia as improved quality and quantity of harvest result.

In recent years, hydroponic agriculture has gained prominence as a sustainable method for cultivating crops. However, hydroponic farming comes with its own set of challenges, including the need for precise monitoring and control of environmental factors. This has led to the development of systems aimed at improving the efficiency and yield of hydroponic crops.

Previous research has explored various approaches in addressing these challenges. Several studies have focused on developing monitoring and control systems to aid hydroponic farmers. These systems typically involve the placement of sensors directly within hydroponic setups to collect essential data.

For example, prior works have investigated the use of sensors to measure parameters like temperature, humidity, and light intensity. These data are crucial for maintaining optimal growing conditions for hydroponic crops. Additionally, past studies have explored that the integration of database to store and manage the collected data effectively. Thie approach allows for real-time monitoring and provides historical data for analysis.

In this project, a water circulation monitoring and control system has been developed. It was developed to provide convenience to hydroponic activists to monitor and control hydroponics. The initial process of the system was placing the sensor on the hydroponics to detect the sensor’s data along with the parameters needed to hydroponic mustard plants. The system also consists of a database, which was used to store the collected data and display the data as the output of the monitoring process on the website. Then, the hydroponic water circulation will be controlled by the processed sensor data. In this control process the data that will be used are the temperature and humidity, these two data will be processed by using the Sugeno fuzzy method to determine the time to flow hydroponic water.

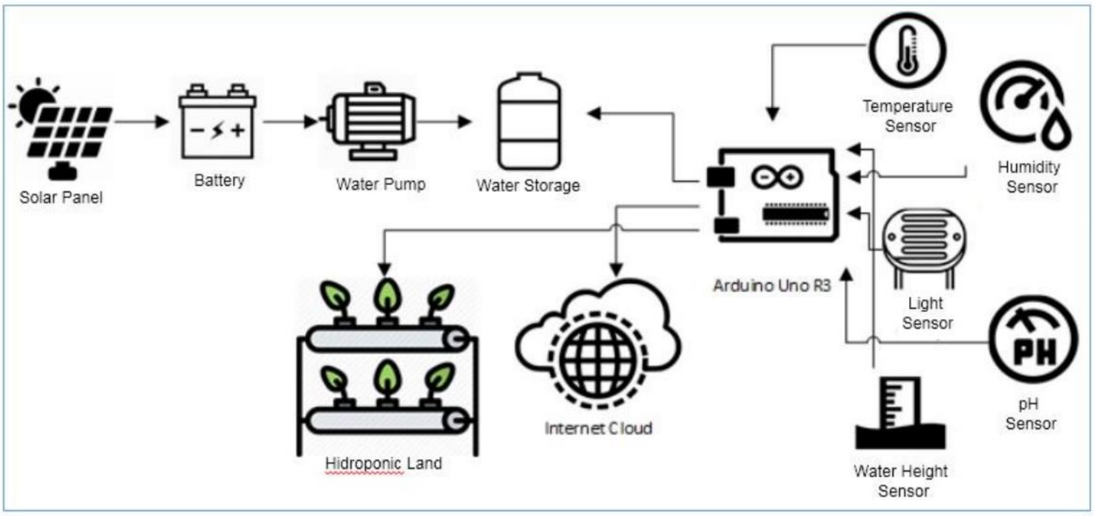


Figure 2.2.1.3.1 Device design of the water circulation monitoring and control system

D K P Aji et al. has mentioned that the development of a water circulation monitoring and control system which developed in this project have the potential to enhance crop yields, optimize resource utilization, and alleviated the burden on hydroponic farmers.

2.2.1.4 Fuzzy-based automated nutrient solution control for a hydroponic tower system.

In recent years, the development of precision agriculture technologies has revolutionized the field of hydroponics, offering more efficient and sustainable methods for crop cultivation. One notable area of advancement is the design and implementation of automated nutrient solution control systems. These systems play a pivotal role in optimizing nutrient delivery to plants, ensuring their health and maximizing yields.

A key contribution to this domain comes from the research conducted by Rahmat et al., who designed a sophisticated nutrient solution control system specifically tailored for Nutrient Film Technique (NFT) hydroponic towers. The system's architecture revolves around the utilization of a microcontroller in conjunction with a suite of sensors, including Electrical Conductivity (EC), pH, light, and DHT11 sensors. This combination of sensors allows for the continuous monitoring of crucial environmental parameters within the NFT hydroponic tower.

The selection of these sensors is not arbitrary; rather, it is a result of a thoughtful consideration of the variables critical for plant growth and nutrient absorption. The EC sensor measures the electrical conductivity of the nutrient solution, providing insights into its salinity and nutrient concentration. The pH sensor ensures the maintenance of an optimal pH level, crucial for nutrient availability. Light sensors gauge the intensity of illumination, while the DHT11 sensor monitors temperature and humidity conditions. These sensors collectively form a comprehensive sensory network, generating real-time data crucial for precision agriculture.

The acquired sensor data is systematically collected and stored in a data storage unit, forming a valuable dataset for subsequent analysis. Here, Rahmat et al's research distinguishes itself with its innovative approach to data analysis. The microcontroller processes the sensor data, determining the runtime of dosing pumps responsible for the treatment solutions. Additionally, it controls the primary pump, which circulates the nutrient solution throughout the NFT hydroponic tower's growing channels.

This level of real-time data processing and control empowers the system to make dynamic adjustments to nutrient levels ensuring that plants receive precisely the nutrients they require at any given moment. The implications of this automated nutrient solution control system extend beyond mere convenience; it directly contributes to enhanced crop yields, resource efficiency, and sustainability in hydroponic farming.

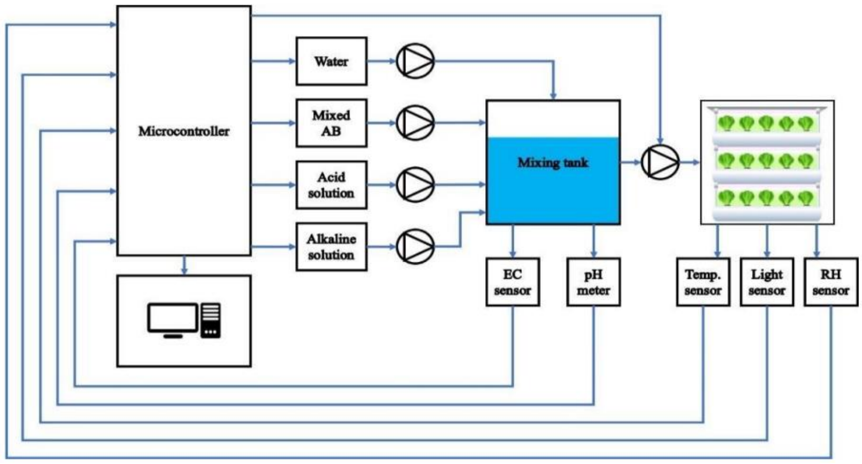


Figure 2.2.1.4.1 System design of the nutrient solution control system

2.2.1.5 Summary of the Existing Systems

No.	Existing Systems	Advantages	Disadvantages	Critical Comment
1.	Hydroponic Nutrient Mixing System based on STM32	- Automated nutrient solution production. - Individual dosing pumps for precise mixing. - Real-time nutrient calculation. - User-friendly interface.	- Requires trained personnel for operation. - Potential for sediment formation in diluted solution.	This project aims to develop a Hydroponic Nutrient Mixing System based on STM32 which offers insight into the accuracy of the dosing pumps and that need people who knows how to operate.
2.	Automatic Hydroponic Nutrient Mixing System	- Enables automatic monitoring and control of nutrient concentration and pH levels. - Utilize IoT cloud for remote user control and monitoring. - Automation is enabled in the system.	- The DHT22 sensor accuracy will decrease in colder rooms. - Humidity readings from DHT22 are less accurate in open areas. - DS18B20 sensor takes over 11 seconds to reach its stability.	This project aims to develop a automatic hydroponic nutrient mixing system which utilizes IoT cloud to provide remote user control and monitoring.
3.	Smart ecosystem for hydroponic	- Integration with smartphone interface.	- Dependency on internet connectivity. - Requires technical	This project aims to develop a smart ecosystem

	land	- Fuzzy logic for precise water circulation.	expertise. - Manual supervision is time consuming. - Limited remote monitoring.	for hydroponic land with trained personnel.
4.	Fuzzy-based automated nutrient solution control hydroponic tower system	- Precise control of pH and EC levels. - Real-time monitoring of sensor data. - Automated correction with treatment solutions. - Simplicity in setup and operation. - Less reliance on technical expertise	- Requires personnel to calibrate the sensors. - Limited precision in control. - Requires manual monitoring. - Not suitable for precise control.	The project aims to develop a user-friendly automated agriculture system with minimum effort on setup and operation.

Table 2.2.1.5.1 Summary of Existing System

2.3 Concluding Remark

To address the advantages and limitations identified in the previous studies, several proposed solutions have been formulated. First, for the "Hydroponic Nutrient Mixing System based on STM32," a comprehensive user training program and detailed documentation will be provided to ensure efficient system operation. Additionally, a filtration mechanism will be implemented to prevent sediment formation. Secondly, the "Automatic Hydroponic Nutrient Mixing System" will improve sensor accuracy in varying environments and enhance remote monitoring capabilities. For the "Smart Ecosystem for Hydroponic Land," efforts will focus on improving internet connectivity and offering specialized training to farmers, while introducing automation features to reduce manual supervision. Lastly, the "Fuzzy-based Automated Nutrient Solution Control for Hydroponic Tower System" will incorporate sensor calibration schedules, advanced control algorithms, and sophisticated remote monitoring to enhance precision, usability, and reliability. These solutions aim to enhance the efficiency, reliability, and user-friendliness of automated hydroponic nutrient mixing and management systems, ultimately contributing to the success and sustainability of hydroponic farming practices.

CHAPTER 3

System Methodology

3.1 System Development Models

In this section, several models will be outlined from the Software Development Life Cycle (SDLC) that will guide the development process of our project. Each model offers a structured approach to software development, ensuring systematic planning, implementation, testing, and maintenance. By selecting an appropriate model, we aim to streamline the project development process and achieve our objectives efficiently.

3.1.1 System Development Model 1: Waterfall Model

According to [14], the Waterfall Model, also known as linear-sequential life cycle model, it was the first introduced Process Model. It divides the development process into several distinct phases which are requirement analysis, system design, implementation, testing, deployment, and maintenance. In a waterfall model, each stage must be completed before proceeding to the next stage, with no overlap allowed.

One of the notable advantages of the Waterfall Model lies in its simplicity and ease of understanding and usage. It provides a straightforward framework for project management, with each phase having a specific deliverable and review process. This rigidity makes it easy to manage, particularly for smaller projects where requirements are well understood and clearly defined. Additionally, the model offers well-understood milestones and clearly defined stages, facilitating easy task arrangement and documentation of both the process and results.

However, despite its advantages, the Waterfall Model also has its limitations. One significant drawback is its lack of flexibility, as it does not allow for much reflection or revision once the development process is underway. Changes or updates to the requirements can be challenging to accommodate, leading to high amounts of risk and uncertainty, particularly in complex or object-oriented projects. Furthermore, the model is not suitable for long or ongoing projects, and integrating all components as a "big-bang" at the very end can present challenges in identifying technological or business bottlenecks early on.

In summary, the Waterfall Model is best utilized when project requirements are well known, clear, and fixed, and when there is minimal risk of changes or uncertainties. It offers a structured approach to development, ensuring systematic progression through each phase of the project life cycle. However, it is important to recognize its limitations and consider alternative models for projects with evolving requirements or high levels of uncertainty.

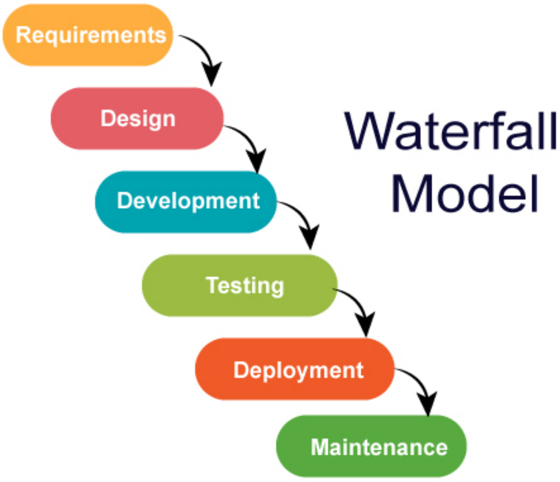


Figure 3.1.1.1 Waterfall Model

3.1.2 System Development Model 2: Iterative Model

The Iterative Model plays a crucial role in the Software Development Life Cycle (SDLC) by breaking down large project development processes into manageable increments. According to [16], Iterative Model unlike traditional models that follow a linear approach, the iterative model emphasizes gradual refinement and enhancement of software functionality. It begins with a focus on simplifying initial implementation, gradually incorporating more complexity and features until the final system is complete. Each iteration consists of five key stages: planning, analysis and design, implementation, testing, and evaluation.

The iterative process follows a cyclic pattern, deviating from the rigid step-by-step approach of traditional models. Once the initial planning is complete, these phases are repeated multiple times, allowing for continuous improvement and iteration on the software. This iterative nature enables developers and testers to identify functional or design-related flaws early in the development process, thereby facilitating timely corrective measures. Additionally, the iterative model is well-suited for addressing changing project scopes or requirements, as it offers a cost-effective approach to incorporating modifications.

However, despite its advantages, the iterative model also has its drawbacks. Each phase of an iteration is rigid, with limited opportunity for overlaps or concurrent activities. Moreover, system architecture or design issues may arise due to incomplete gathering of requirements at the beginning of the life cycle, potentially necessitating additional resources for resolution. Despite these challenges, the iterative model remains a valuable approach for software development, offering flexibility, adaptability, and continuous improvement throughout the development process.

In conclusion, the iterative model stands as a valuable approach within the Software Development Life Cycle (SDLC), offering flexibility, adaptability, and continuous improvement. By breaking down projects into manageable increments and emphasizing gradual refinement, it enables early detection of flaws and cost-effective adaptation to changing requirements. While it may present challenges such as rigid phase boundaries and potential resource implications, its benefits in fostering efficient development and accommodating evolving project needs underscore its importance in modern software development practices.

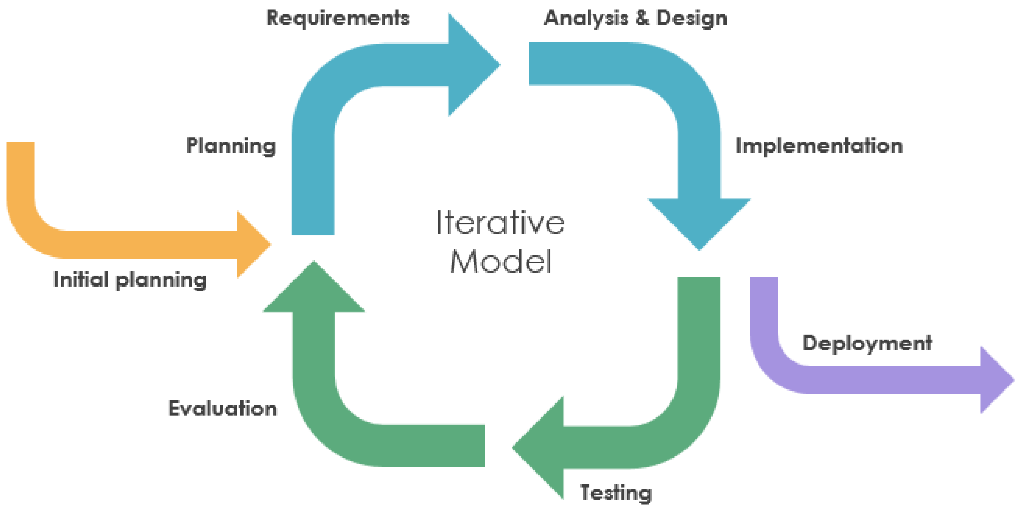


Figure 3.1.2.1 Iterative Model

3.1.3 System Development Model 3: Spiral Model

The Spiral Model stands out as a vital Software Development Life Cycle (SDLC) framework, seamlessly integrating elements of the waterfall and iterative models while prioritizing risk management [18]. This model embodies a systematic and iterative approach to SDLC, depicted as a spiral with adjustable loops that accommodate varying project demands. Each loop, representing a phase, encompasses a comprehensive cycle of software development activities.

Central to the Spiral Model is the role of the project manager, who dynamically adjusts the number of phases based on ongoing risk evaluations. This dynamic nature underscores a holistic progression through key development stages, including requirements gathering, analysis, design, implementation, testing, and maintenance.

The Spiral Model's emphasis on risk management is pivotal, following a risk-driven approach across its phases: planning, risk analysis, engineering, and evaluation. During planning, the team defines project scope and outlines future iterations. Subsequently, in the risk analysis phase, project risks are identified and assessed to inform subsequent development efforts. The engineering phase entails software development aligned with previously gathered requirements. Finally, in the evaluation phase, the software's compliance with customer needs and quality standards is thoroughly assessed.

In essence, the Spiral Model's iterative nature facilitates adaptive planning and proactive risk mitigation throughout the software development journey, ensuring alignment with project objectives and stakeholder expectations.

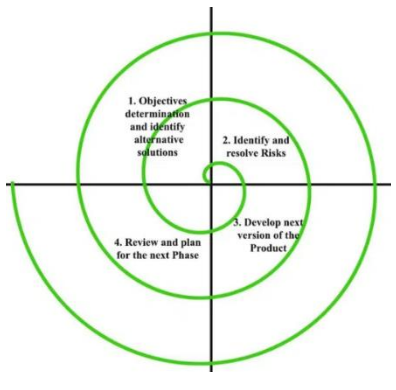


Figure 3.1.3.1 Spiral Model

3.1.4 System Development Model 4: Kanban Model

The Kanban Model is an agile approach utilized for progress tracking in system development. It adheres to a set of basic principles aimed at managing and improving workflow. Based on [18], these principles include "Start with What You Do Now", "Agree to Pursue Incremental, Evolutionary Change", "Respect the Current Process, Roles, and Responsibilities", and "Encourage Acts of Leadership at All Levels".

Implementing the Kanban Model in software development offers numerous benefits. Firstly, it aids in identifying bottlenecks and limiting work-in-progress, thereby enhancing efficiency. Secondly, its flexibility allows for easy adjustment of backlogs in collaboration with the development team, ensuring maximum value delivery to the organization.

Moreover, the Kanban Model promotes competitiveness in the market by optimizing the flow of work delivered to the customer. Additionally, it shortens project cycle time, enabling the development team to predict outcomes and delivery more accurately.

Visual metrics like control charts and cumulative flow diagrams provide valuable insights into workflow performance, aiding in process optimization and decision-making. The Kanban board serves as a visual representation of workflow, facilitating task visualization, bottleneck identification, and process streamlining.

In conclusion, leveraging the Kanban Model enhances workflow management by providing virtual Kanban boards accessible to entire teams, promoting collaboration, and optimizing lean processes.

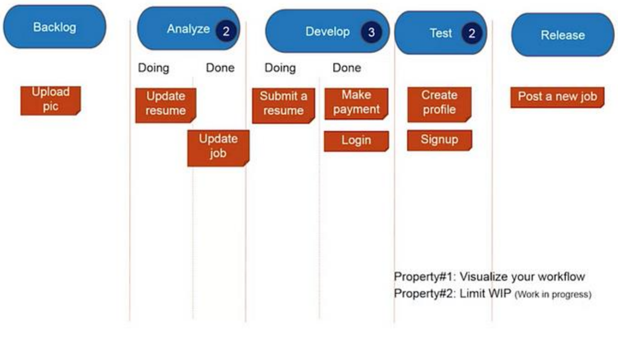


Figure 3.1.4.1 Kanban Model

3.1.5 Selected Model

After reviewing all models, the most suitable development model is the Iterative Model. Its flexibility allows for gradual improvements, adaptability to changing requirements, and continuous refinement, which ensures optimal performance in dynamic agricultural environments. This model's emphasis on risk management aligns well with the complexity of developing such a system, minimizing the likelihood of critical failures or malfunctions.

3.2 System Requirement

3.2.1 Hardware

The hardware involved in this project are computer, android mobile device, Raspberry Pi 3B+, sensors, camera. On the other hand, the software which will used are Raspberry Pi OS, Node-RED, InfluxDB, HiveMQ, Telegram, and Telegram's BotFather. A computer issued for the data processing, and data visualization. A mobile device is used for monitor and control the framework.

Laptop

Figure 3.2.1.1 MSI Cyborg 15 A12V

Description	Specifications
Model	MSI Cyborg 15 A12VF
Processor	Intel Core i7-12650H
Operating System	Windows 11
Graphic	NVIDIA GeForce RTX 4060 8GB DDR6
Memory (RAM)	32GB DDR5 RAM
Storage (ROM)	512GB PCIE SSD

Table 3.2.1.1 Specifications of laptop

Smartphone

Figure 3.2.1.2 OnePlus 8

Description	Specifications
Model	OnePlus 8
Processor	Snapdragon 865
Operating System	Android 13
Graphic	Adreno 650
Memory (RAM)	12GB RAM
Storage (ROM)	256GB

Table 3.2.1.2 Specifications of smartphone

Raspberry Pi 3B+

Figure 3.2.1.3 Raspberry Pi 3B+

Description	Specifications
Model	Raspberry Pi 3B+
Processor	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit
Operating System	Raspberry Pi OS (Bulleyes)
Graphic	Cortex-A72(ARM v8)
Memory (RAM)	1GB DDR2 SDRAM
Storage (ROM)	16GB Micro SD Card
Connectivity	- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac WLAN - Gigabit Ethernet (Maximum throughput 300Mbps)
Onboard Port	- 40 Pin GPIO Header - CSI Camera Port - 4 USB Port - 1 Auxiliary Port - 1 HDMI Port - 1 Micro USB Port - 1 Ethernet Port

Table 3.2.1.3 Specifications of Raspberry Pi 3B+

DHT22 Module

Figure 3.2.1.4 DHT22 Module

Description	Specifications
Input Voltage	3.3V – 6V
Sensor element	Snapdragon 865
Operating Range	Humidity: 0 to 100% RH Temperature: -40°C to 80°C
Accuracy	Humidity: +- 5% RH
Long-term stability	+- 0.5% RH/year

Table 3.2.1.4 Specifications of DHT22 Module

5MP Night Vision Pi Camera

Figure 3.2.1.5 5MP Night Vision Pi Camera

Description	Specifications
Photosensitive Module	OV5647
Resolution	5 Million Pixels
Dimension	25mm x 24mm
Support	Access to infrared light or fill light

Table 3.2.1.5 Specifications of 5MP Night Vision Pi Camera

<u>Water Level Sensor</u>

Figure 3.2.1.6 Water Level Sensor

Description	Specifications
Sensor Type	Analogue
Input Voltage	3.3V – 5V
Operating Current	<20mA
Detection Area	40mm x 16mm
Operating Temperature	10°C - 30°C

Table 3.2.1.6 Specifications of water level sensor

<u>8-Channel Relay Module</u>

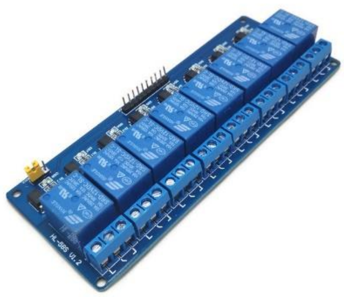


Figure 3.2.1.7 8-Channel Relay Module

Description	Specifications
Maximum Current	10A
Maximum Voltage	250V AC or 30V DC
Input Voltage	5V

Table 3.2.1.7 Specifications of 8-Channel Relay Module

Micro Submersible Water Pump

Figure 3.2.1.8 Micro Submersible Water Pump

Description	Specifications
Input Voltage	3.3V – 5V
Operating Current	0.1A – 0.2A
Size	45mm x 30mm x 25mm
Weight	30g
Operating Temperature	80°C

Table 3.2.1.8 Specifications of Micro Submersible Water Pump

MPC3008 Analogue to Digital Converter

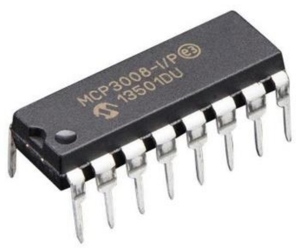


Figure 3.2.1.9 MPC3008 Analogue to Digital Converter

Description	Specifications
Operating Voltage	3.3V – 5V
ADC Channels	8
Interface	Serial Peripheral Interface (SPI)

Table 3.2.1.9 MPC3008 Analogue to Digital Converter

ACS712 Current Sensor Module

Figure 3.2.1.10 ACS712 Current Sensor Module

Description	Specifications
Supply Voltage	4.5V – 5.5V
Measure Current Range	-5A – 5A
Sensitivity	180mV/A – 190 mV/A
Dimension	31mm x 12mm

Table 3.2.10 Specifications of ACS712 Current Sensor Module

Outdoor Water Filtration System Water Filter Straw Purifier

Figure 3.2.1.11 Outdoor Water Filtration System Water Filter Straw Purifier

Description	Specifications
Material:	BPA-free ABS
Pores Size:	Min. 0.01 microns
Applied Temperature	32°F~90°F
Dimension	36mm * 148mm

Table 3.2.1.11 Outdoor Water Filtration System Water Filter Straw Purifier

<u>Mini DC Motor Water Pump RS-360SH</u>

Figure 3.2.1.12 Mini DC Motor Water Pump RS-360SH

Description	Specifications
Input Voltage	3V – 6V
Operating Current	0.3A – 0.5A
Size	42mm x 45mm x 65mm.
Weight	30g

Table 3.2.1.12 Mini DC Motor Water Pump RS-360SH

3.2.2 Software

<u>Raspberry Pi OS/ Raspbian</u>

Raspbian is a Linux-based operating system tailored for the Raspberry Pi. It features a user-friendly GUI that enables to configure settings, write code, and run script effortlessly. With a range of pre-installed software, it has offered stability and security in a compact package.

