IOT SMART-BASED FLOOD DETECTION SYSTEM

MUHAMMAD AMMAR BIN ISZAMLI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

IOT SMART-BASED FLOOD DETECTION SYSTEM

MUHAMMAD AMMAR BIN ISZAMLI

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Computer Networking) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

IOT SMART-BASED FLOOD DETECTION SYSTEM

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT: ______________________________ Date : 28/8/2024

(MUHAMMAD AMMAR BIN ISZAMLI)

I hereby declare that I have read this project report and found
this project report is sufficient in term of the sope and quality for the award
of Bachelor of Computer Science (Computer Networking) with Honours.

SUPERVISOR: ________________________ Date:28/8/2024

(DR. NAZRULAZHAR BIN BAHAMAN)

DEDICATION

First and foremost, I am grateful to Allah for His abundant blessings and grace, which have enabled me to complete this final year project. I would also like to dedicate this report to my parents and my lecturer for their guidance and support.

To my parents, thank you for providing both spiritual and material support, enabling me to complete this final year project report.

To my friends, thank you for standing by me through thick and thin, and for all the assistance you have provided in the creation of this final year project report.

To my lecturer, Dr. Nazrulazhar, I am immensely thankful and grateful for your guidance, instructions, and patience in helping me complete this report.

I am truly honored and grateful to my parents, lecturer, and friends for their unwavering support, dedication, and trust in me.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to Allah for His blessings and guidance throughout the completion of this final year project. I extend my sincere appreciation to my parents for their unwavering support, both spiritually and materially, which has been instrumental in my academic journey and the completion of this report.

I am deeply thankful to my friends for their companionship, encouragement, and assistance, especially during challenging times. Your presence and support have made this journey more meaningful and enjoyable.

I am also indebted to my lecturer, Dr. Nazrulazhar, for his invaluable guidance, mentorship, and patience throughout the development of this project. His expertise and encouragement have significantly contributed to the quality and success of this report.

Lastly, I would like to acknowledge the trust and confidence placed in me by my parents, lecturer, and friends. Your belief in my abilities has been a driving force behind my motivation and determination to excel.

ABSTRACT

Floods are among the most severe natural disasters, causing enormous damage to infrastructure, agriculture, and human lives. Climate change is increasing the frequency and intensity of floods, necessitating the development of an effective early detection and warning system. This project proposes creating an IoT-based smart flood detection system that continuously monitors water levels in sensitive locations via a network of sensors. The device uses water level sensors to determine the distance to the water's surface and other environmental conditions. When the water level reaches a critical level, the system sends an alert and real-time messages to local authorities and residents via SMS, emails, and mobile apps. This immediate response capability allows for timely evacuation and disaster management, reducing the potential for loss of life and property damage. The system is designed to be scalable, allowing deployment in diverse environments, from urban areas to rural communities and agricultural regions. By providing accurate and timely flood warnings, this project aims to enhance flood preparedness, improve disaster response, and safeguard both people and assets.

ABSTRAK

Banjir adalah salah satu bencana alam yang paling mengerikan, banjir merosakkan infrastruktur, pertanian, dan kehidupan manusia. Perubahan iklim menyebabkan banjir semakin kerap dan teruk. Sistem pengesanan dan amaran awal yang berkesan adalah perlu sekarang. Sistem pengesanan banjir pintar berasaskan Internet of Things (IoT) akan menggunakan rangkaian sensor untuk memantau tahap air secara berterusan di kawasan yang terdedah kepada banjir. Sensor paras air digunakan dalam sistem ini untuk mengukur jarak ke permukaan air dan ciri persekitaran lain. Sistem ini akan mengeluarkan amaran dan menghantar pemberitahuan dalam masa nyata kepada penduduk dan pihak berkuasa tempatan apabila paras air mencapai ambang kritikal melalui pelbagai saluran komunikasi, seperti aplikasi mudah alih, e-mel dan SMS. Keupayaan untuk bertindak balas segera ini memudahkan evakuasi dan pengurusan bencana yang tepat pada masanya, mengurangkan risiko kehilangan nyawa dan kerosakan harta benda. Sistem ini direka untuk berskala, membolehkan penyebaran di pelbagai persekitaran, dari kawasan bandar hingga komuniti luar bandar dan kawasan pertanian. Dengan menyediakan amaran banjir yang tepat dan tepat pada masanya, projek ini bertujuan untuk meningkatkan kesiapsiagaan banjir, memperbaiki tindak balas bencana, dan melindungi nyawa serta aset.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP - Final Year Project

IoT - Internet of Things

GSM - Global System for Mobile Communication

DHT - Digital Humidity Temperature

JPS - Jabatan Pengairan dan Saliran

GPIO - General Purpose Input/Output GPS - Global Positioning System

GSM - Global System for Mobile Communication

IDE - Intergrated Development Environment

PC - Project Contribution

PO - Project Objective

PQ - Project Question

PS - Problem Statement

CHAPTER 1: INTRODUCTION

1.1 Introduction

Flood is an unavoidable natural disaster that been happen in all over the world. Flood give a lot of effect, and we need to take prevention to avoid it from occur. It is causing heavy flow of water and can cause damages. Flood cannot be stopped so early warning and detection system is needed to reduce any damages from happening. We need to create flood level sensing devices that can help to detect the water level. The system that is going to be developed is named IoT Smart-Based Flood Detection System.