Figure 3.2.2.1 Raspberry Pi OS's user interface

<u>Node-RED</u>

Node-RED is a user-friendly, visual programming tool used for connecting various hardware devices, APIs, and web services. It simplifies the process by allowing users to create connections by connecting nodes together. Its browser-based editor and intuitive interface make it easy to design and deploy systems. Users can select from a wide-range of available nodes to build and implement their workflows efficiently.

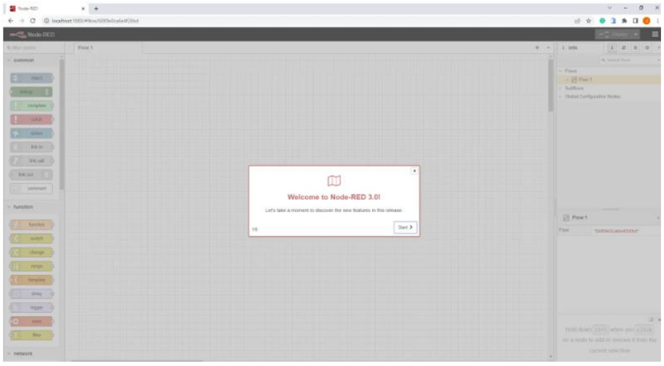


Figure 3.2.2.2 Node-RED's user interface

<u>HiveMQ</u>

HiveMQ is an MQTT broker, essentially a server that facilitates communication between devices. It enables users to both send (publish) and receive (subscribe) messages. When a device sends a message, HiveMQ ensures it reaches the intended recipient by directing messages from publishers to subscribers, making communication between devices smooth and efficient.

Figure 3.2.2.3 HiveMQ's logo

InfluxDB

InfluxDB is an open-source time-series database designed for storing and retrieving time-stamped data. It serves various purposes, including monitoring operations, collecting application metrics, gathering data from IoT sensors, and conducting real-time data analysis. It's a specialized database optimized for efficiently managing data that changes over time, making it ideal for time-series data applications.

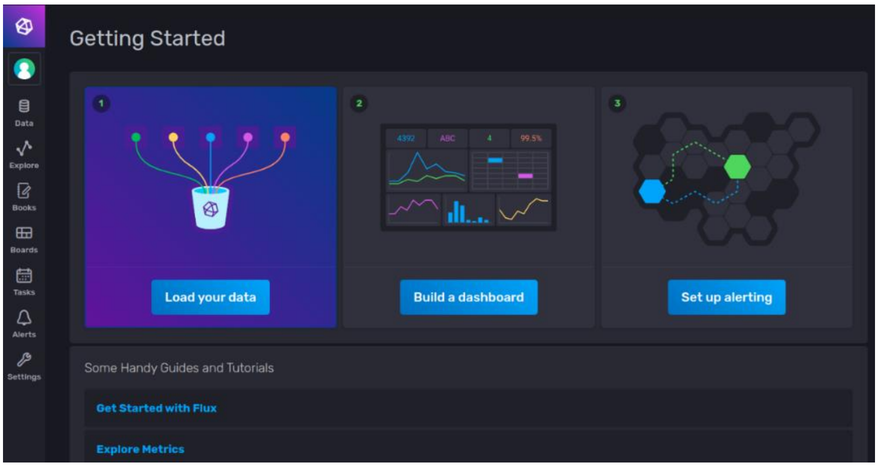


Figure 3.2.2.4 InfluxDB's user interface

Telegram

Telegram is a freemium, cloud-based instant messaging service available on multiple platforms. It prioritizes speed and security, offering a super-fast and secure way to communicate. It features a straightforward interface and is free for all users. Message sent through Telegram sync seamlessly across various devices, ensuring users can access their conversations from phones, tablets, or computers effortlessly.

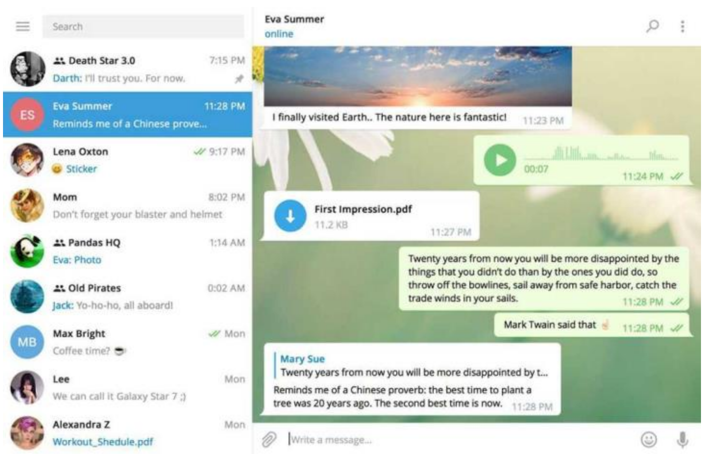


Figure 3.2.2.5 Telegram's user interface

Telegram's BotFather

Telegram's BotFather is a bot designed to help users to create and manage their own custom Telegram bots. It act as a bot "parent" or creator, allowing users to generate new bots by providing them with unique names and functions. BotFather also provides users with access tokens, which are like keys that allow the new bots to interact with the Telegram platform.

Figure 3.2.2.6 Telegram BotFather's logo

3.3 Functional Requirements

No.	Description
1	Raspberry Pi should collect the data from the Water Level sensor and publish through a MQTT every minutes.
2	Raspberry Pi should collect the data from the EC TDS sensor and publish through a MQTT every minutes.
3	Raspberry Pi should turn on and off the hydroponics pump when specific EC TDS value.
4	Raspberry Pi should turn on and off the water pump when the water level has exceeded or lower than specific value.
5	InfluxDB will store the real-time sensors data that receive from Raspberry Pi through MQTT.
6	The Node-RED dashboard should display all data that pulled from InfluxDB.

Table 3.3.1 Description for Functional Requirements

3.4 Project Milestones

FYP1

Task Description	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
FYP1														
Project Planning														
Collect information regarding the project														
Project Analysis & Design														
Define the project objective and scope														
Determine the hardware that will be used														
Project Implementation														
Order and receive sensors and actuators														
Connect Raspberry Pi with sensors and actuators														
Project Testing														
Create the flow of the system by using Node-RED														
Project Evaluation														
Publish the data to user device through MQTT														
Further Iterations														
Attach the hardware to the framework														
Run the system overnight in the lab														
Observe the system														

Table 3.4.1 Milestone of FYP1

FYP2

Task Description	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
FYP2														
Project Planning														
Review the weakness of the current system														
Project Analysis & Design														
Enhance current system														
Project Implementation														
Setup Filtration System														
Connect Raspberry Pi with filtration system														
Create the flow of the system by using Node-RED														
Publish the data to user device through MQTT														
Project Testing														
Attach the hardware to the framework														
Place the strawberries and lettuce in the framework														
Run the system overnight in the lab														
Project Evaluation														
Performance Evaluation														
Project Presentation														
Presentation Preparation														

Table 3.4.2 Milestone of FYP2

3.5 Estimated Cost

The following tables shows the hardware that purchase from the local market which is. The cost of the hardware for commercialization are dynamic since it is based on production supply and the discount campaign that held by the seller.

No.	Product Description	Quantity	Price (RM)	Total (RM)
1	Raspberry Pi Model 3B+	1	156.00	156.00
2	5MP Night Vision Camera for Raspberry Pi	1	60.00	60.00
3	DHT22 Sensor Module Breakout	2	16.00	32.00
4	Water Level Sensor	1	1.50	1.50
5	8 Channel Relay Module with Opto-Isolator (5V)	2	26.50	53.00
6	40 Ways Male to Male Jumper Wire	1	2.50	2.50
7	40 Ways Male to Female Jumper Wire	1	2.50	2.50
8	40 Ways Female to Female Jumper Wire	1	2.50	2.50
9	DC Jack (Female) to Screw Terminal Adapter	2	2.00	4.00
10	Adapter 12V 2A (UK Plug)	2	9.00	18.00
11	Breadboard 16.5 x 5.5cm (830 Holes)	2	4.00	8.00
12	DC 12V 2 Pin Desktop / CPU Cooling Fan	2	5.00	10.00
13	12V LED Grow Light	2	15.00	30.00
14	MCP3008 – Analogue to Digital Converter	1	24.50	24.50
15	ACS712 Current Sensor Module	2	9.50	19.00
16	Water Quality TDS Meter Sensor	1	58.00	58.00
17	DC 12V to 5V Converter	1	8.00	8.00
18	Micro Submersible Water Pump DC 3V-5V	5	4.00	20.00
19	RS350HS Water Pump	2	28.00	56.00
		Grand Total:		**568.50**

Table 3.5.1 Estimated cost of the project

3.6 Concluding Remark

In this chapter, several system development models have been reviewed, and one has been selected and implemented for this project. Additionally, an analysis of the hardware and software utilized in system development has been conducted. Functional requirements have been explicitly outlined to ensure the project's development trajectory remains on course. Moreover, project milestones for FYP1 and FYP2 have been delineated, distinguishing tasks and their respective execution timelines. Lastly, a clear statement of estimated costs for purchasing the hardware used in this project has been provided in a tabular format.

CHAPTER 4

System Design

4.1 System Architecture

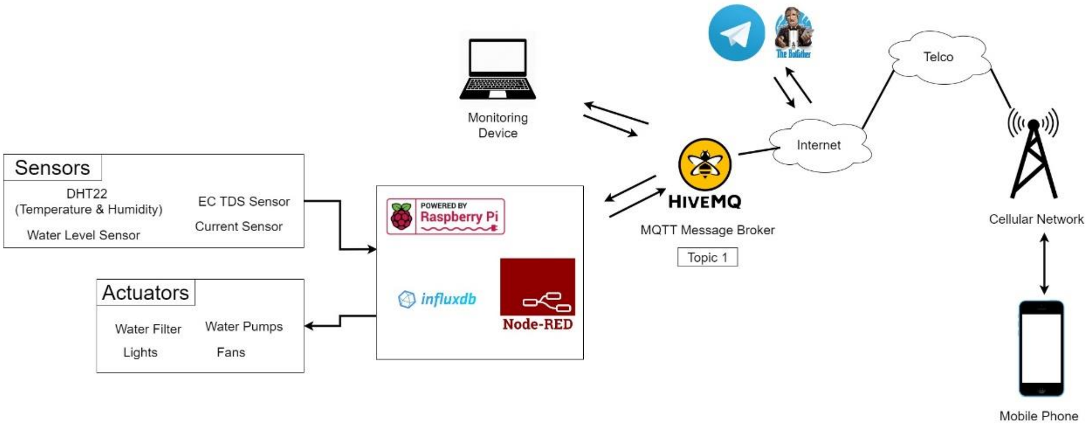


Figure 4.1.1 Architecture diagram of the project

In this project, the Raspberry Pi serves as the central controller, orchestrating the entire system, while Node-RED functions as the backend software, facilitating seamless system operations. Additionally, several sensors are connected to the Raspberry Pi to gather environmental data. This data is then transmitted to the HiveMQ Message Broker via MQTT topics.

On the receiving end, a monitoring device equipped with Node-RED retrieves data from HiveMQ through MQTT topics. The acquired data is subsequently stored in a time-series database, specifically InfluxDB. Once stored, the Node-RED dashboard retrieves and presents the data to users. Moreover, users can utilize the dashboard to control actuators connected to the Raspberry Pi. User-initiated actions are relayed to the Raspberry Pi through HiveMQ MQTT topics.

Lastly, a Telegram chatbot is developed using BotFather within the Telegram application. This chatbot is configured to respond to preset commands and offers functionalities such as checking environmental conditions, automating controls, and sending alert messages to users.

4.2 Functional Modules in the System

1. **Module 1 (Data Collection):**
    a. Sensor data, including temperature, humidity, EC TDS, water level, and current level, is collected and transmitted to the monitoring device via MQTT. This data is then stored in InfluxDB for further analysis.
2. **Module 2 (Live Stream):**
    a. A Raspberry Pi Camera is integrated with the Raspberry Pi. Utilizing the raspivid command, the Pi Camera records video feed, which is streamed lieve to YouTube through FFmpeg.
3. **Module 3 (Automation Control):**
    a. Automation control encompasses various actuators. The grow light operates between sunset and sunrise daily to mimic sunlight. The exhaust fan and cooling fan activates when the ambient temperature exceeds 25 Celsius to maintain optimal environment conditions. The clean water pump draws water from the clean water tank and pump it into the mixed solution tank, while pumps for Hydroponics solution A and solution B distribute the hydroponics into the mixed solution tanks. Once the EC TDS value in the mixed solution tank reaches the desired range for Lettuce and Strawberry, the mixed solution is pumped to the respective plants.
4. **Module 4 (Data Visualization):**
    a. Node-RED is configured to retrieve sensors data from InfluxDB and display it on the Node-RED dashboard, facilitating real-time monitoring and analysis.
5. **Module 5 (Telegram Chatbot):**
    a. The Telegram chatbot interacts with users based on their commands. Functionality includes retrieving environmental conditions, enabling automation within the framework, and capturing snapshots of the environment. Additionally, users receive alert messages when uncommon data is detected, ensuring proactive management of system anomalies.

4.3 System Flow

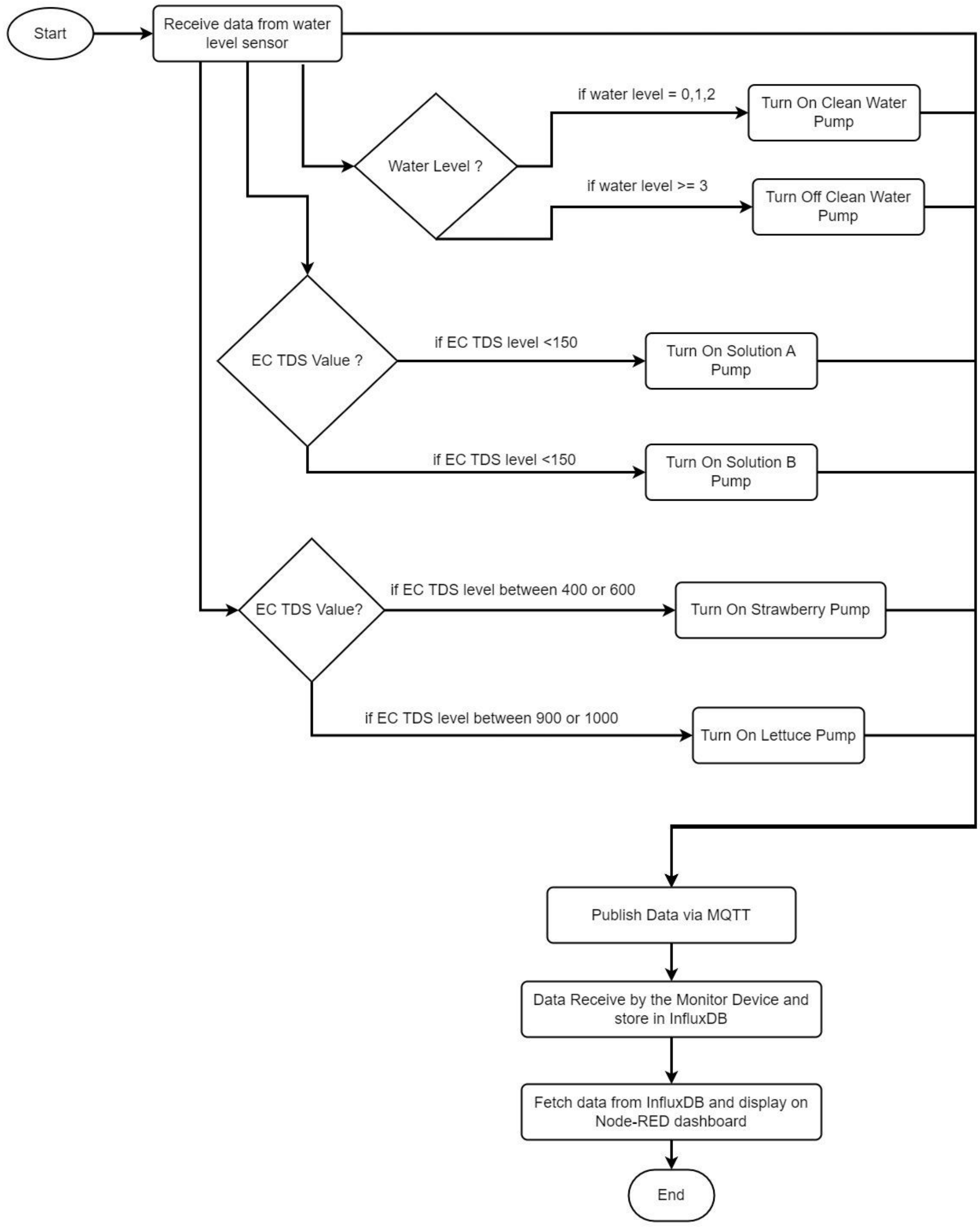


Figure 4.3.1 System flowchart of the monitoring framework

Once the system initializes, the Raspberry Pi diligently commences its task of gathering vital data from the interconnected sensors, serving as the eyes and ears of the hydroponic setup. The water level sensor dutifully reports the current reservoir status, while the EC/TDS sensor meticulously measures the nutrient concentration in the solution.

Armed with this real-time data, the Raspberry Pi embarks on a journey of decision-making, navigating through three pivotal switch cases to orchestrate the system's response. Firstly, it evaluates the water level, determining whether intervention is warranted. If the water level registers at 0, signaling an empty reservoir, the Clean Water Pump springs into action, replenishing the system's lifeblood. Conversely, if the water level stands at 3 or above, the Clean Water Pump remains dormant, safeguarding against overflow.

Secondly, attention turns to the EC/TDS readings, the barometer of nutrient concentration. Should the value dip below 150, indicating a nutrient deficiency, the Solution A and Solution B pumps unite, infusing the solution with essential nutrients. As the nutrient concentration climbs to the optimal range, between 400 and 600, both Solution A and B pumps gracefully retire, while the Lettuce Pump steps into the spotlight, tailored to the specific needs of lettuce plants. Yet, should the EC/TDS readings soar to the 900 to 1000 range, signaling excessive nutrient buildup, the Solution A and B pumps bid adieu, making way for the Strawberry Pump to assume control, delivering precisely calibrated nourishment for strawberries.

In parallel, the system ensures seamless data dissemination and storage, ensuring transparency and accessibility for users. Through the MQTT protocol, all sensor data is promptly broadcasted to the designated subscriber. This subscriber, equipped with the prowess of InfluxDB, faithfully captures and stores this wealth of information, laying the foundation for informed decision-making and trend analysis.

But the journey does not end there. With the aid of Node-RED's intuitive dashboard, users are granted a window into the hydroponic realm, where sensor data dances across the screen, providing a real-time snapshot of the system's health and performance. Armed with this insight, users can navigate the complexities of hydroponic cultivation with confidence, empowered by the fusion of technology and nature.

4.4 GUI Design

In the realm of user interface design, the Node-RED dashboard emerges as the cornerstone of the monitoring framework, offering a user-friendly platform for visualizing and managing crucial data collected from sensors. This intuitive dashboard seamlessly pulls data from InfluxDB, presenting users with real-time insights into environmental conditions.

Figure 4.4.1 Node-RED dashboard for Farm condition.

Figure 4.4.1 showcases the Node-RED dashboard tailored for monitoring farm conditions. Here, users can effortlessly track metrics such as EC TDS value and water level, enabling informed decision-making regarding nutrient management and reservoir maintenance. Additionally, the synchronization of actuator controls ensures users remain abreast of the operational status of fans, grow lights, and water pumps, empowering them to fine-tune the hydroponic environment to meet the needs of their plants. A central main switch provides a convenient means of enabling or disabling the automated system, granting users ultimate control over system functionality.

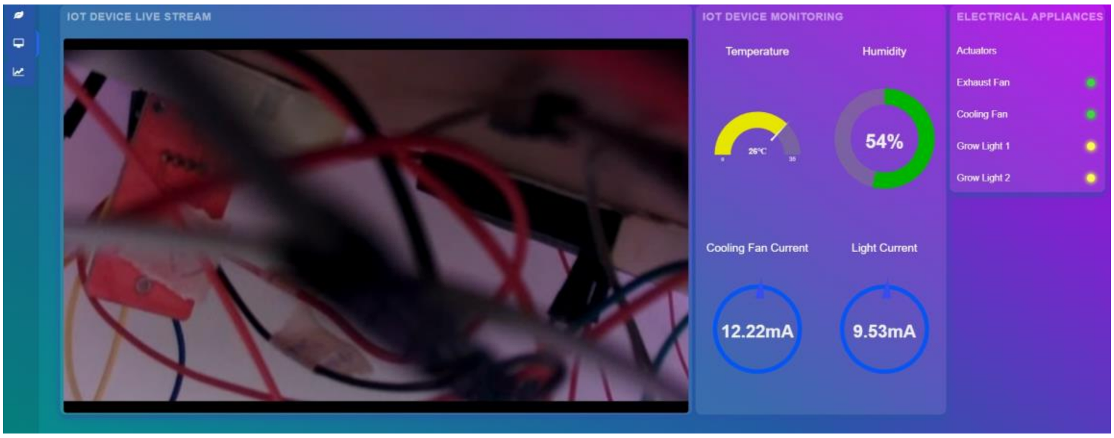


Figure 4.4.2 Node-RED dashboard for IoT devices condition.

Expanding beyond the confines of the hydroponic setup, Figure 4.4.2 unveils an IoT Device Monitoring dashboard, offering users insights into the health and functionality of sensors placed around IoT devices. This comprehensive interface allows users to monitor the running status of these devices, ensuring seamless operation and preempting potential issues before they escalate.

Figure 4.4.3 Node-RED dashboard for Farm and IoT Devices Line Charts & Tables

Additionally, a standalone dashboard has been developed to provide real-time monitoring of both farm and IoT device data. This dashboard features interactive line charts and tables, displaying key metrics such as farm temperature, humidity, EC/TDS levels, and water levels. It also tracks the environmental temperature and humidity of IoT devices, along with power usage data for the light and fan currents. This allows for comprehensive monitoring and efficient management of both the farm environment and connected devices.

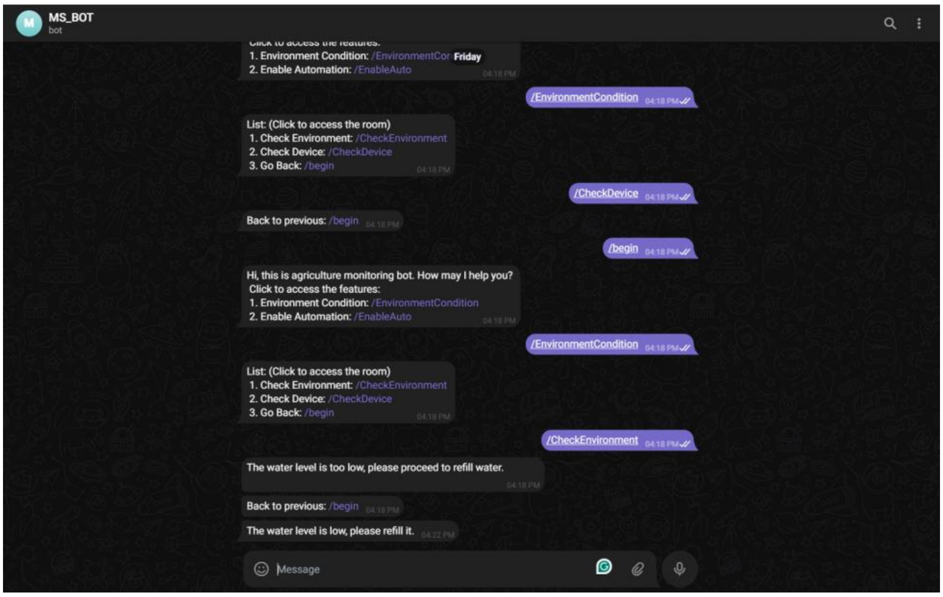


Figure 4.4.4 Telegram Chatbot Interface.

In addition to the Node-RED dashboard, the integration of a Telegram Chatbot further enhances the user experience, offering a streamlined interface for accessing key functionalities. Configured with an intuitive menu system, users can effortlessly execute commands such as checking environment conditions, enabling automation, or capturing photos with a simple click. This seamless integration of chatbot technology simplifies user interaction, facilitating efficient monitoring and management of the hydroponic system.

Through these innovative interfaces, users are empowered with unprecedented visibility and control over their hydroponic setup, fostering a seamless integration of technology and agriculture for optimal plant growth and cultivation.

4.5 Concluding Remark

This chapter delves into the comprehensive architecture of the hydroponic system, offering readers a detailed understanding of its inner workings. The discussion begins with an elucidation of the hardware connections, elucidating the intricate web of sensors and actuators that form the backbone of the system. Each functional module is meticulously dissected, illuminating the unique role it plays in optimizing plant growth and cultivation.

Furthermore, the system flow chart serves as a roadmap, guiding readers through the intricate dance of data collection, processing, and action. This visual representation offers a clear and concise overview of the system's operation, elucidating the decision-making process that drives optimal plant care.

In addition to dissecting the hardware and functional modules, this chapter provides a glimpse into the user interface, showcasing the intuitive Node-RED dashboard and Telegram Chatbot. These graphical interfaces serve as windows into the hydroponic world, offering users unprecedented visibility and control over their cultivation environment. From monitoring environmental conditions to managing actuators with a simple click, these interfaces elevate the user experience, fostering seamless interaction with the system.

By weaving together hardware connections, functional modules, system flow, and graphical interfaces, this chapter offers a holistic understanding of the hydroponic system architecture. Armed with this knowledge, readers are equipped to harness the power of technology to cultivate thriving, bountiful crops with ease and precision.

CHAPTER 5

System Implementation

5.1 Hardware Setup

This project will be focusing on building a prototype agriculture framework by using Raspberry Pi, some micro submersible water pumps, water filter, relays and other modules.

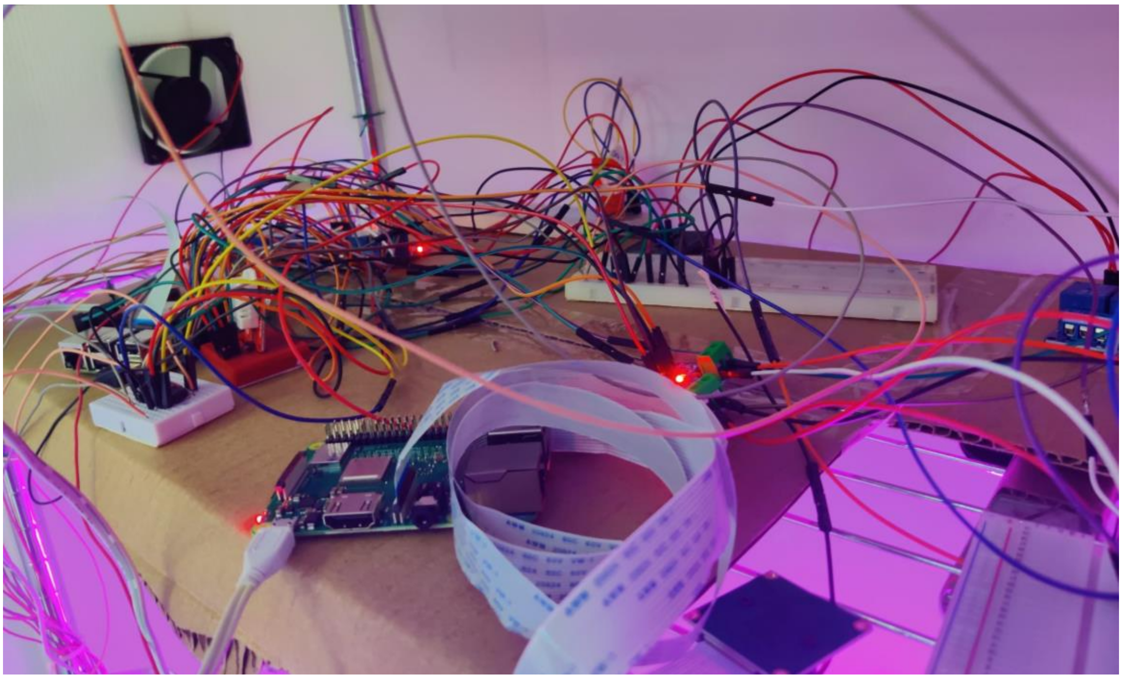


Figure 5.1.1 Overview of the whole structure

Based on the figure above, the Raspberry Pi is fully connected with a bunch of sensors and actuators such as water level sensor, EC sensor, water pumps, etc.

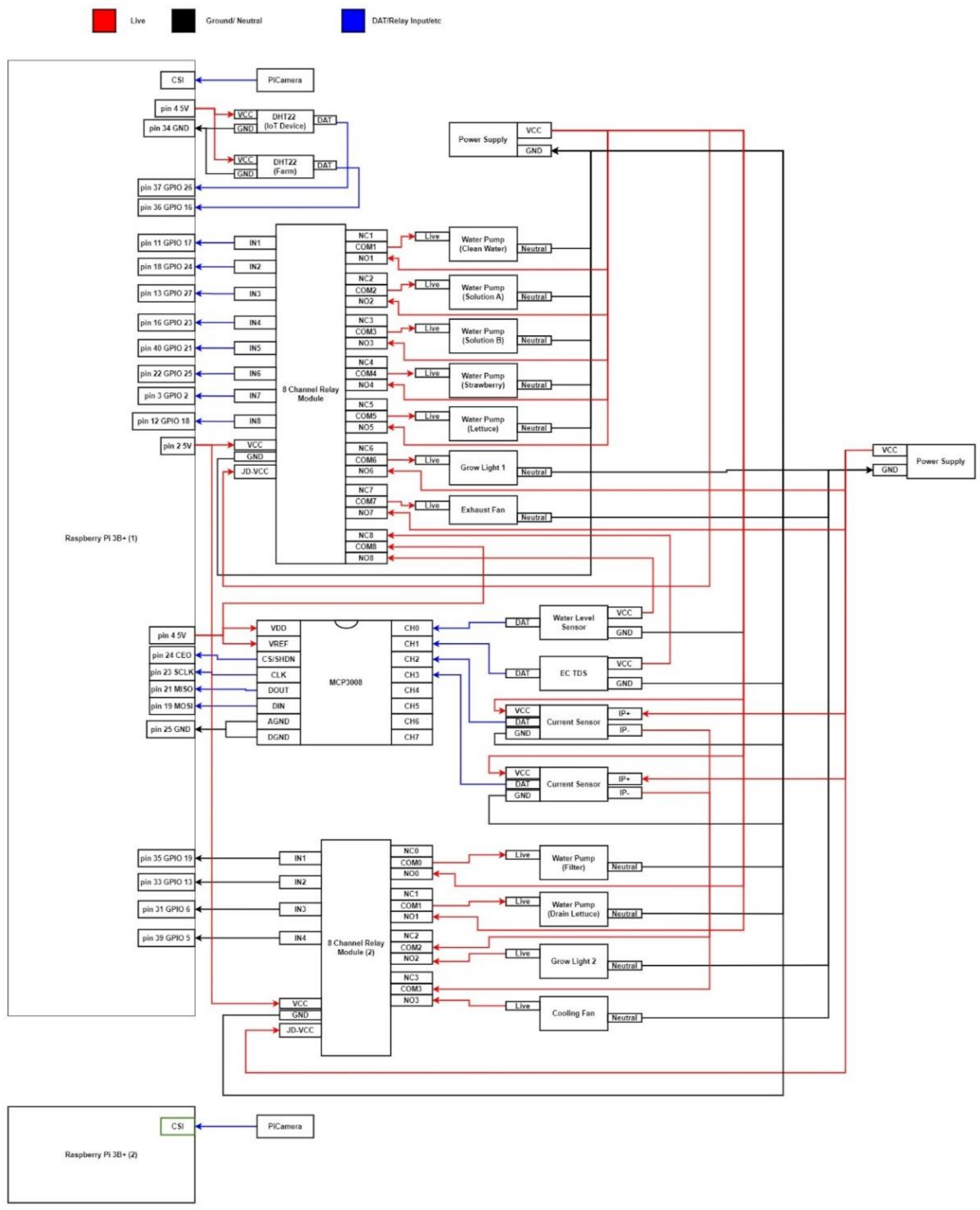


Figure 5.1.2 Detailed hardware connections between Raspberry Pi, Sensors, Actuators

Figure 5.1.2 provides a comprehensive overview of the hardware connections between the Raspberry Pi and the various sensors and actuators used in the system. These connections are designed to facilitate seamless communication and control, enabling the Raspberry Pi to collect data from the sensors and actuate the corresponding devices as needed.

Firstly, there are two DHT22 sensors connected to the Raspberry Pi, which responsible for collecting the temperature and humidity of the devices and the farming environment.

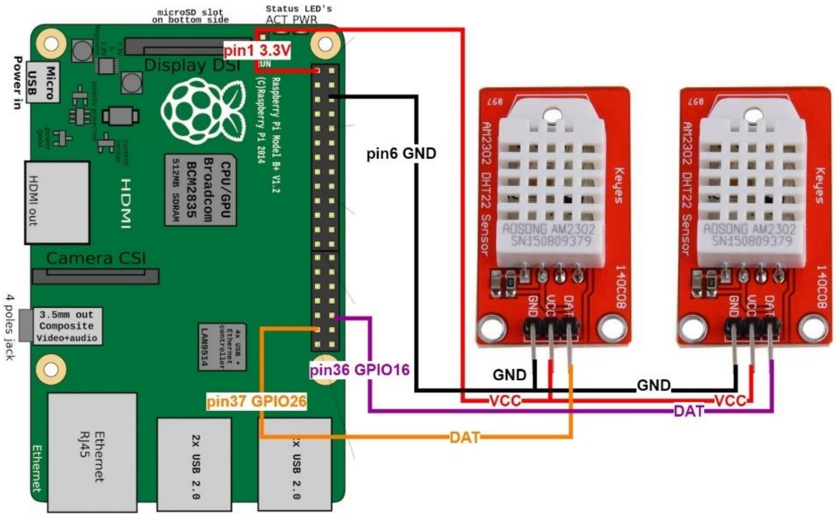


Figure 5.1.3 The connection between Raspberry Pi and two DHT22

Secondly, the Raspberry Pi Camera Module will be attached to another Raspberry Pi. The second raspberry pi will be focusing on hosting the YouTube Live Stream, which provide users with live feeds on monitoring the crops.

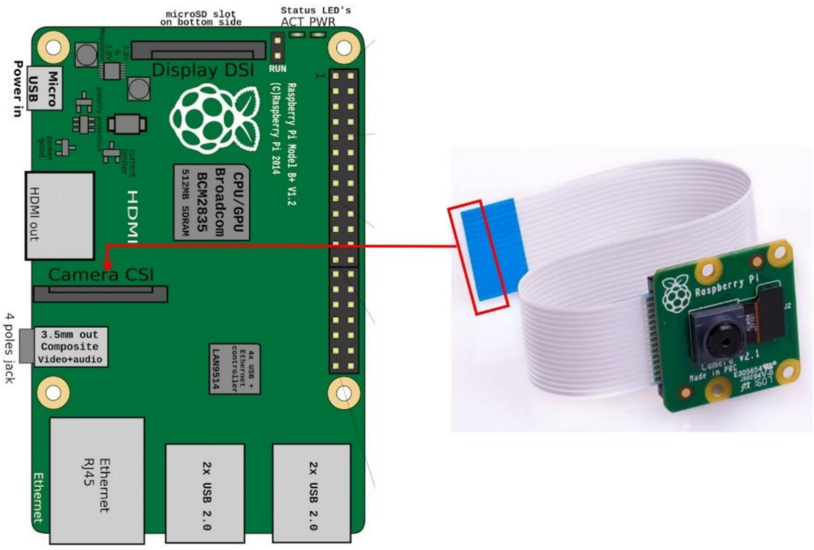


Figure 5.1.4 The connection between Raspberry Pi and the PiCam module

Moving on to the next part, the water level sensor, EC sensor, and the current sensors will be connected to the MCP3008 Analogue to Digital Converter respectively. The water level sensor, which suggested by its name, it responsible for the water level readings, while the EC sensor will be collecting the electrical conductivity of the water, and the remaining current sensors will be collecting the current readings from both growing light and fan. The MCP3008 module will be connected to the Raspberry Pi through pin 1, pin 6, pin 2, pin 23, pin 21, pin 19, and pin 24.

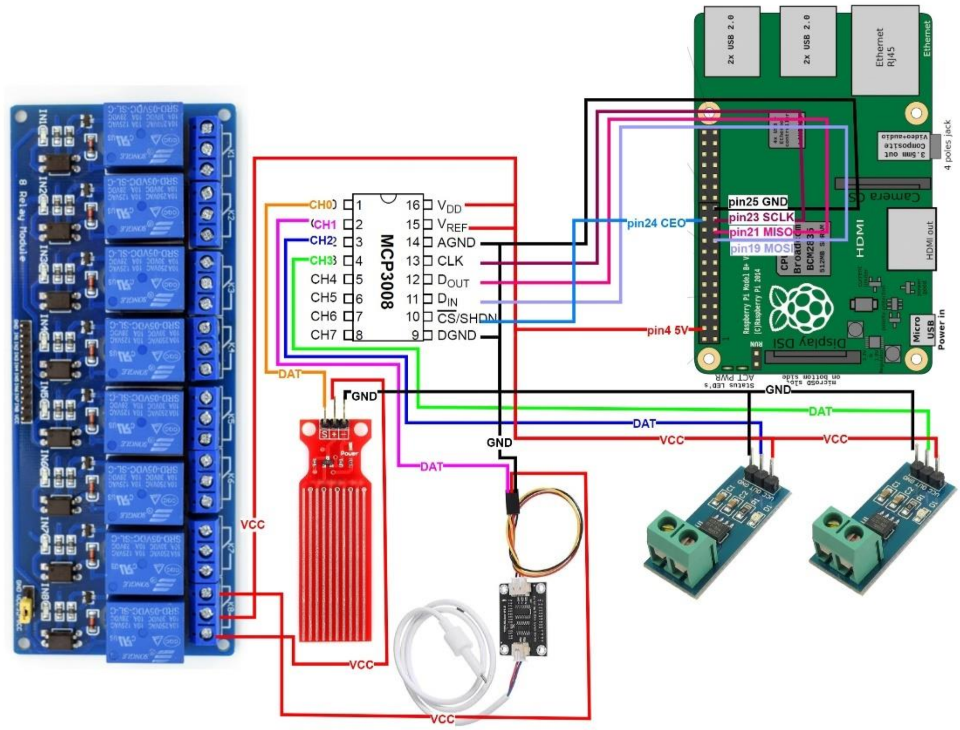


Figure 5.1.5 The connection between Raspberry Pi, MCP3008 and other sensors

Furthermore, there are two unit of 8-channel relay modules involve in the project which were connected with some water pumps, water valve, etc. For the first relay, there will be having one water level sensor, one EC sensor, and five water pumps connected on this relay. When the system starts, the relay will first switch on the water level sensor to determine whether there is any water in the water tank. When the water level reach at its preset level, it will switch off the water level sensor, and switch on the EC sensor, which will be starting to read water's electrical conductivity. If the readings does not meet the preset settings for the strawberry or lettuce, both water pumps that connected to the hydroponics solution a and b will be starting to pump in the solution until it meet the settings. Once the mixed solution has reached the settings, it will be pumped into the respective tanks to supply the nutrients to the crops.

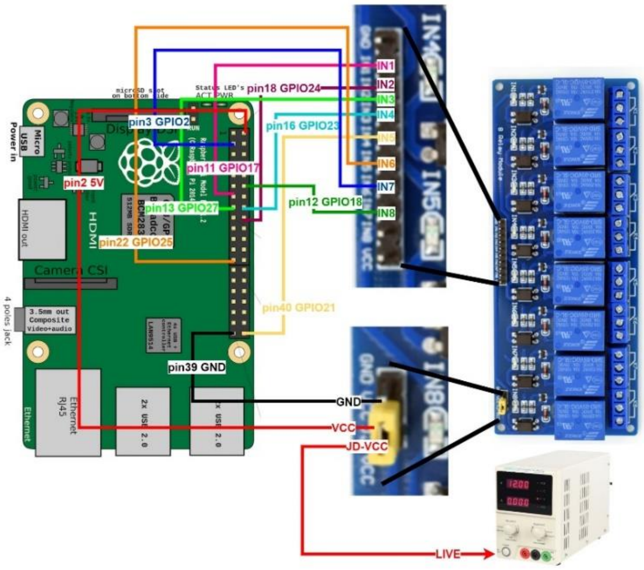


Figure 5.1.6 The connection between Raspberry Pi and the 8-channel relay1

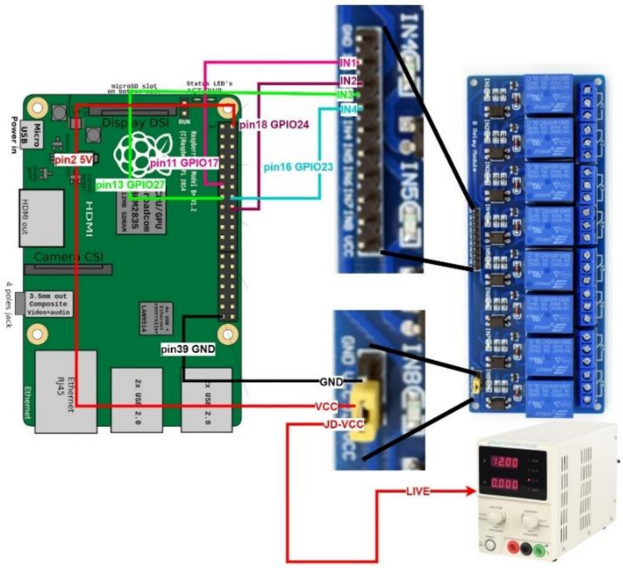


Figure 5.1.7 The connection between Raspberry Pi and the 8-channel relay2

Lastly, once the mixed solution has been fully utilized, the water pump which attached to the tank will be switch on and pump out the solution into the waste tank. In this part, there will be having a water filter which will be responsible for filtering the solution and make the water to be reusable.

5.2 Software Setup

5.2.1 Software Setup for Raspberry Pi

Before booting up the Raspberry Pi, the users are required to install the Raspberry Pi OS (Raspbian) into a micro SD card, and attach it to the Raspberry Pi. The Raspberry Pi OS can be easily installed by using the Raspberry Pi Imager which also can be easily get from the Raspberry Pi official website, **https://www.raspberrypi.com/software/**

Figure 5.2.1.1 Raspberry Pi Imager

Once the OS installation has been completed, the users are required to install the Node-RED on their devices. The installation can be done by following the instructions below:

1. Open the Terminal.
2. Enter the following command: **$bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/masters/deb/update-nodejs-and-nodered)**

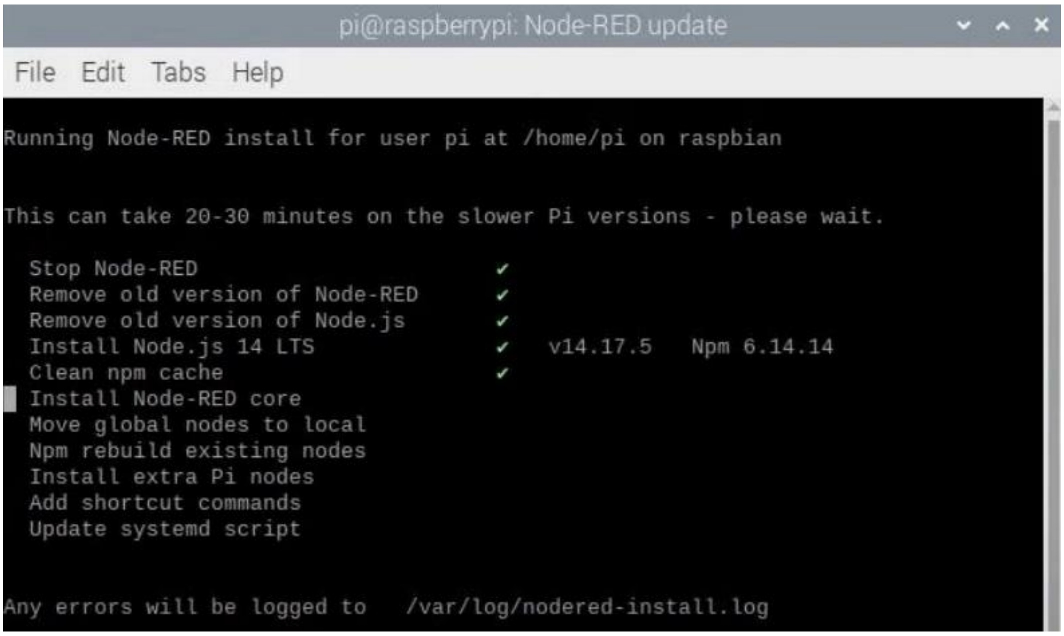


Figure 5.2.1.2 Terminal running Node-RED (Raspbian)

After the installation is done, the users will get to see the following output on their Terminal which showing that the Node-RED is ready to use.

Next, the users can continue to open up their Node-RED, with a simple search of “Node-RED” from the startup menu. After that, the users has to open up their browsers and enter the following ip address to access the Node-RED’s UI panel. **127.0.0.1:1880.**

After that, the users are required to install the palettes that are required for this project. To navigate to the palette installation panel, the users can click on the “Burger” button aka the three line button on their top left corner, and then click on the manage palette option. The users will be able to see the installed palette and manage the palette.

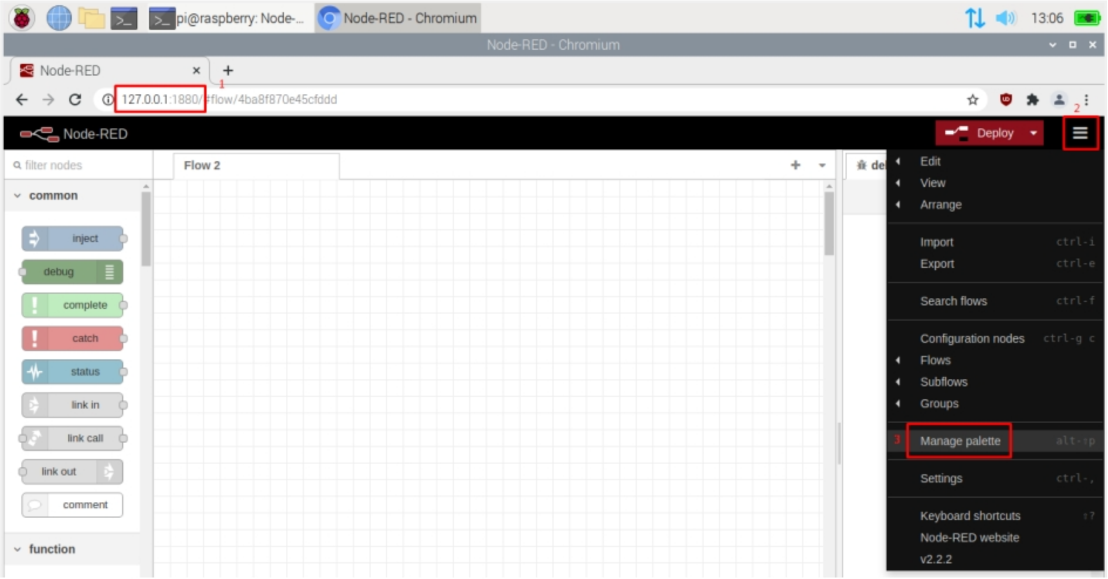


Figure 5.2.1.3 Node-RED Panel (Raspbian)

In order to ensure the project works perfectly, here are the palette that the users required to install on their devices.

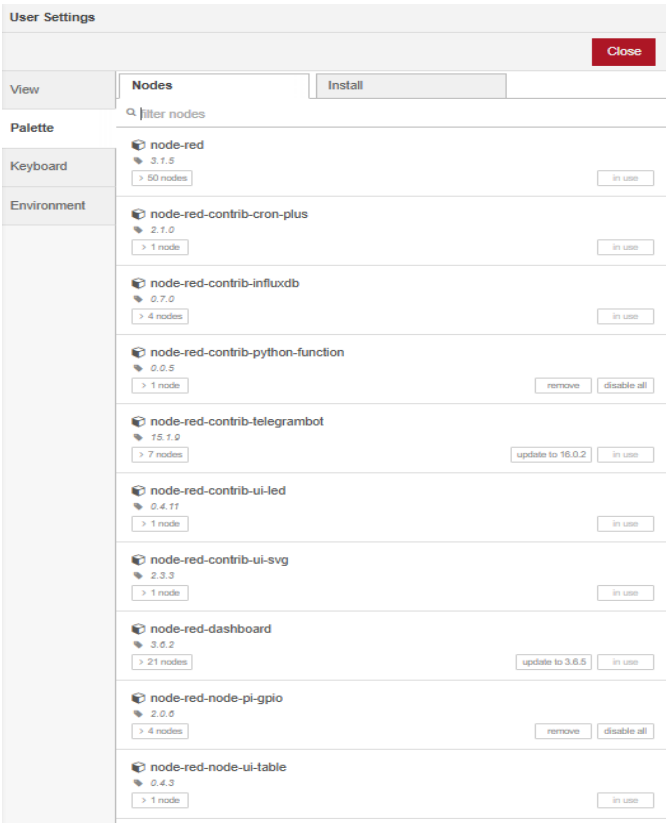


Figure 5.2.1.4 Installed palette in Node-RED on Raspbian

5.2.2 Software Setup for Laptop (Monitoring Device)

Next up, the software configuration on the laptop. In this project, the laptop will be acting as the monitoring device, it also required to installed with Node-RED for the monitoring purpose. Besides that, the real-time database, InfluxDB, is also required to be installed on the laptop for the data logging purpose.

Firstly, the users will need to install Node-RED on their laptop, the installation guide is almost the same as the Raspberry Pi does. Instead of just open up the Terminal, the users will need to download the nodeJS software from their official website, **https://nodejs.org/en/download/**. Once the users have been navigated to this page, they will get to see the four options on the page, and click on the Prebuilt Installer, then they will get to see the download button, and just download the recommended version which having a LTS wording on the version and install it on their laptops.

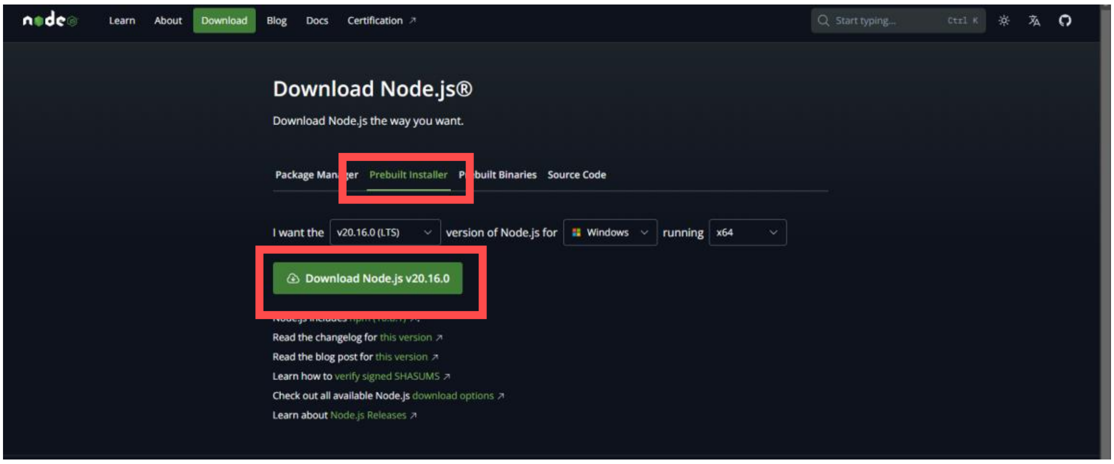


Figure 5.2.2.1 Node.js Official Website

After installing the NodeJS, the users have to open their CMD and enter the following command to install Node-RED, **npm install -g –unsafe-perm node-red**. After the installation, the users can easily start it by entering node-red in the cmd.

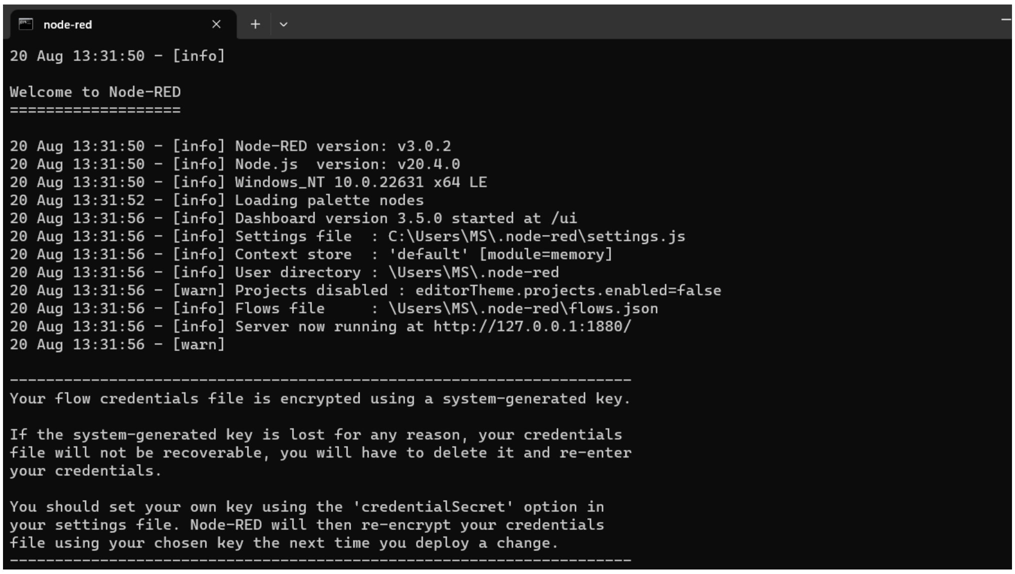


Figure 5.2.2.2 Node-RED installation on Windows Command Prompt

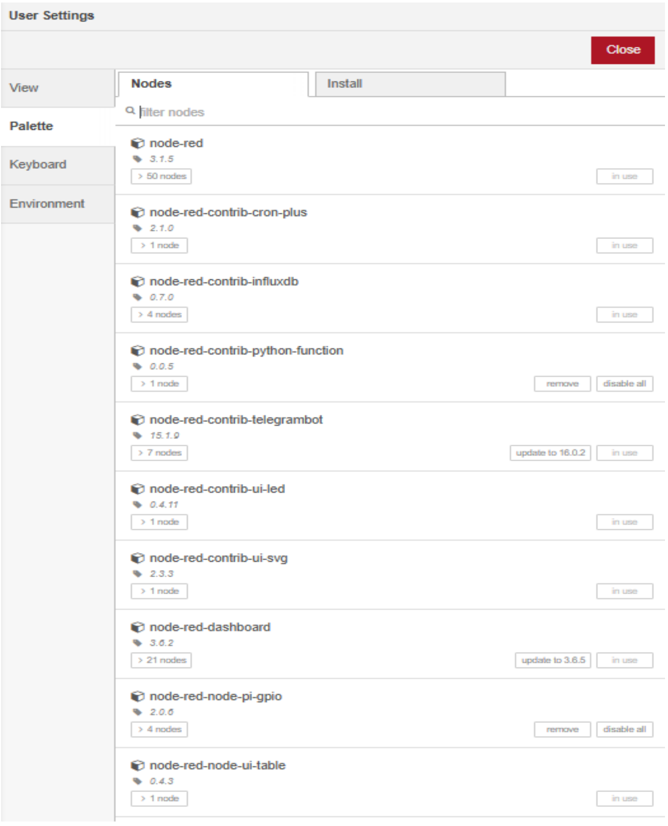


Figure 5.2.2.3 Installed palette in Node-RED on Windows

Moving on to the next part, the InfluxDB installation. The users can easily get the installation file by entering the following command into their command prompt or browser.

Command Prompt: **wget https://dl.influxdata.com/influxdb/releases/influxdb2-2.0.7-windows-amd64.zip**.

Any Browser: **https://dl.influxdata.com/influxdb/releases/influxdb2-2.0.7-windows-amd64.zip**

Figure 5.2.2.4 Download InfluxDB file using Windows Command Prompt

Once the file has been downloaded, the users will be required to redirect to the file location and extract the file. Then open up the file and click on the file path and enter 'cmd'. Once the Command Prompt has been opened, the user may continue with entering 'influxd' to start InfluxDB.

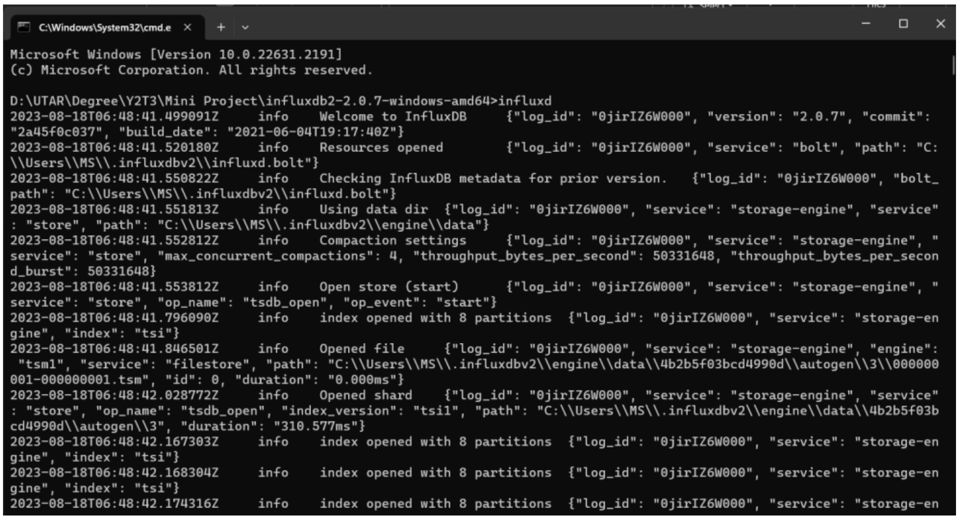


Figure 5.2.2.5 Command Prompt which running InfluxDB

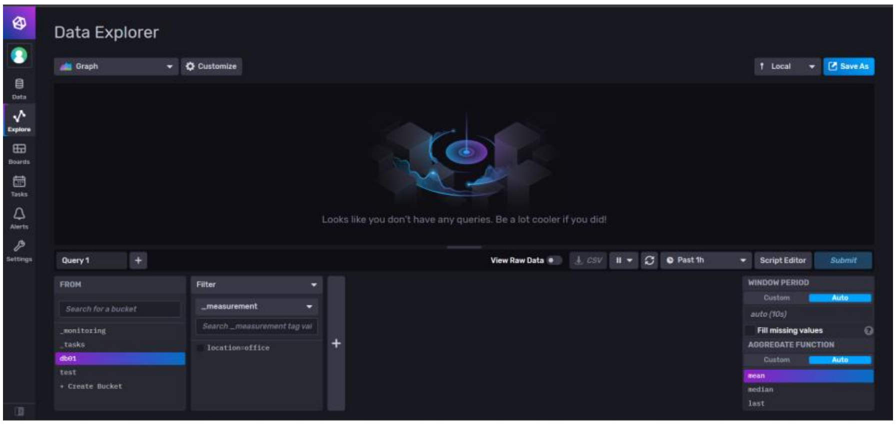


Figure 5.2.2.6 InfluxDB Panel

5.2.3 Software Setup for BotFather (Telegram)

To setup a Telegram Chatbot, the users will be required to own a Telegram account. Once they registered their account, they can continue with setting up their own Telegram ChatBot by using a feature by Telegram called BotFather. The users can easily get this feature by searching it with the Telegram seach bar.

Once the users have searched the BotFather, they can start a conversation by typing **/start**.

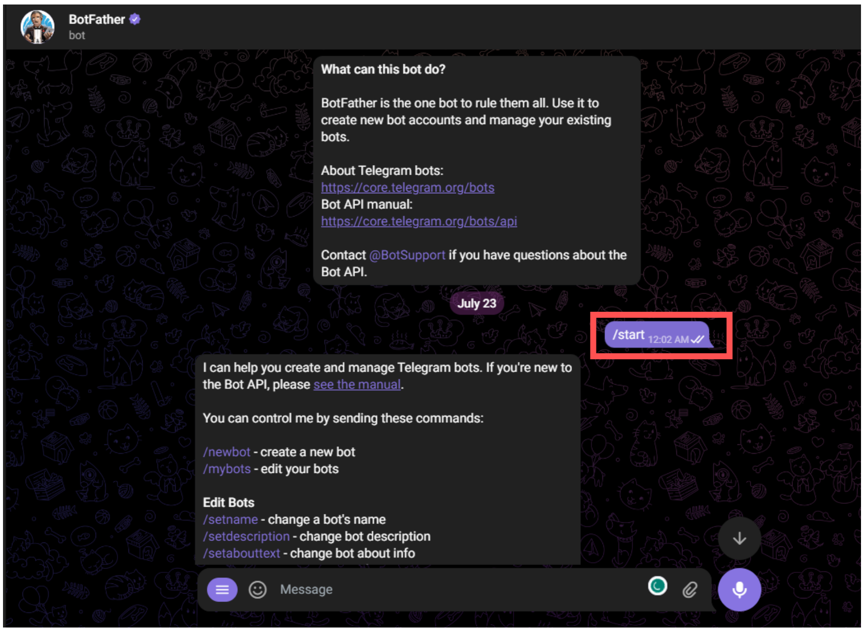


Figure 5.2.3.1 BotFather Setup

After that, the BotFather will show the users a bunch of instructions to create a ChatBot, starting with setting a name for the chatbot. If the users have entered any bot name's that had been used, the BotFather will inform the users to enter a new name for the bot, to ensure that there is no bots with the same name.

Once the ChatBot has been created, BotFather will generate an unique token for each chatbot, the users will be required to keep the token for further usage.

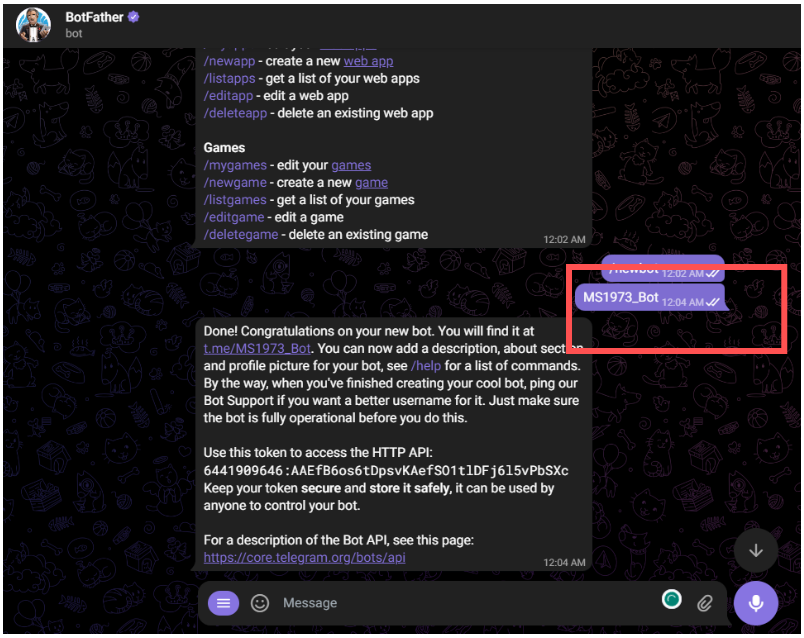


Figure 5.2.3.2 Process of setting up BotFather

5.3 Setting and Configuration

5.3.1 Setting and Configuring on Raspberry Pi

Before configuring the system, users must enable several features that are disabled by default, including the Camera, SSH, VNC, and SPI interfaces. Enabling the Camera interface allows the PiCam to perform live streaming. SSH and VNC interfaces are enabled so that users can remotely access the Raspberry Pi using the command prompt or UI control. The SPI interface is required for the Raspberry Pi to read data from the MCP3008.

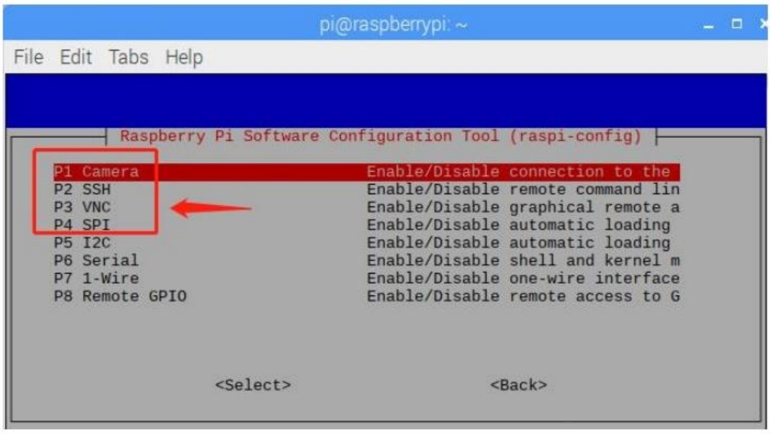


Figure 5.3.1.1 Raspberry Pi Interface Configuration Page

Once all the interfaces have been enabled, users can start programming the flow for the agriculture monitoring framework using Node-RED. First, users need to create a flow to collect temperature and humidity data from both DHT22 sensors—one placed near the IoT devices and the other at the farm.

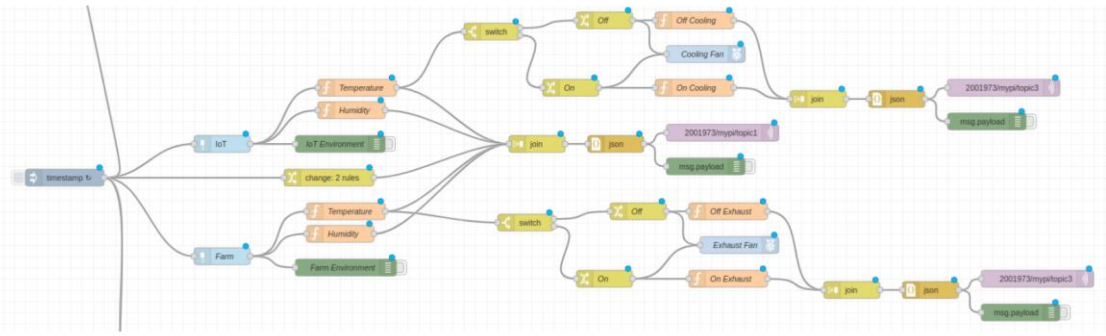


Figure 5.3.1.2 Programming flow for both DHT22 sensors, Exhaust Fan, and Cooling Fan

Since there are two DHT22 sensors, each connected to a different pin, users need to configure the pin numbers according to their respective connections, as shown previously. Additionally, users must configure the function nodes connected to the rpi-dht22 node to ensure that the system accurately collects temperature and humidity values from the sensors.

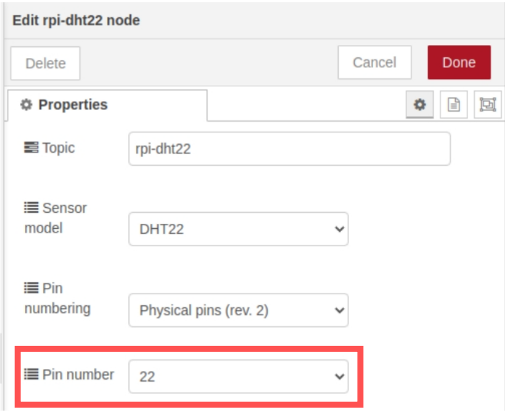


Figure 5.3.1.3 Configuration of the rpi-dht22 node

Besides that, the users are also needed to configure the commands that shown below for the function nodes that are connected to each rpi-dht22 node to separate the readings into temperature and humidity.

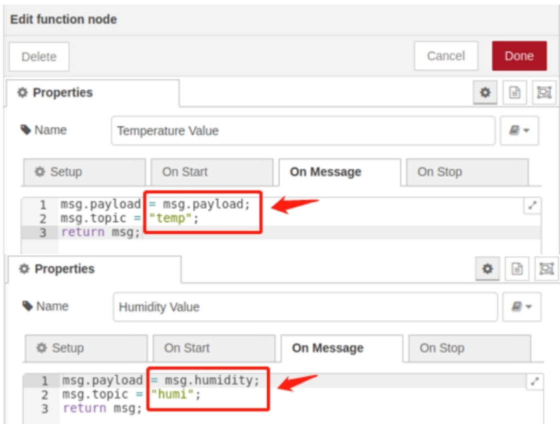


Figure 5.3.1.4 Configuration of the function nodes for temperature and humidity

Secondly, the users are needed to configure both exhaust and cooling fans' rpi-gpio out nodes with their respective pins by following the figures shown below.

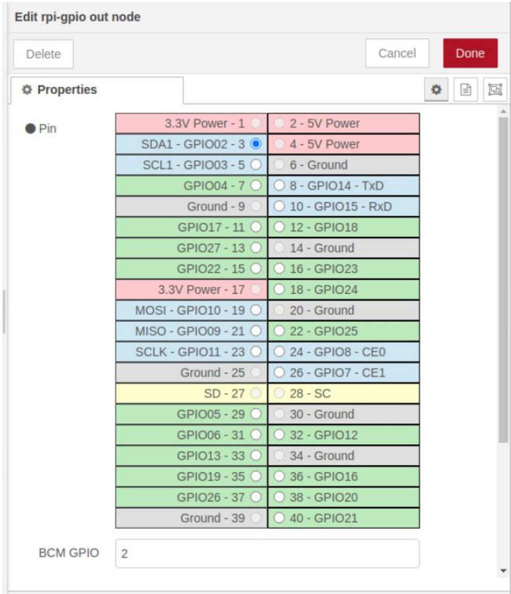


Figure 5.3.1.5 Configuration on both Exhaust & Cooling Fans rpi-gpio out node

Apart from that, users need to configure the topic and server settings for each MQTT out node to ensure data transmission to the monitoring device. By following the figures shown below, users will be able to transmit the data to the monitoring device once both sides are properly configured.

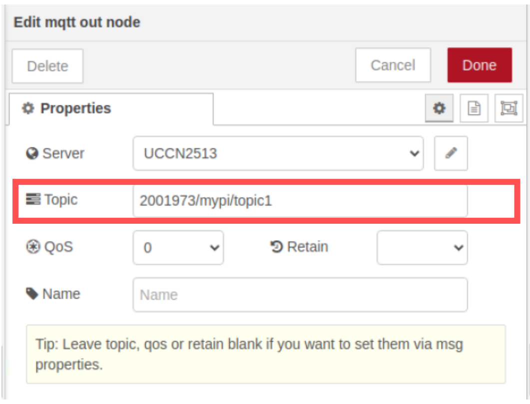


Figure 5.3.1.6 Configuration on mqtt out node (topic)

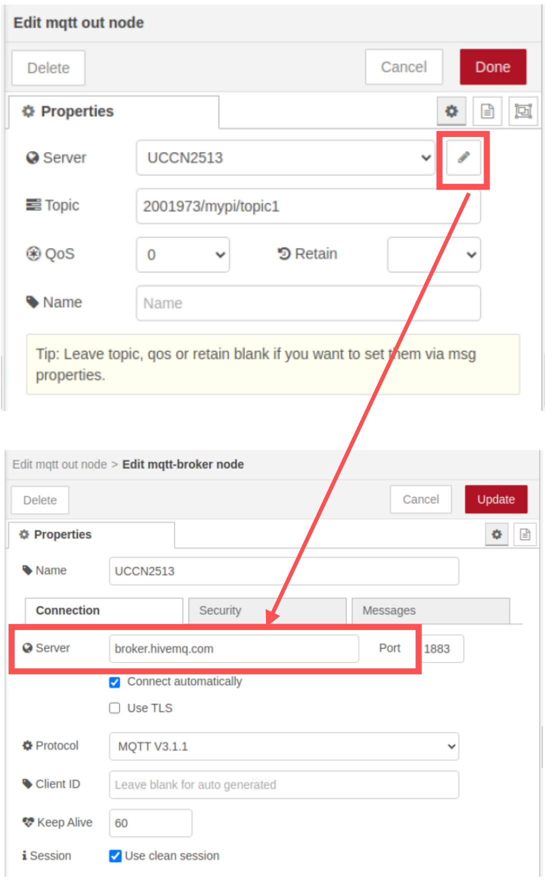


Figure 5.3.1.7 Configuration on mqtt out node (server)

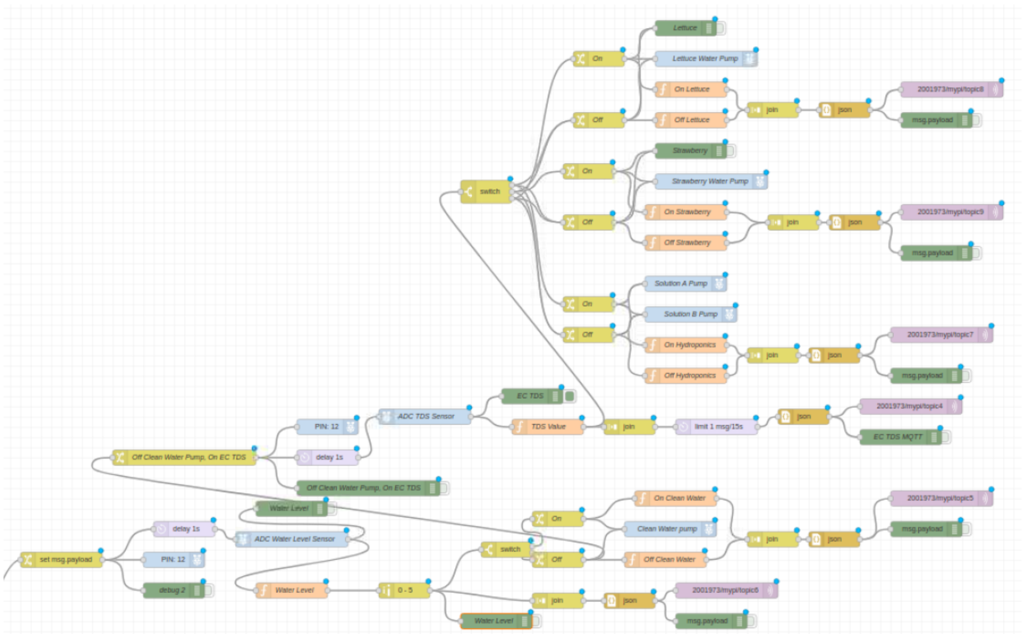


Figure 5.3.1.8 Programming flow for Water Level sensor, EC TDS sensor, and Water Pumps

Since both the water level sensor and the EC TDS sensor rely on releasing or detecting electrons to obtain readings from the water solution, users need to set up two rpi-gpio out nodes to control both sensors.

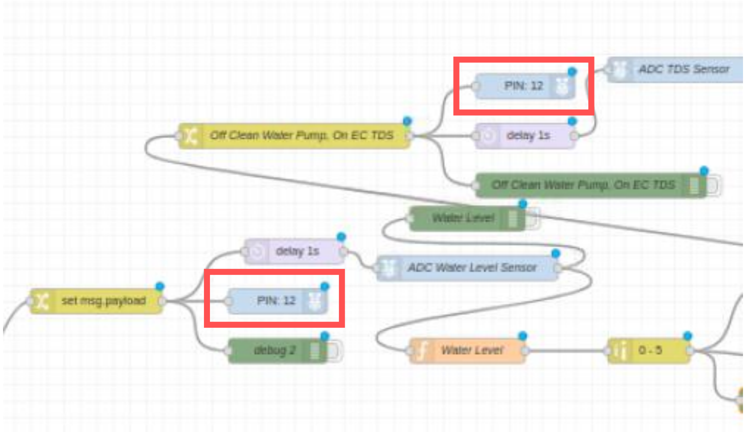


Figure 5.3.1.9 GPIO pin nodes for both water level and EC TDS sensors

Moving on to the next part, to standardize the readings from the water level sensor, users need to add a range node. The water level sensor provides readings in the range of 0 to 600, but by using the range node, these readings can be mapped to a more manageable range, such as 0 to 5. Additionally, the range node allows for rounding the values to the nearest integer, simplifying the data and making it easier to interpret.

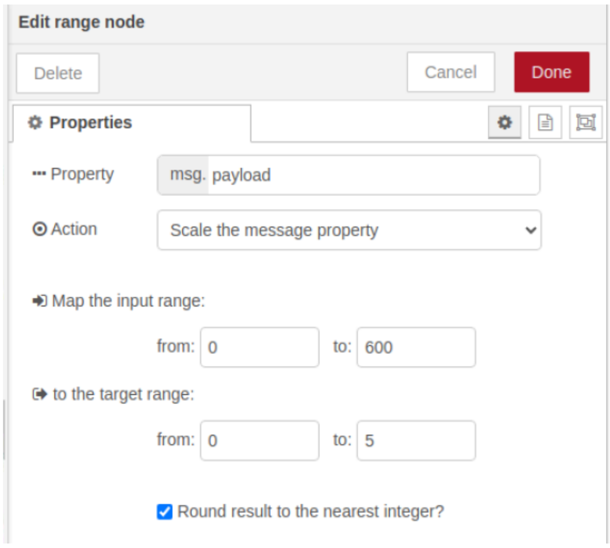


Figure 5.3.1.10 Configuration on range node

To collect data from analogue sensors such as the water level sensor, EC TDS sensor, and two current sensors via the MCP3008 ADC converter, users must program the nodes using four different A/D converter nodes. Each A/D converter node will correspond to a different sensor, and each node must be configured with the correct input pin based on the sequence in which the sensors are connected. For example, if the water level sensor is connected to the first input pin, the input pin on the corresponding A/D converter node should be configured as 'A0'.

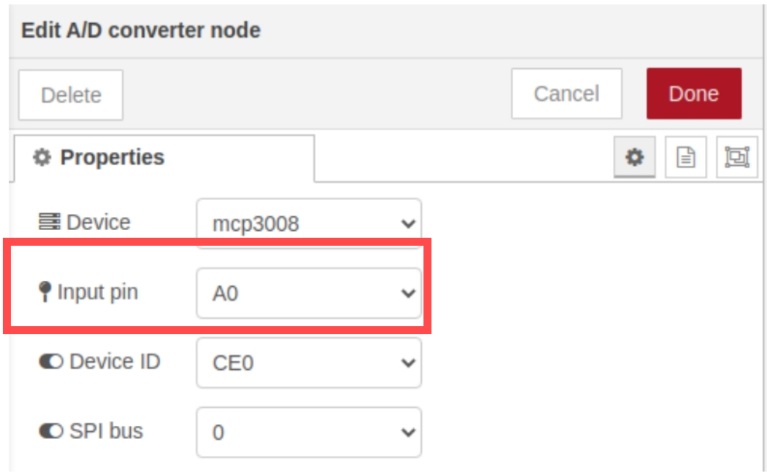


Figure 5.3.1.11 Configuration of the A/D converter node

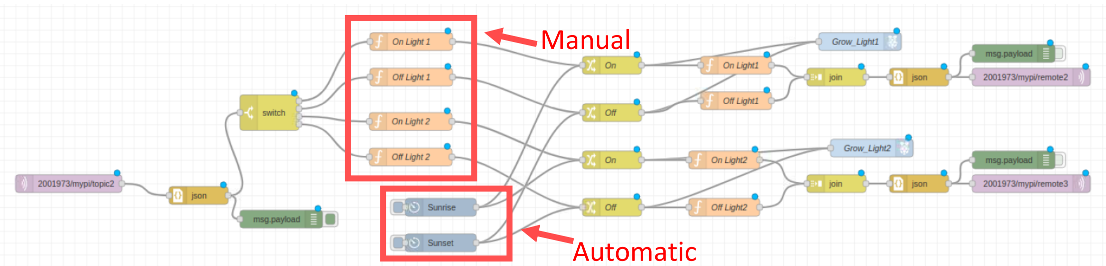


Figure 5.3.1.12 Programming flow for the grow lights

Furthermore, there are two methods for controlling the grow lights: manual and automatic. For manual control, users need to configure function nodes for each grow light. For automatic control, users must configure the cronplus node, which schedules the on and off times for the grow lights based on the solar events of the location's latitude and longitude. For example, if the framework is located in Malaysia, users can configure it with Kuala Lumpur's coordinates to align with the standard sunrise and sunset times.

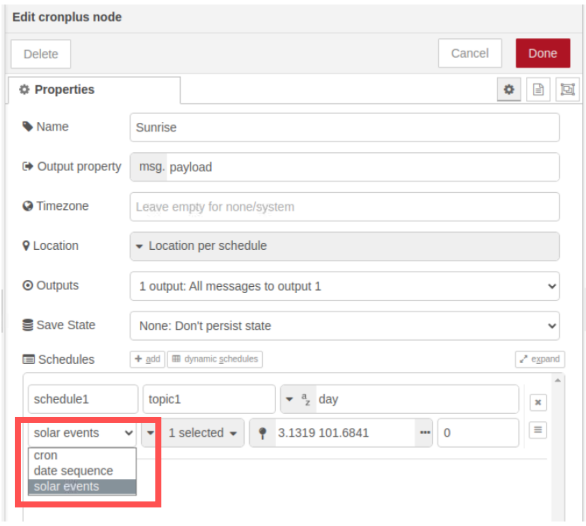


Figure 5.3.1.13 Configuration on the cronplus node (modes)

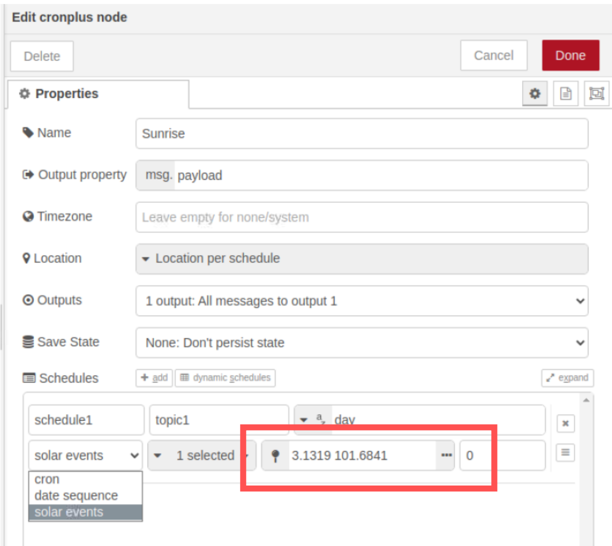


Figure 5.3.1.14 Configuration on the cronplus node (coordinate)

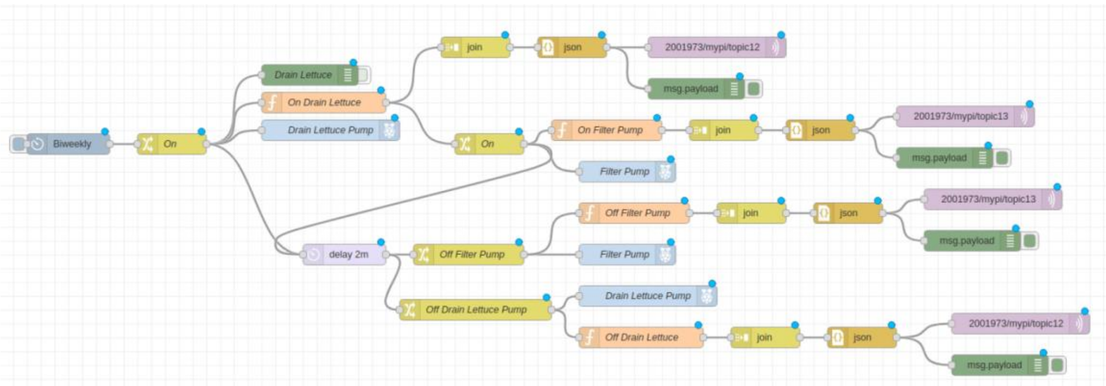


Figure 5.3.1.15 Programming flow for the water filtration system

5.3.2 Setting and Configuring on Laptop

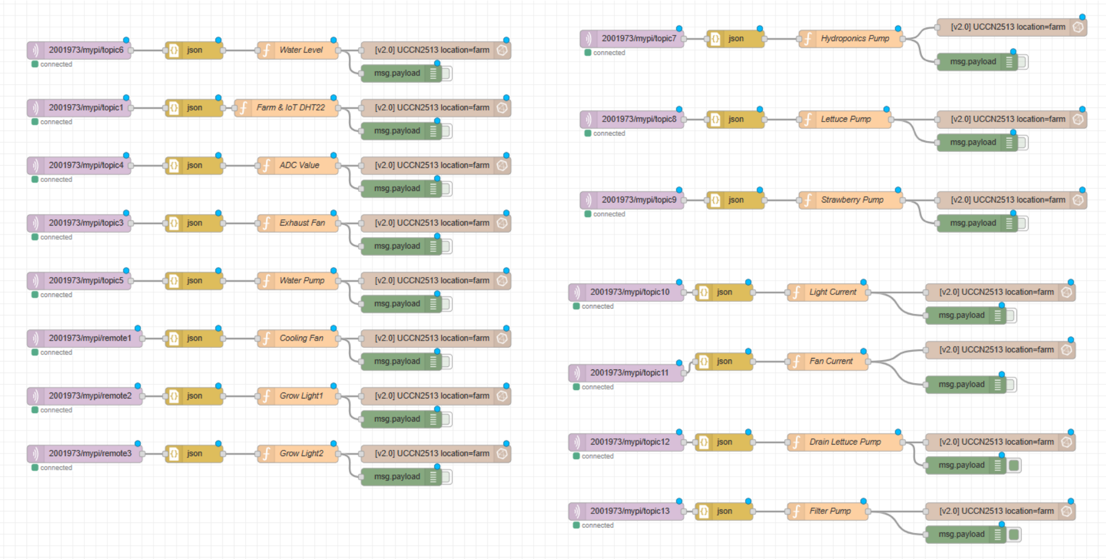


Figure 5.3.2.1 Programming flow for the data receiving from Raspberry Pi

As a monitoring device, users must program the flow in Node-RED to receive and transmit data to the Raspberry Pi. First, the laptop will be configured as the receiver to collect data from the Raspberry Pi, with all received data stored in InfluxDB. Each MQTT in node must be programmed with its respective topic, matching the one configured on the Raspberry Pi, to ensure the correct data is received. Additionally, the action property in the JSON node should be set to "Always convert to JavaScript Object."

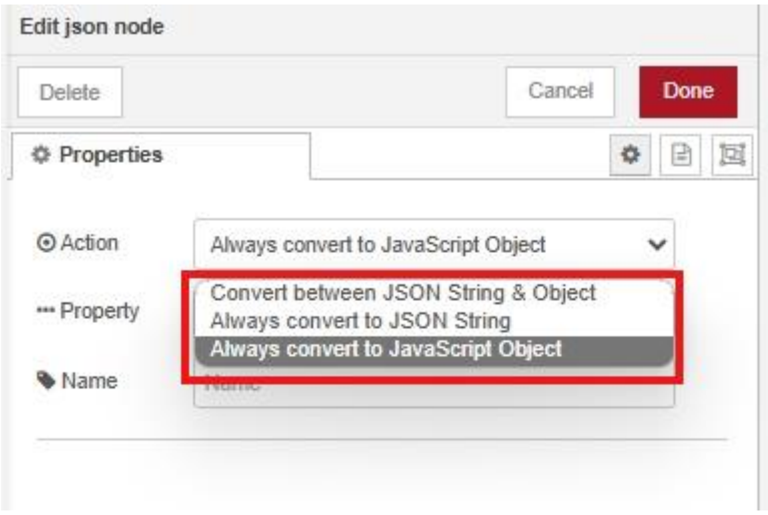


Figure 5.3.2.2 Configuration on json node

Next, function nodes need to be configured similarly to those on the Raspberry Pi to process the specific data collected. For example, if topic1 is assigned to the DHT22 sensor, the function node will process the temperature and humidity data from both the IoT devices and the farm.

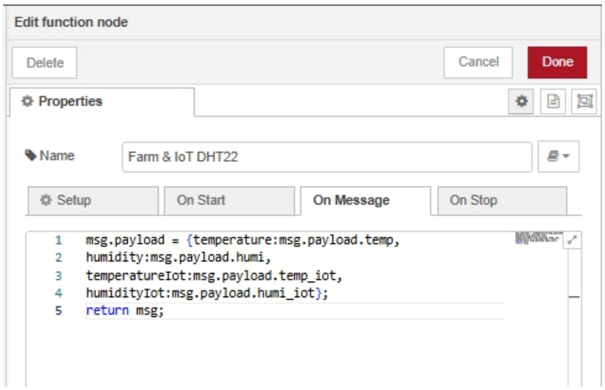


Figure 5.3.2.3 Configuration on the function node (DHT22)

Moving on to the next part, to send the sensor and actuator data into InfluxDB, users must configure each influxdb in node with a specific server token. This token, which is unique to each user's InfluxDB instance, identifies the server from which the data will be retrieved.

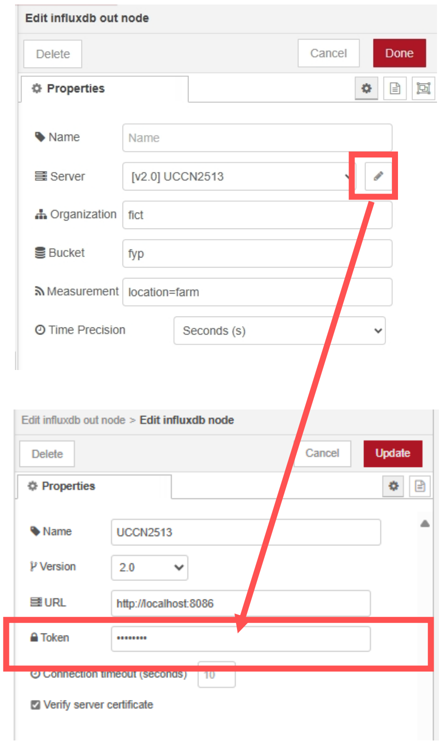


Figure 5.3.2.4 Configuration on influxdb node

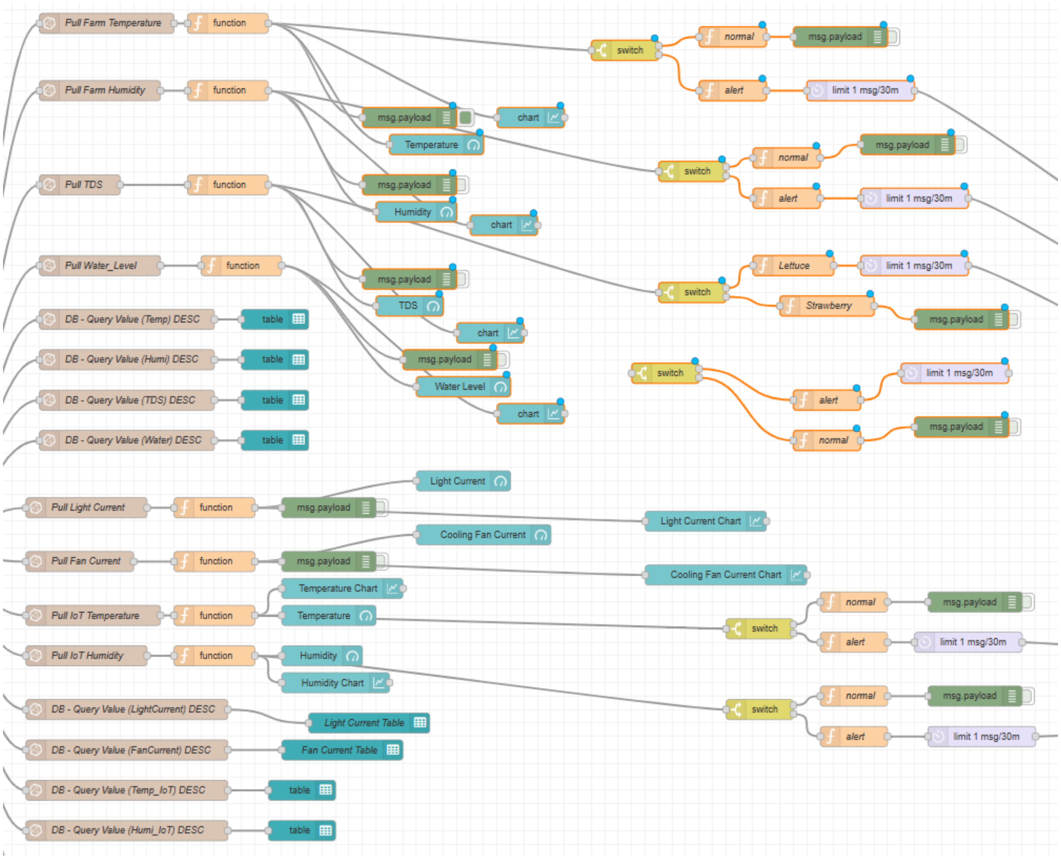


Figure 5.3.2.5 Programming flow for retrieving data for visualization

Moving on to the data visualization part of the Node-RED dashboard, each data point can be displayed in various formats such as gauges, charts, and tables. Users need to configure each influxdb in node to correspond with the respective influxdb out node configured earlier. Additionally, the function nodes connected to the influxdb in node must be configured with the commands shown in the figure below to ensure that the data retrieved matches the respective payloads. For example, the figure below shows a query configured to retrieve data from the "fyp" bucket, with r._measurement set to "location=farm" and r._field set to "temperature."

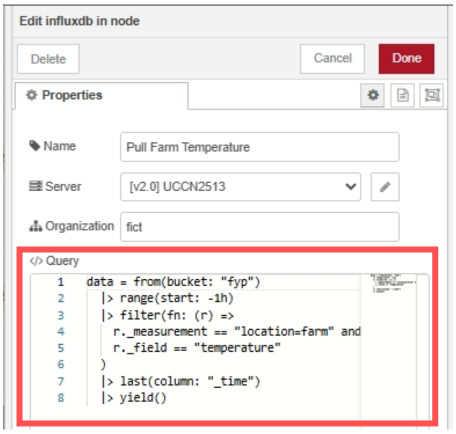


Figure 5.3.2.6 Configuration on influxdb in node (farm temperature)

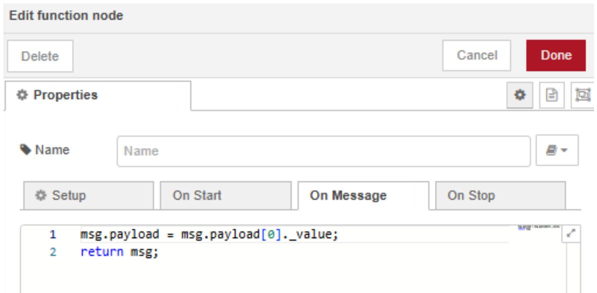


Figure 5.3.2.7 Configuration for all function nodes for influxdb in node

For data visualization, users can customize the dashboard according to their preferences. For example, the temperature data is configured to be displayed in a gauge format. Users can set different colors for different temperature ranges. The figure below illustrates that when the temperature is below 30°C, the gauge will be green; between 30-50°C, it will turn yellow; and above 50°C, it will display red.

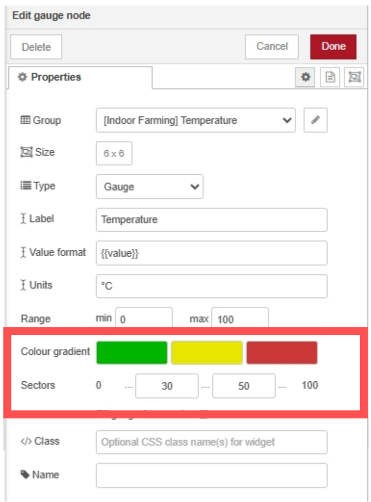


Figure 5.3.2.8 Configuration for gauge node (temperature)

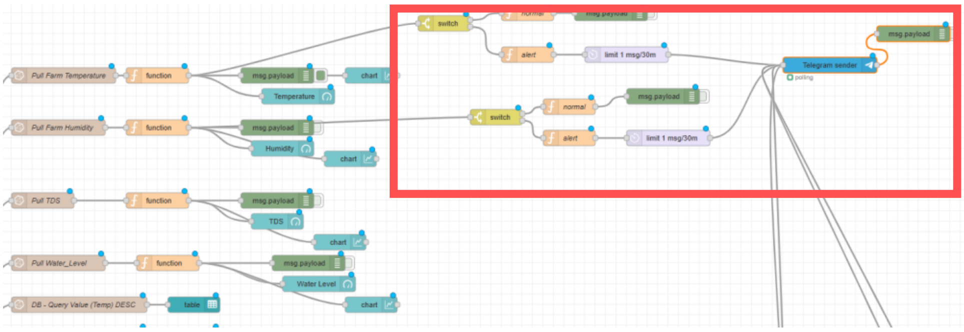


Figure 5.3.2.9 Programming flow for the message alerting function

Moreover, additional nodes connected to the flow that retrieves data from InfluxDB are used to detect abnormal data and send alert messages to users via Telegram Chatbot. Users need to input their correct chatId to ensure that alerts are sent to the appropriate chat. For example, in the figure below, if the temperature at the farm exceeds 35°C, a message will be sent through Telegram to notify users of the situation.

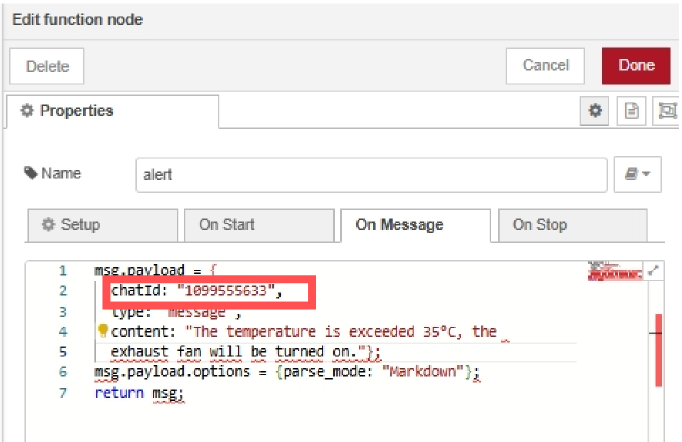


Figure 5.3.2.10 Configuration of the function node (farm temperature alerts)

Next, for the Telegram sender node configuration, users need to set it up with their respective chatbot. They must enter the chatbot's name and its unique token. It is important to ensure that these details are correct; otherwise, the message will not be received by the chatbot.

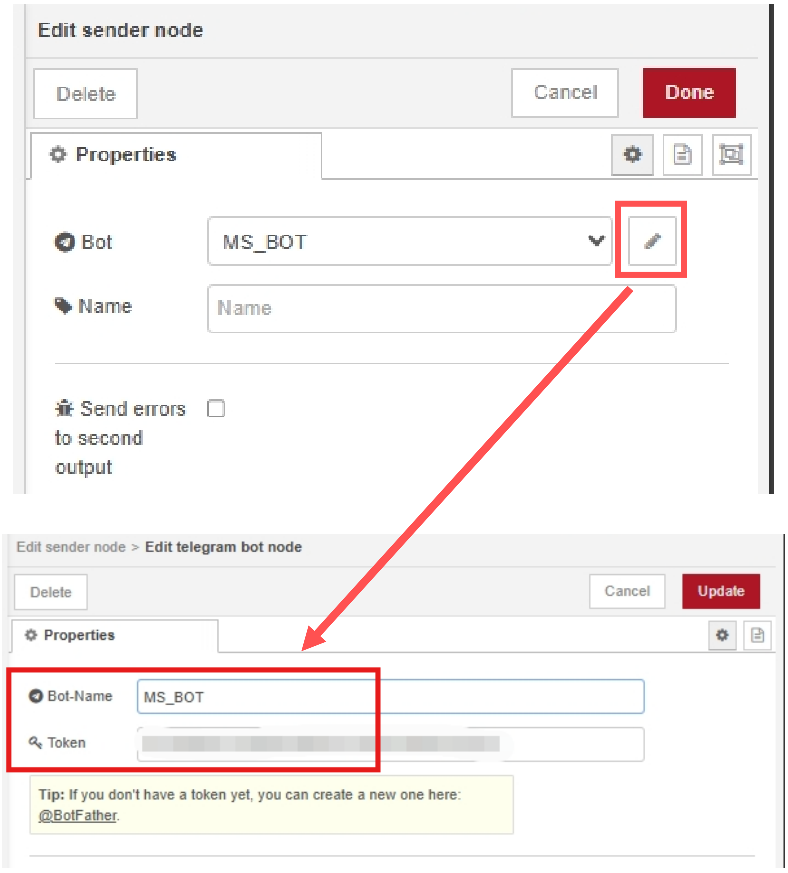


Figure 5.3.2.11 Configuration for Telegram Bot

Furthermore, to display the status of the actuators on the dashboard, users need to configure LEDs for the water pumps, fans, and grow lights.

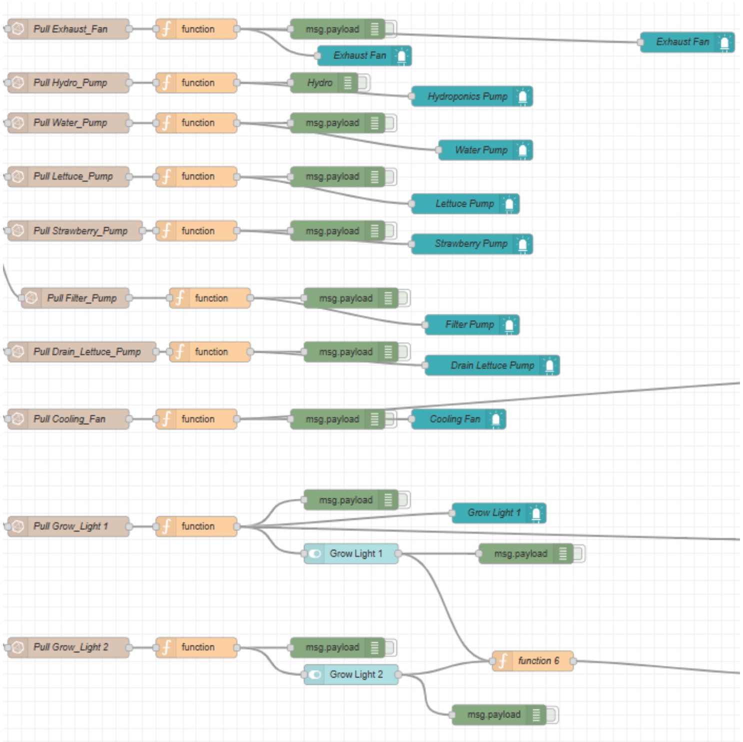


Figure 5.3.2.12 Programming flow for actuators visualization

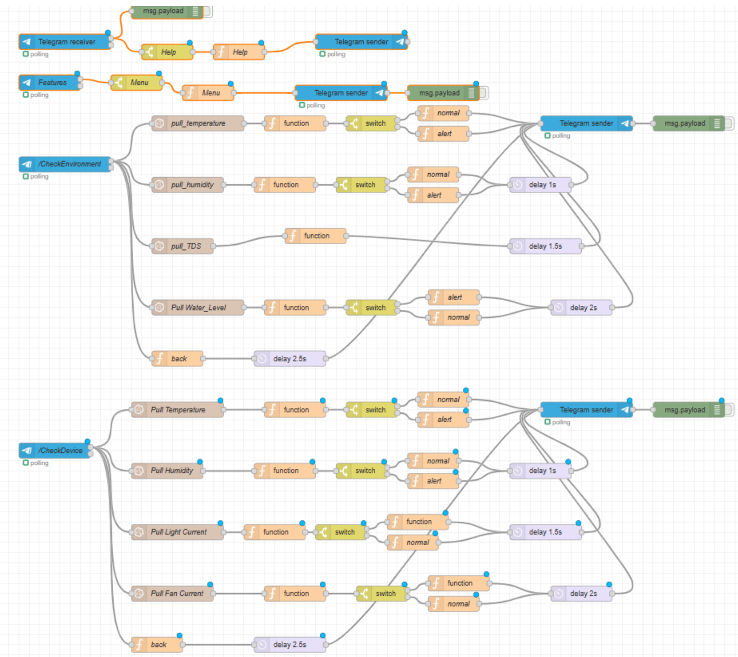


Figure 5.3.2.13 Programming flow for Telegram Chatbot Control

Moving on to the Telegram Chatbot, users need to configure the function nodes with the same chatId used for message alerts to display guidelines and menu options on the chatbot. This ensures that the chatbot provides the appropriate responses to user interactions. Additionally, users must configure the command node with the respective command. For example, as shown in the figure below, the /CheckEnvironment command will prompt the chatbot to reply with the temperature, humidity, TDS level, and water level conditions.

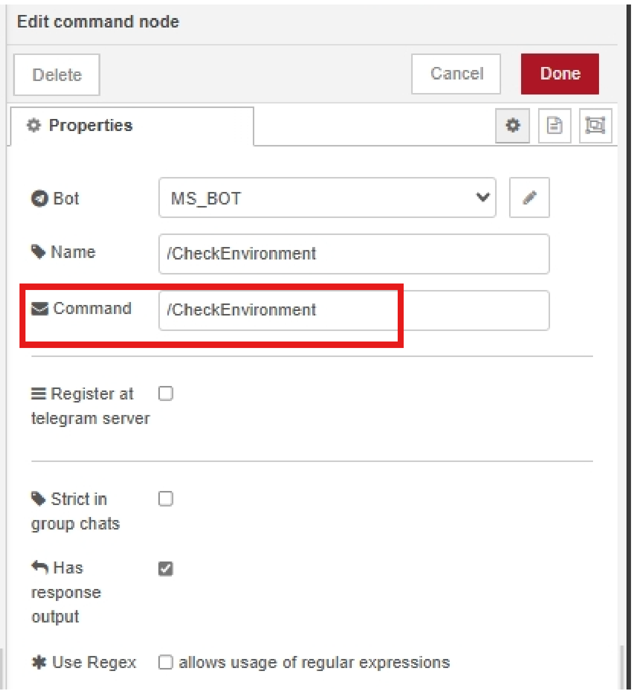


Figure 5.3.2.14 Configuration for telegram command node

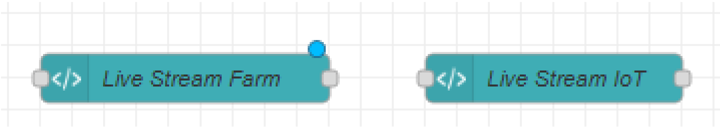


Figure 5.3.2.15 Programming flow for farm and IoT live streaming

Lastly, for live streaming display, users need to configure the template nodes with the respective live streaming channels, as shown below.

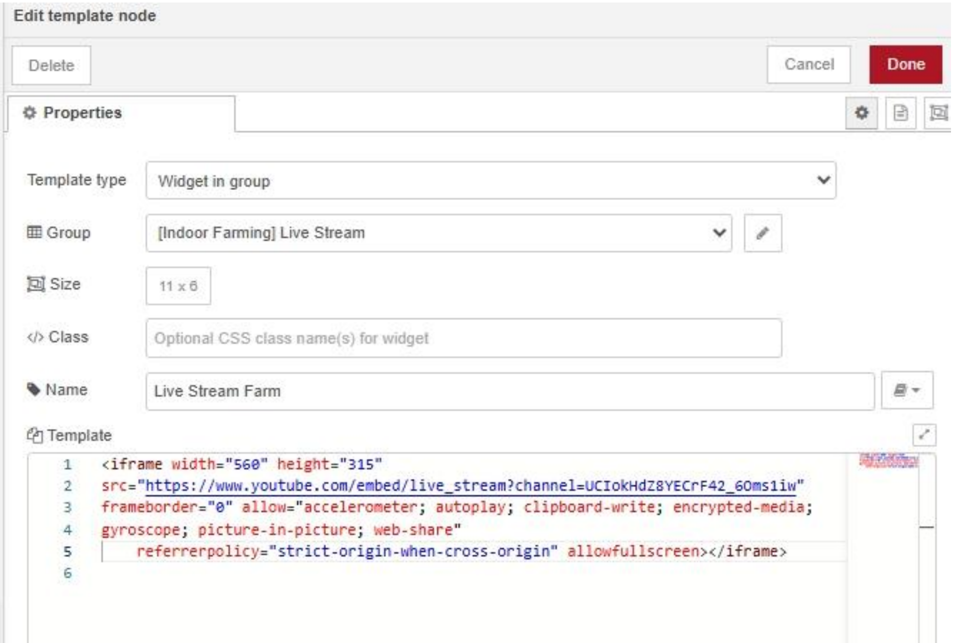


Figure 5.3.2.16 Configuration for template node (farm)

5.4 System Operation

In this system, there are three main element that helps the whole system to run smoothly, which are the operations on Raspberry Pi, the monitoring device and the Telegram Chatbot.

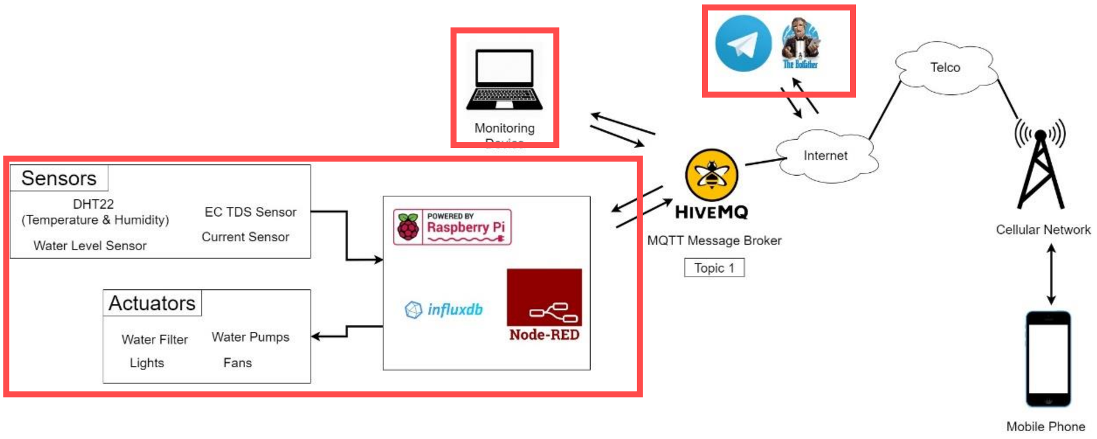


Figure 5.4.1 Architecture diagram of the project

5.4.1 Operation on Raspberry Pi

Firstly, the users must open the Terminal and configure the following command to ensure that the Node-RED will run automatically when the users bootup the Raspberry Pi. The Node-RED will automatically run all the flow which has been deployed by the users when the Raspberry Pi bootup.

Command: **sudo systemctl enable nodered.service**

Next, for the live streaming setup, the PiCam is configured to stream the farm environment to YouTube using raspivid and ffmpeg. By using the command below, users can configure the parameters and orientation of the live feed and start streaming via RTMP and ffmpeg. The stream key, which is unique, can be obtained from YouTube Studio.

Command: **raspivid -o - -t 0 -hf -fps 30 -b 6000000 | ffmpeg -re -ar 44100 -ac 2 -acodec pcm_s16le -f s16le -ac 2 -i /dev/zero -f h264 -i - -vcodec copy -acodec aac -ab 128k -g 50 -strict experimental -f flv rtmp://a.rtmp.youtube.com/live2/enter-stream-key**

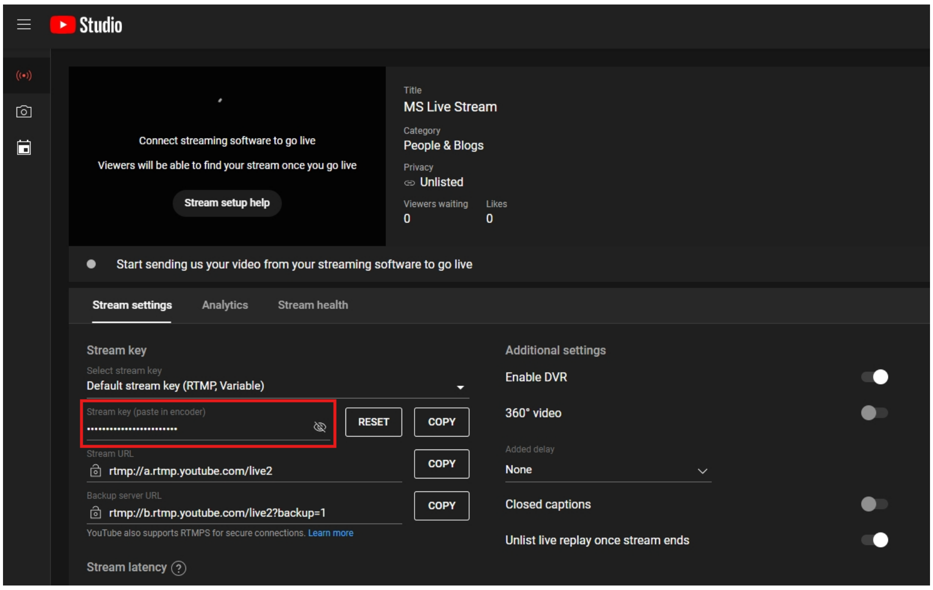


Figure 5.4.1.1 YouTube Studio Panel

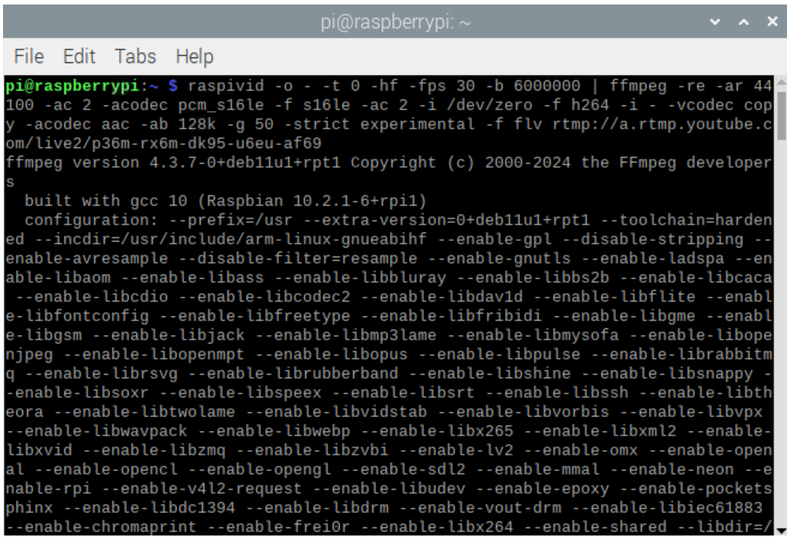


Figure 5.4.1.2 YouTube Live Streaming Process

5.4.2 Operation on Laptop

Before starting the Node-RED, the users are required to run the InfluxDB by entering the command, "**influxd**" in Command Prompt under the InfluxDB directory**.**

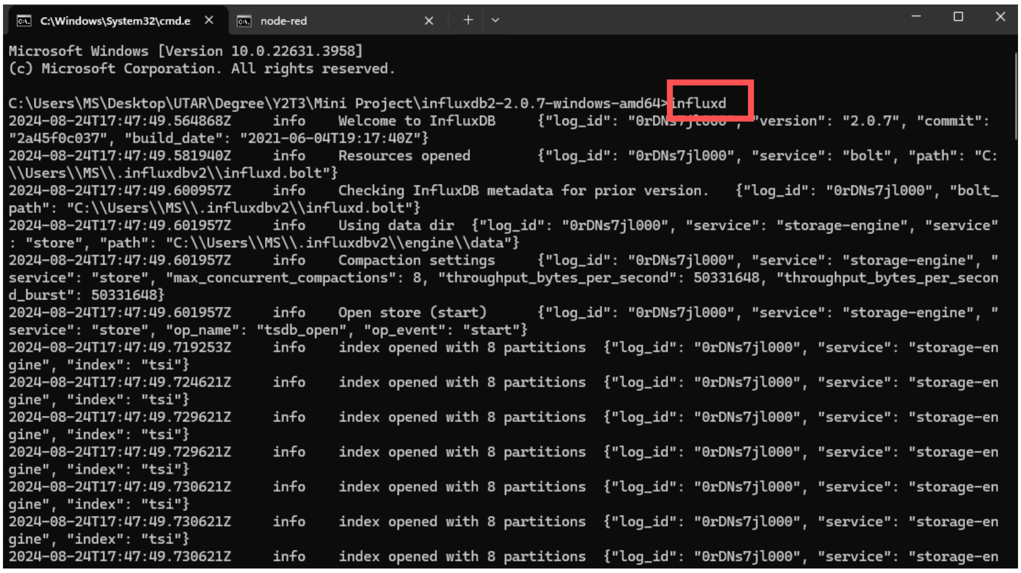


Figure 5.4.2.1 Command Prompt running InfluxDB

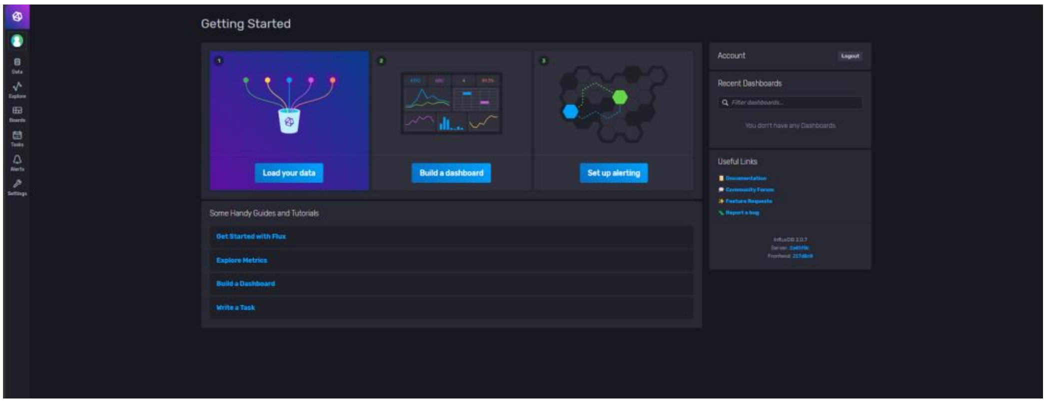


Figure 5.4.2.2 InfluxDB Panel

Once the database is up and ready, the users can start Node-RED by entering the command, "**node-red**" into Command Prompt.

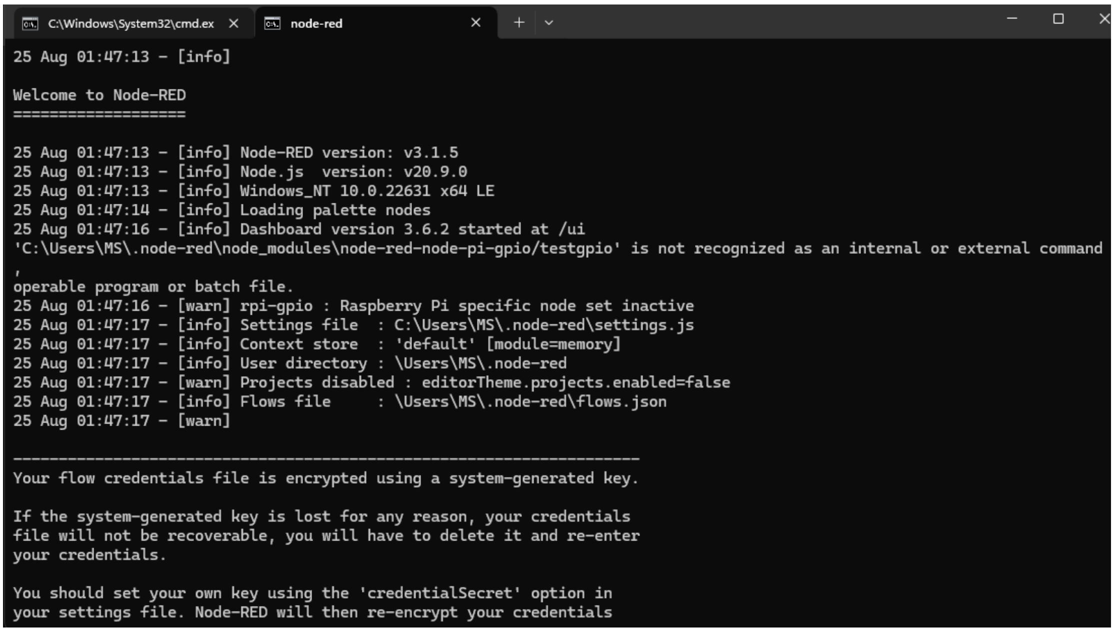


Figure 5.4.2.3 Command Prompt running Node-RED

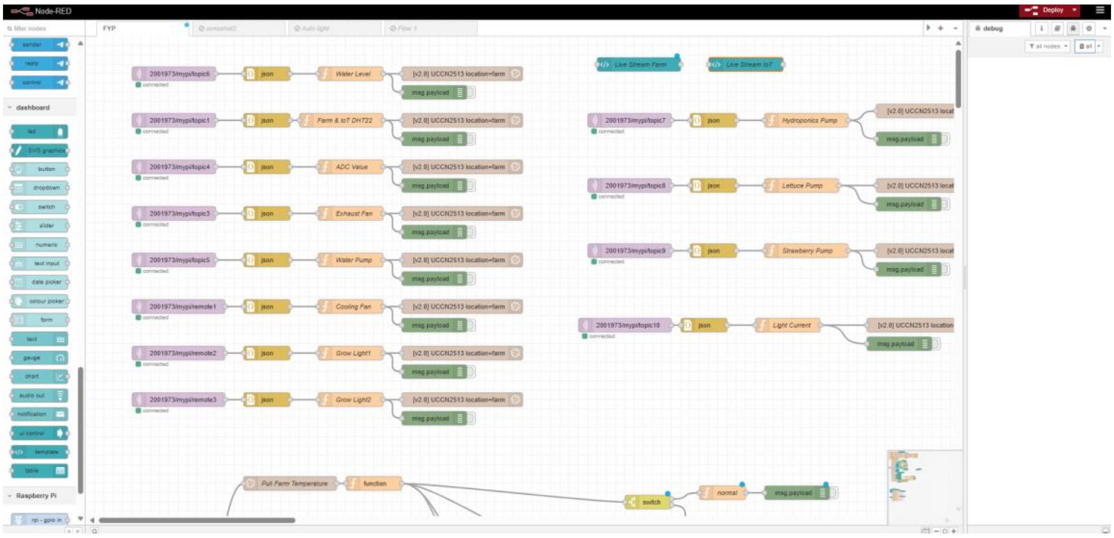


Figure 5.4.2.4 Node-RED panel on monitoring device

By entering the following web address into the any browser, the users can get to access to the Node-RED dashboard which displaying both agriculture monitoring system UI and the IoT device monitoring UI.

Figure 5.4.2.5 Agriculture Monitoring Dashboard

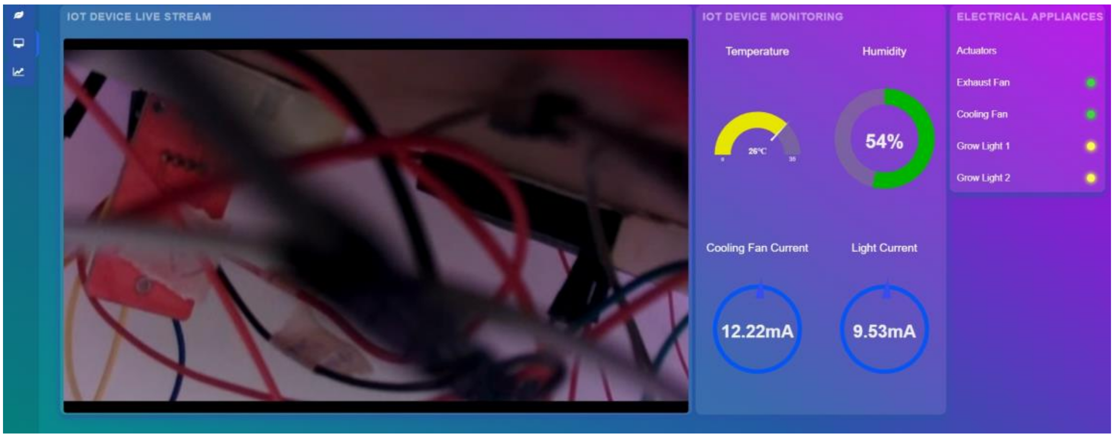


Figure 5.4.2.6 IoT Device Monitoring Dashboard

Figure 5.4.2.7 Farm & IoT Line Charts & Tables

5.4.3 Operation on Telegram

Firstly, the users need to open their Telegram application via laptop or smartphone. They can start the conversation with the chatbot with the command **/Menu**. After the chatbot respond to the users, they need to enter the respective command that they wish to retrieve from the system.

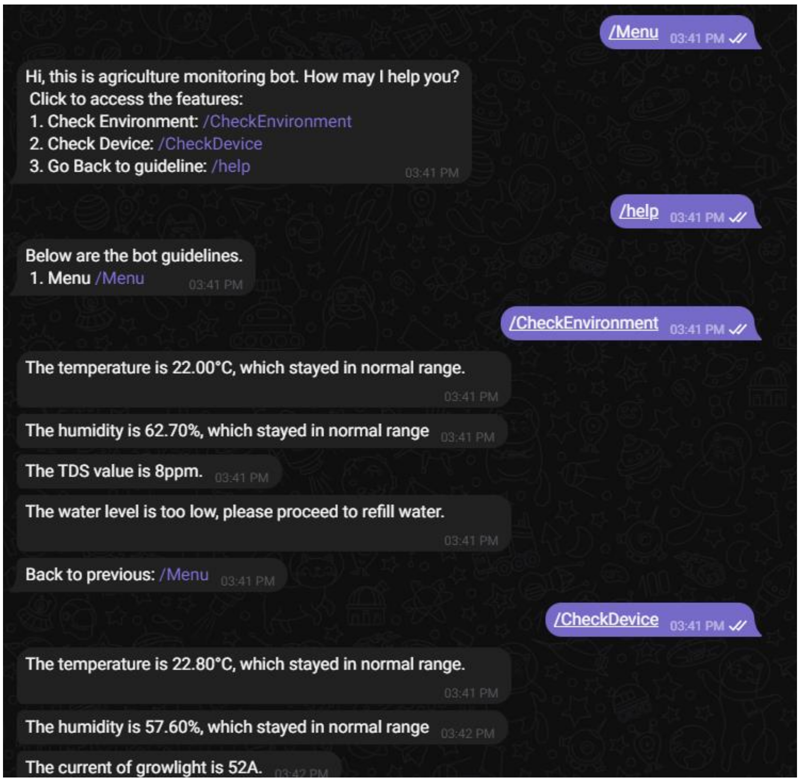


Figure 5.4.3.1 Operation on Telegram ChatBot

Also, the chatbot will be sending alert messages when there is any abnormal activity happened to the farm's condition.

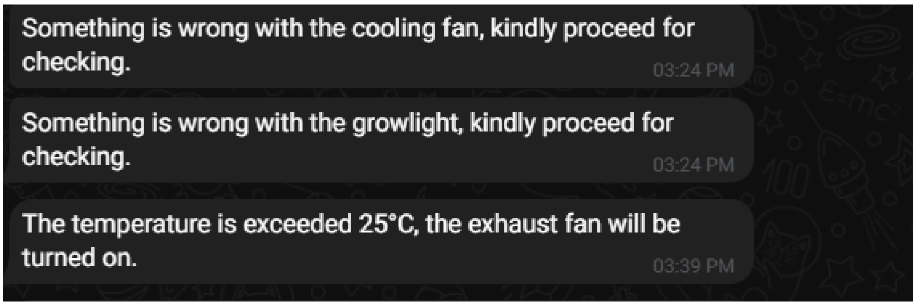


Figure 5.4.3.2 Receiving alert messages when abnormal incidents

5.5 Concluding Remark

This chapter outlines the steps involved in setting up the hardware and software components of the agricultural framework. The hardware setup involves connecting a Raspberry Pi to various sensors and actuators, such as DHT22 sensors, a PiCam module, water level sensors, EC sensors, and water pumps, all of which are managed by an 8-channel relay module. The Raspberry Pi acts as the central control unit, processing data from the sensors and controlling the actuators accordingly. A second Raspberry Pi, equipped with a Camera Module, is used for live-streaming the crop environment via YouTube. The software setup includes installing Raspberry Pi OS and Node-RED, followed by configuring the necessary nodes and flows to manage the data and control the hardware. InfluxDB is used for data logging, while a Telegram Chatbot, configured through BotFather, is employed for sending alerts and receiving user commands. The chapter concludes with instructions on how to configure the system to run automatically on startup and how to set up the live-streaming and monitoring components to ensure smooth operation.

CHAPTER 6

System Evaluation and Discussion

6.1 System Testing and Performance Metrics

To ensure system stability, system testing will be conducted to analyze the operation and performance of the system. In this project, four metrics have been selected for system evaluation: the accuracy of the system, the continuity of live feed streaming, and the growing condition of both lettuce and strawberries.

Firstly, the accuracy of the sensors is crucial in every agricultural monitoring system, as it directly reflects the farm's conditions to the users. In this project, the farm’s conditions—such as temperature, humidity, EC TDS, water level, and current value—will be collected through respective sensors. All the collected data will directly affect system performance as these data points will be used to determine system operation.

Secondly, the continuity of the live feed streaming is another important element, as users receive real-time information about the farm through the streaming. If anything, abnormal occurs on the farm, users can take immediate action to handle the incident.

Thirdly, component removal testing is a critical aspect of the system, as malfunctioning devices can impact overall system functionality. This testing ensures that the system can continue operating smoothly even if certain components fail, minimizing disruptions and maintaining performance under various conditions. Testing this aspect is essential to ensure system resilience and reliability.

Lastly, the growing conditions of both lettuce and strawberries are the main elements evaluated in the project, as these will prove the framework's usability and effectiveness. The conditions that will be evaluated include the number of leaves on the lettuce and the fruit on the strawberries.

6.2 Testing Setup and Evaluation

The metrics that have been selected to evaluate the system are the accuracy of the system, the continuity of live feed streaming, and the growing condition of both lettuce and strawberry.

6.2.1 The accuracy of the sensors

Comparing both temperature and humidity data collected by DHT22 & Thermometer.

DHT22		**Thermometer**	
Temperature (°C)	**Humidity (%)**	**Temperature (°C)**	**Humidity (%)**
26.30	54.50	26.00	55.00
25.40	55.00	25.00	56.00
24.50	56.10	24.00	55.00
24.70	57.2	24.00	54.00

Table 6.2.1 Comparison between DHT22 & Thermometer

Comparing both the Total Dissolved Solids (TDS) and Electric Conductivity (EC) value that collected by the TDS sensor and TDS meter.

TDS sensor		**TDS meter**	
TDS (ppm)	**EC**	**TDS (ppm)**	**EC**
472	738	474	741
469	734	471	736
295	462	297	465
359	561	356	557

Table 6.2.2 Comparison between TDS sensor and TDS meter

Comparing both the current value that collected by the current sensor and multimeter.

Current sensor	Multimeter
Current (mA)	Current (mA)
0.12	0.13
0.11	0.10
0.8	0.9
0.10	0.10

Table 6.2.3 Comparison between current sensor and multimeter

6.2.2 The continuity of YouTube live streaming

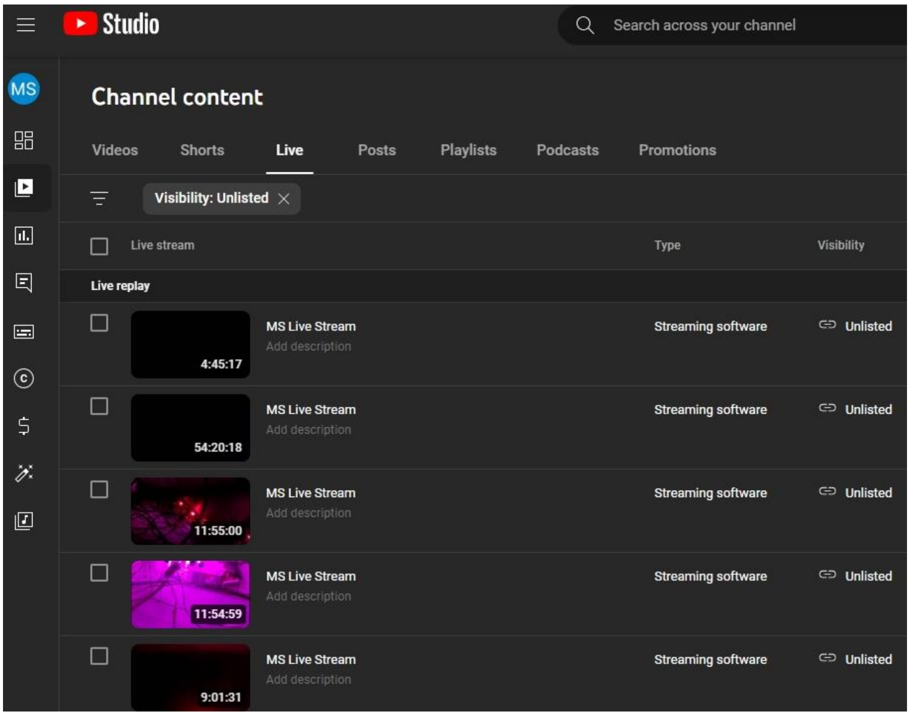


Figure 6.2.1 YouTube Live Stream Records

Figure 6.2.1 shows the previous YouTube live stream records, clearly indicating varying video durations. The framework is set up in the FYP lab, where multiple students share the Wi-Fi connection, so the quality and duration of the live streams depend largely on the stability of the lab's Wi-Fi. As shown in the figure, the longest live stream lasted for 54 hours, suggesting that streaming can continue for extended periods as long as the Wi-Fi bandwidth remains stable.

6.2.3 Components Removal Testing

Component	Impact on System Continuity
Power Supply	Complete system shutdown. No components will function without power.
Raspberry Pi Power Supply	Complete system shutdown. The entire system depends on its power supply for operation.
Internet Connectivity	The system continues locally, but remote monitoring and control through web interfaces will be unavailable.
MQTT	The system continues locally, but loses the ability to send and receive messages between devices and the server, disrupting data communication for remote control.
Telegram ChatBot	The system continues locally, but no alerts or updates will be sent via the chatbot, impacting remote notifications.
YouTube Live Stream	The system continues to function, but live streaming of the farm & IoT devices will be disabled, removing remote visual monitoring capability.
Node-RED Dashboard	The system continues to operate, but real-time sensor data visualization and control via the dashboard will be disabled.
Water Level Sensor	The system continues to function but loses the ability to monitor and control the water level. This could lead to inefficiency in water management.
EC TDS Sensor	The system remains operational, but no longer monitors or adjusts nutrient levels in the water. The precision of the hydroponic solution may degrade.
DHT22 Sensors (Farm)	The system continues, but environmental data like temperature and humidity in the farm area are no longer available, affecting automated climate control.
DHT22 Sensors (IoT)	The system continues, but loses its ability to monitor the IoT device's temperature and humidity, potentially affecting device performance monitoring.
Current Sensors (Exhaust Fan)	The system remains operational, but it cannot measure the exhaust fan's current draw, making energy monitoring for this

	component impossible.
Current Sensors (Grow Light)	The system continues, but current draw for the grow lights is unmonitored, which could affect energy usage reporting and fault detection.
Grow Lights	The system remains operational, but the lighting control for plant growth is disabled, impacting photosynthesis and plant health.
Clean Water Pump	The system continues, but cannot manage water circulation, risking stagnant water that can harm plant health.
Hydroponic Solution Pumps	The system continues, but the delivery of nutrient solution is halted, which may severely impact plant growth.
Lettuce Pump	The system remains operational, but without water circulation for the lettuce, nutrient distribution will be impaired.
Drain Lettuce Pump	The system continues, but draining water from the lettuce setup becomes impossible, potentially causing flooding or waterlogging.
Filtration Pump	The system continues, but loses filtration, resulting in unfiltered water being used in the system, which could lead to clogging or contamination.
Camera	The system remains operational, but live video feeds for monitoring crop conditions will be disabled.

Table 6.2.4 Impact of Removed Components on System Continuity

6.2.4 Growing condition of the lettuce

Weeks	Conditions	Number of leaves
1		2
2		2
3		3
4		3

5		**4**
6		**4**
7		**4**

Table 6.2.5 Lettuce's growing conditions

Based on the observations, the data presented above confirms that the framework is suitable for lettuce growth.

6.3 Project Challenges

In our life, not everything runs smoothly, and the same goes for this project development process. Throughout the development process, several challenges have been encountered, including plantation knowledge, environmental conditions, network connectivity, and hardware and software integration.

First, let's talk about plantation knowledge. Each user must be equipped with some basic knowledge to ensure that the plants grow healthily. Planting lettuces or strawberries is not as straightforward as it may seem; users need to consider several factors before starting, such as selecting good seeds, choosing the appropriate hydroponic solution, and determining the correct nutrient levels required by the plants.

Additionally, environmental conditions such as temperature and humidity are considered uncontrollable by humans, as each crop has its own preferable growing conditions. For example, the ideal temperature for growing lettuce is between 12-21°C, with 18°C being optimal. Therefore, the plantation framework must be designed to operate effectively under varying environmental conditions, which can impact the system's performance.

Thirdly, network connectivity also presents a challenge. As this project heavily relies on the network, data transmission between the Raspberry Pi and the monitoring device depends on a stable connection. If any network issues disrupt the connection, users may need to troubleshoot the hardware to restore system functionality.

Lastly, integrating various hardware and software components can be complex, especially for users with limited technical backgrounds. Compatibility issues between different hardware and software components can hinder overall system functionality. Ensuring seamless interaction between sensors, actuators, and the Raspberry Pi requires careful planning and testing.

Addressing these challenges requires careful planning and consideration to ensure the system operates effectively and reliably throughout its lifecycle.

6.4 SWOT

Strengths

Node-RED is a low-code platform that simplifies programming, making it accessible for users with minimal technical backgrounds. Its intuitive interface enables easy understanding and programming of the framework with minimal effort, which is particularly advantageous for non-IT students and users. Additionally, this project involves using cost-effective sensors and components, encouraging a wider range of applications among hobbyists and indoor farming enthusiasts. Furthermore, this system provides real-time monitoring and data-driven decision-making, which helps enhance crop management and productivity.

Weaknesses

The project's performance and output can be influenced by various factors, including environmental conditions and network stability. As this project heavily relies on a stable internet connection, all the data from the Raspberry Pi will be transferred via the MQTT protocol to the laptop, which can be a point of failure if network issues arise. Additionally, the integration of various hardware and software components may be challenging for non-IT students and may require significant troubleshooting.

Opportunities

This project utilizes affordable sensors and components, making it accessible to individuals and encouraging exploration in indoor farming, which promotes sustainable agriculture practices. Additionally, this system can be scaled to larger operations or customized for different types of crops, increasing its applicability and reach. Leveraging IoT devices and automation technologies can lead to further innovations and improvements in agricultural practices.

Threats

There are several threats that may impact this project, with network stability being a primary concern. Wireless network issues, including connectivity drops or interference, can disrupt system operations and data transmission, as the entire infrastructure relies on a stable network to function smoothly. Additionally, uncontrollable conditions such as temperature, humidity fluctuations, and other environmental factors can directly affect system performance and plant health. As previously mentioned, the system is connected to the internet for data transmission, which may expose it to security risks.

6.5 Objectives Evaluation

During Chapter 1, the project's objectives have been described as the guiding direction for the system to achieve its goals.

Objective 1: Ensure the accuracy of the EC TDS reading

This objective has been successfully achieved, as the EC TDS sensor used in the project delivers highly accurate readings. Despite being a standard IoT sensor, it demonstrates remarkable precision, with less than a one percent difference compared to industry-grade sensors. For instance, while the industry-grade sensor recorded a value of 741, the EC TDS sensor measured 738, showcasing its reliability and accuracy for practical use.

Objective 2: Implement automation features in the framework

This objective has been considered as successfully achieved. The framework can automatically operate once the power is turned on. When powered on, the water level sensor begins taking readings, followed by the EC TDS sensors. Other actuators, such as grow lights, exhaust fan, cooling fan, and water pumps, operate according to the data collected from the sensors. Also, there grow lights will be turning on according to the real time sunrise and sunset which has been configured to the Kuala Lumpur Longitude and Latitude.

Objective 3: Develop an automated water filter system

This objective has been successfully achieved. The mixed solution water used to provide nutrients for the crops is recycled through a water filter straw. Once filtered, it is pumped back into the clean water tank for further reuse.

6.6 Concluding Remark

This chapter lists several metrics used to measure the system's performance. Various tests have been conducted, and their results are clearly stated in the report. Additionally, the challenges faced throughout the development process are discussed. Lastly, a SWOT analysis of the project has been conducted, and the achievement of project objectives has been evaluated.

CHAPTER 7

Conclusion and Recommendation

7.1 Conclusion

In conclusion, the research presented in this project addresses the urgent need to modernize agricultural practices in response to escalating global challenges, particularly in the areas of food security and environmental sustainability. With the world's population steadily increasing, the pressure to meet the growing demand for food while minimizing the ecological impact of agricultural activities is intensifying. Conventional farming methods often fall short of these goals, contributing to problems such as soil degradation, water contamination, and excessive greenhouse gas emissions.

To tackle these challenges, this research project has focused on developing an innovative IoT-based Agriculture Monitoring Framework, integrated with automated fertigation and filtration capabilities. By utilizing advanced technologies such as Raspberry Pi, Node-RED, sensors, actuators, and InfluxDB, the framework facilitates real-time, data-driven decision-making alongside automated actions. These components enable farmers to optimize resource utilization, enhance crop yields, and reduce environmental impacts.

The project objectives, including ensuring precise EC TDS readings and creating an automated water filtration system, have been meticulously crafted to address critical aspects of modern agricultural practices. By leveraging IoT technology and automation, the framework enables precise nutrient management tailored to the specific requirements of plants, particularly in hydroponic environments. This approach not only improves productivity and efficiency but also promotes environmental sustainability by minimizing waste and conserving resources.

Moreover, the literature review in this research highlights the importance of sensor-based technology and real-time data processing in optimizing hydroponic farming. Through a critical analysis of existing research, gaps and opportunities for improvement were identified, providing a foundation for the proposed solutions to enhance the efficiency, reliability, and user-friendliness of automated agricultural systems.

Additionally, four development models were examined to ensure the project could be completed on time: the waterfall model, iterative model, spiral model, and Kanban model. Among these, the Kanban model was selected for its flexibility, allowing new requirements to be added during any development phase. The hardware and software requirements for the project were clearly specified, and the functional requirements were outlined to ensure the project's development followed the correct direction. The project milestones for FYP1 and FYP2 were also stated clearly to ensure progress could be tracked and achieved. Furthermore, the estimated costs for the hardware components used in the project were provided in detail.

The architecture diagram of the system was explained thoroughly to ensure that readers could understand how the system functions. The connections between the Raspberry Pi and other hardware components, such as the water level sensor, EC TDS sensor, current sensors, and relays, were clearly described. Additionally, the functional modules of the system—data collection, live streaming, automation control, data visualization, and Telegram ChatBot—were explained in detail. A system flowchart was also included to provide a clear overview of the processes involved in the system. Both the Node-RED dashboard GUI and the Telegram ChatBot interfaces were showcased in the report.

Regarding the setup procedure, all the steps for configuring the hardware and software for both the Raspberry Pi and the laptop were detailed. Screenshots of the system's operation were described in simple terms so that users could easily understand how it works.

Finally, several system and performance tests were conducted, evaluated using four metrics: sensor accuracy, live stream continuity, component removal testing, and the growing conditions of the lettuce. The results of the tests indicated that while the system is reliable and perfect for planting. Several challenges were encountered during the development process, including limited knowledge of plantation techniques, hardware-software integration, environmental conditions, and network connectivity. A SWOT analysis and objectives evaluation were conducted to ensure the system is fully operational.

In summary, this research project represents a significant step towards modernizing agricultural practices, aligning with global efforts to ensure food security and promote environmental sustainability. By leveraging IoT and automation, the developed framework offers a promising solution to the challenges facing modern agriculture, paving the way for a more efficient, environmentally conscious, and productive future.

7.2 Recommendation

Although the agriculture monitoring framework designed and built in this project has achieved most of its objectives, there are still some improvements that can be made to further extend its utilization for farmers.

- **Test the Suitability of Other Crop Species in the Framework:** The crops that have been planted in the framework include lettuce, which has been shown to grow healthily, almost similar to lettuce grown in outdoor soil. To further utilize the framework, users can consider placing other crop species in the framework and observing their growing conditions.

- **Enhance Data Analytics and Reporting:** Integrate advanced data analytics tools to provide deeper insights into crop health, growth patterns, and environmental conditions. Automated reporting can help farmers make informed decisions and improve crop management.

- **Integrate Machine Learning for Predictive Analysis:** Implement machine learning algorithms to predict potential issues such as pest infestations, nutrient deficiencies, or irrigation needs. This can help in proactive management and ensure optimal crop growth.

- **Expand IoT Integration:** Future versions of framework could integrate with additional IoT technologies to enhance the precision of environmental monitoring and optimize control over system components such as sensors and actuators. This would allow for a greater automation, further minimizing the need for manual intervention.

REFERENCES

[1] E. W. Stein, "The transformative environmental effects large-scale indoor farming may have on air, water, and Soil," Air, Soil and Water Research, vol. 14, p. 117862212199581, 2021. doi:10.1177/1178622121995819

[2] H. Morales, I. Perfecto, and B. Ferguson, "Traditional fertilization and its effect on corn insect populations in the Guatemalan Highlands," Agriculture, Ecosystems & Environment, vol. 84, no. 2, pp. 145–155, 2001. doi:10.1016/s0167-8809(00)00200-0

[3] J. M. Scriber, "Nitrogen nutrition of plants and insect invasion," Nitrogen in Crop Production, pp. 441–460, 2015. doi:10.2134/1990.nitrogenincropproduction.c29

[4] H. D. MCGUINNESS, The Importance of Plant Diversity and the Nutritional Content of the Diet on the Population Dynamics of Herbivorous Insects. 1987.

[5] Bernama, "Malaysia's population to grow 2.1% in 2023," MalaysiaNow, https://www.malaysianow.com/news/2023/07/31/malaysias-population-to-grow-2-1-in-2023 (accessed Aug. 30, 2023).

[6] Food and Agriculture Organization of the United States, "The State of Food Security and Nutrition in the World 2023," Food and Agriculture Organization of the United States, https://www.fao.org/documents/card/en/c/cc3017en (accessed Aug. 30, 2023).

[7] N. Patterson, "What is Environmental Sustainability? Goals with Examples ," Southern New Hampshire University, https://www.snhu.edu/about-us/newsroom/stem/what-is-environmental-sustainability (accessed Aug. 30, 2023).

[8] United States Environmental Protection Agency, "Sustainability and the Roe | US EPA - U.S. Environmental Protection Agency," United States Environmental Protection Agency, https://www.epa.gov/report-environment/sustainability-and-roe (accessed Aug. 28, 2023).

[9] IEA, "CO2 Emissions in 2022," IEA, https://www.iea.org/reports/co2-emissions-in-2022 (accessed Aug. 28, 2023).

[10] R. Tandil, J. Yapson, W. Atmadja, S. Liawatimena, and R. Susanto, "Hydroponic nutrient mixing system based on STM32," IOP Conference Series: Earth and

Environmental Science, vol. 195, p. 012052, 2018. doi:10.1088/1755-1315/195/1/012052

[11] A. V. Hartanto, F. S. Kristiady, W. Atmadja, and J. Linggarjati, "Automatic hydroponic nutrient mixing for hydroponic NFT and fertigation," IOP Conference Series: Earth and Environmental Science, vol. 794, no. 1, p. 012128, 2021. doi:10.1088/1755-1315/794/1/012128

[12] D. K. Aji, U. Nurhasan, R. Arianto, and O. D. Triswidrananta, "Smart ecosystem for hydroponic land in the hydroponic farmers group Guided by CSR Pt. Otsuka Indonesia as an improved quality and quantity of harvest results," IOP Conference Series: Materials Science and Engineering, vol. 1073, no. 1, p. 012030, 2021. doi:10.1088/1757-899x/1073/1/012030

[13] R. F. Rahmat, S. Adnan, R. Anugrahwaty, E. P. Alami, and B. Siregar, "Red Onion Growth Monitoring System in hydroponics environment," Journal of Physics: Conference Series, vol. 1235, no. 1, p. 012117, 2019. doi:10.1088/1742-6596/1235/1/012117

[14] World Bank Group, "Food security: Rising food insecurity in 2023," World Bank, https://www.worldbank.org/en/topic/agriculture/brief/food-security-update (accessed Sep. 3, 2023).

[15] C. Jayathilaka, "Waterfall vs. Agile Methodology," Medium, https://medium.com/@chathmini96/waterfall-vs-agile-methodology-28001a9ca487 (accessed Mar. 7, 2024).

[16] G. Learner, "Iterative model: Software engineering," Medium, https://genius-learner.medium.com/iterative-model-software-engineering-b11a76da7895 (accessed Mar. 8, 2024).

[17] ProfessionalQA, "What is iterative model?," Iterative Model: Advantages and Disadvantages, https://www.professionalqa.com/iterative-model (accessed Mar. 8, 2024).

[18] GeeksforGeeks, "Spiral Model - Software Engineering," GeeksforGeeks, https://www.geeksforgeeks.org/software-engineering-spiral-model/ (accessed Mar. 8, 2024).

[19] nTask, "A brief introduction to the Kanban methodology," Medium, https://medium.com/ntask/a-brief-introduction-to-the-kanban-methodology-eb455f5f7c94 (accessed Mar. 9, 2024).

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2024	**Study week no.: 2**
Student Name & ID: See Keng Lek (20ACB01973)	
Supervisor: Ts Dr. Goh Hock Guan	
Project Title: Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED	

1. WORK DONE

In evaluating my progress during the first two weeks of FYP2, I am satisfied with the groundwork laid for the project during FYP1. By prioritizing tasks, reviewing the previous weakness in FYP1, I have successfully identified the weakness in the previous phase of FYP.

During this period, I dedicated significant effort to collecting relevant information and identify the current project's weakness. This process is trying to ensure the system to work smoothly.

2. WORK TO BE DONE

In the following two weeks, I will be starting to look into the solution in resolving the weaknesses of that has been identified. Once the solutions has been found, I will start to improve the current system.

3. PROBLEMS ENCOUNTERED

Fortunately, I did not encounter any significant problems during this period, allowing for smooth progress and momentum. However, I remain vigilant and prepared to address any challenges that may arise in the future.

4. SELF EVALUATION OF THE PROGRESS

In terms of self-evaluation, I am satisfied with the progress made during these initial weeks of FYP2. Moving forward, I remain committed to executing the project plan diligently and addressing any challenges that may arise along the way.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2024	Study week no.: 4
Student Name & ID: See Keng Lek (20ACB01973)	
Supervisor: Ts Dr. Goh Hock Guan	
Project Title: Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED	

1. WORK DONE

During week 3 and 4, I began working on the water filtration system. I have started on researching on the water filter for the system. Meanwhile, I have also done with the improvement on the identified weakness.

2. WORK TO BE DONE

Over the next two weeks, I will be starting to order the water filter for the system and connecting Raspberry Pi with other water filtration equipment which ordered from these two weeks and create flow of the system.

3. PROBLEMS ENCOUNTERED

The problem that I have encountered was the various type of water filters and their specification. As there are many kinds of similar water filters in the market, so I must find the suitable one for my project.

4. SELF EVALUATION OF THE PROGRESS

In evaluating my progress, I am satisfied with the accomplishments achieved during weeks 3 and 4.

Goh

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2024	**Study week no.: 6**
Student Name & ID: See Keng Lek (20ACB01973)	
Supervisor: Ts Dr. Goh Hock Guan	
Project Title: Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED	

1. WORK DONE During week 5 and 6, I have ordered the water filter for the project and meanwhile I have also started developing the programming flow for the filtration system.
2. WORK TO BE DONE Over the next two weeks, I will be starting to connect Raspberry Pi with the water filtration system and continue with developing the programming flow.
3. PROBLEMS ENCOUNTERED Fortunately, I did not encounter any significant problems during this period, allowing me to maintain focus and progress smoothly.
4. SELF EVALUATION OF THE PROGRESS In evaluating my progress, I am satisfied with the continued momentum and dedication to securing essential hardware components for the project. By diligently pursuing the hardware procurement process and maintaining focus on upcoming tasks, I am confident in the project's trajectory. Moving forward, I remain committed to executing the project plan diligently and addressing any challenges that may arise.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2024	**Study week no.: 8**
Student Name & ID: See Keng Lek (20ACB01973)	
Supervisor: Ts Dr. Goh Hock Guan	
Project Title: Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED	

1. WORK DONE

During weeks 7 and 8, I initiated the process of connecting the Raspberry Pi with the water filter that was ordered during this period. Additionally, I began to publish the data to the monitoring device through MQTT.

2. WORK TO BE DONE

Over the next two weeks, my focus will remain on attaching the Raspberry Pi with the sensors, actuators and water filter on to the framework. I will start to conduct overnight tests in the lab to thoroughly evaluate the functionality of the system and ensure its reliability.

3. PROBLEMS ENCOUNTERED

Fortunately, I did not encounter any significant problems during this period, allowing me to maintain focus and progress smoothly.

4. SELF EVALUATION OF THE PROGRESS

In evaluating my progress, I am satisfied with the continued momentum and dedication to securing essential hardware components for the project.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2024	**Study week no.: 10**
Student Name & ID: See Keng Lek (20ACB01973)	
Supervisor: Ts Dr. Goh Hock Guan	
Project Title: Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED	

1. WORK DONE

During weeks 9 and 10, I continued the previous tasks of attaching the Raspberry Pi, sensors, actuators and water filter on to the framework. Additionally, I initiated the process of running the system overnight in the lab.

2. WORK TO BE DONE

Over the next two weeks, my focus will remain on conducting overnight tests in the lab to thoroughly evaluate the functionality of the system and ensure its reliability, as well as refining the system flow.

3. PROBLEMS ENCOUNTERED

I encountered a minor setback while attempting to continuously perform live streaming on YouTube, which this problem was caused by the YouTube embedded code from my senior.

4. SELF EVALUATION OF THE PROGRESS

Reflecting on my progress, I'm pleased with the progress made in integrating the hardware and testing the system. The issue with YouTube live streaming reminded me of the importance of thorough testing. Moving forward, I'll focus on refining the system and addressing any challenges to ensure the project's success.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

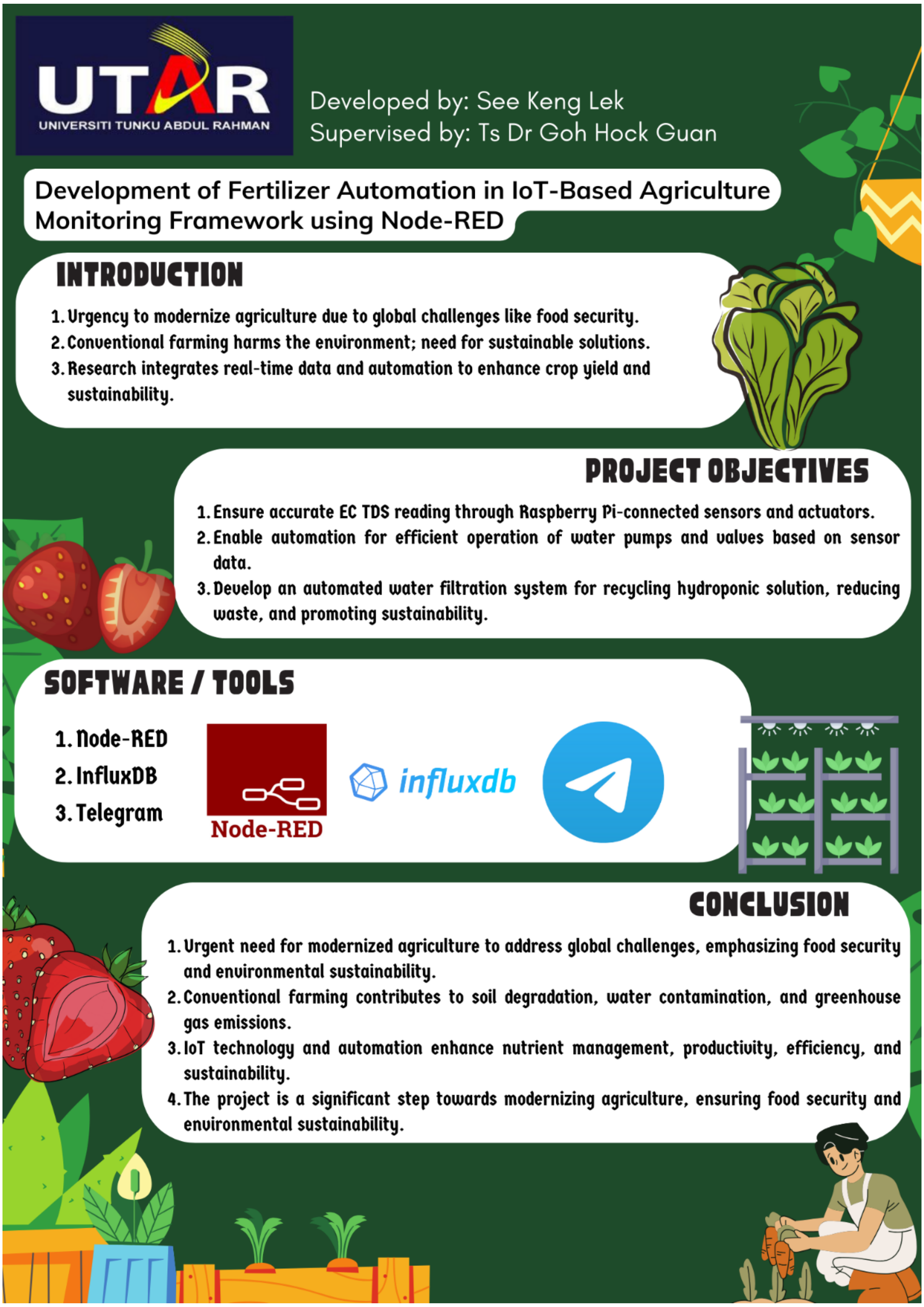
Trimester, Year: June, 2024	**Study week no.: 12**
Student Name & ID: See Keng Lek (20ACB01973)	
Supervisor: Ts Dr. Goh Hock Guan	
Project Title: Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED	

1. WORK DONE Throughout weeks 11 and 12, I persisted in testing the system in the lab and closely monitoring its performance.
2. WORK TO BE DONE Refine the presentation slides for the upcoming presentation.
3. PROBLEMS ENCOUNTERED Fortunately, I did not encounter any significant problems during this period, allowing me to maintain focus and progress smoothly.
4. SELF EVALUATION OF THE PROGRESS Reflecting on the recent progress, I am satisfied with the dedication and effort invested in testing and observing the system's behavior.

Supervisor's signature

Student's signature

POSTER


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Developed by: See Keng Lek
Supervised by: Ts Dr Goh Hock Guan
Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED
INTRODUCTION
1. Urgency to modernize agriculture due to global challenges like food security.
2. Conventional farming harms the environment; need for sustainable solutions.
3. Research integrates real-time data and automation to enhance crop yield and sustainability.
PROJECT OBJECTIVES
1. Ensure accurate EC TDS reading through Raspberry Pi-connected sensors and actuators.
2. Enable automation for efficient operation of water pumps and valves based on sensor data.
3. Develop an automated water filtration system for recycling hydroponic solution, reducing waste, and promoting sustainability.
SOFTWARE / TOOLS
1. Node-RED
2. InfluxDB
3. Telegram
Node-RED
influxdb
CONCLUSION
1. Urgent need for modernized agriculture to address global challenges, emphasizing food security and environmental sustainability.
2. Conventional farming contributes to soil degradation, water contamination, and greenhouse gas emissions.
3. IoT technology and automation enhance nutrient management, productivity, efficiency, and sustainability.
4. The project is a significant step towards modernizing agriculture, ensuring food security and environmental sustainability.

PLAGIARISM CHECK RESULT

2001973_Turnitin.pdf

ORIGINALITY REPORT

9%	6%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	1%
2	D K P Aji, U Nurhasan, R Arianto, O D Triswidrananta. "Smart ecosystem for hydroponic land in the hydroponic farmers group guided by CSR PT. Otsuka Indonesia as an improved quality and quantity of harvest results", IOP Conference Series: Materials Science and Engineering, 2021 Publication	<1%
3	Amir Shachar. "Introduction to Algogens", Open Science Framework, 2024 Publication	<1%
4	iopscience.iop.org Internet Source	<1%
5	maxogden.com Internet Source	<1%

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)

Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	See Keng Lek
ID Number(s)	20ACB01973
Programme / Course	Communications and Networking
Title of Final Year Project	**Development of Fertilizer Automation in IoT-Based Agriculture Monitoring Framework using Node-RED**

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 9 **%** **Similarity by source** Internet Sources: 6 % Publications: 4 % Student Papers: 4 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required, and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Goh

______________________________ Signature of Supervisor

Name: Goh Hock Guan

Date: 11/9/2024

______________________________ Signature of Co-Supervisor

Name: ______________________

Date: ______________________

FYP 1 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP1 THESIS SUBMISSION

Student ID	20ACB01973
Student Name	See Keng Lek
Supervisor Name	Ts Dr. Goh Hock Guan

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgment
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

______________ __________

(Signature of Student)

Date: 10/9/2024