This IoT flood detection system consists of embedded electronics, software, and sensors that can send and receive data remotely over the internet. The primary objective of this system is to construct flood level sensing devices that can aid to detect the water level and include alert system that easier to inform the community to take precaution that flood is coming. The data of the water level will be store on the website thus it will be easy for the prediction of future uses.

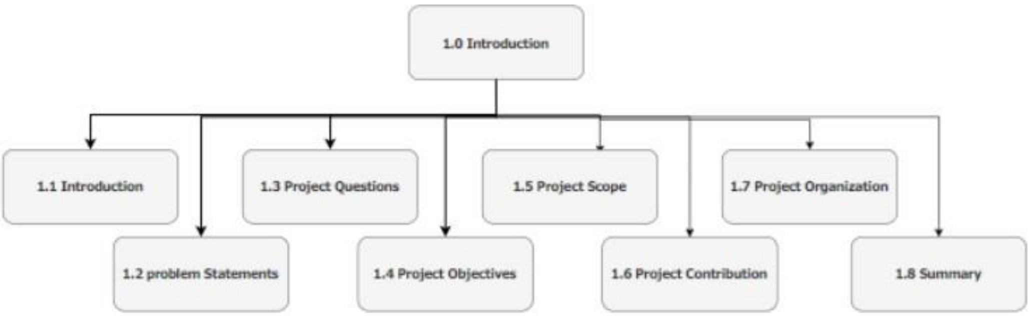


Figure 1.1 Structured Framework

1.2 Problem Statement

According to article from Davies, R. (2023, January 25). Malaysia – Floods in Johor and Pahang After 430mm of Rain in 24 Hours. Malaysia's experience 344 mm rain in one day on January 24 and this affected that 5 rivers in Malaysia are above danger threshold. This shown that flood detection and monitoring system in Malaysia not in a successful project. To summarize this, the problem statements for this project are shown in Table 1.1.

Table 1.1 Problem Statement

PS	Problem Statement
PS1	Hard to collect and interpret real-time flood pattern from the normal environment so the used of the real sensor is important.

1.3 Project Question

The project research question is used to identify how we can develop an IoT Smart-Based Flood Detection System that is efficient and reliable with real-time communications. Based on a few kinds of research, it can be concluded that there are some difficulties and challenges using manual methods in monitoring the flood and water level. Table 1.2 shows the summary of the project research question.

Table 1.2 Summary of the Project Question

PS	PQ	Project Question
PS1	PQ1	What are the main problems that occur in the existing system and what methods have been used to address these issues?
	PQ2	What are the key components and features required to develop an effective IoT flood detection system for detection and monitoring of floods?
	PQ3	How effective is the system in achieving its intended objectives, based on the evaluation of relevant parameters and statistics?

1.4 Project Objective

Project objectives are tailored to the problem statement and project questions. The objective of this project is shown below. For this project, there are 3 objectives that need to be achieved:

Table 1.3 Summary of Project Objectiives

PS	**PQ**	**PO**	**Project Objective**
PS1	PQ1	PO1	To study the problem that occur, and current solution used in the existing system.
	PQ2	PO2	To develop the IoT Smart-Based Flood Detection System that could help in detection and monitoring flood.
	PQ3	PO3	To evaluate the effectiveness of the proposed system based on the parameter that been involved.

PO1: To study the problem that occurs, and current solution used in the existing system.

To differentiate and figure out sensor that been used in existing system that affects the accuracy of the data collected. A comparison between the existing system with the proposed system is needed to choose the right mobile application.

PO2: To develop the IoT smart-based flood detection system that could help in detection and monitoring floods.

To complete prototype that can integrated to the website application. The website monitor can alert the increasing of the water level at a certain threshold, and it can alert the authorities about the flood that will coming.

PO3: To evaluate the effectiveness of the proposed system based on parameters that have been involved.

The data of parameter will display on the website and the data have been recorded for the authorities. This parameter that has been set up needs to be saved in case anything happens so it can be used for future needs.

1.5 Project Scope

To reduce casualties and the impact of a disaster, disaster management is crucial when it is about to occur. Disaster management is essential for Malaysia, a nation that experiences year-round torrential rainstorms. Despite geographical protection from events like earthquakes and volcanic eruptions, Malaysia had witnessed disastrous floods, and the effects were severe. Malaysia had some of the greatest rainfall in 2010, with an average of 2,500 millimeters across the entire nation. Since heavily urbanized places like Kuala Lumpur and Georgetown are more prone to experience the tragedy, overdevelopment appears to be the primary cause of floods. This is frequently linked to heavy rainfall, insufficient drainage, and inappropriate trash disposal.

The scope of this IoT Smart-Based Flood Detection System is to easier authority at Jabatan Pengairan dan Saliran (JPS) to monitor the environment condition such as detect increasing or decreasing in water distance and changes in humidity and temperature level that have possibility for flood to happen. This system comes with an android application that shows the changes in the environment conditions and alerts the user if a flood is coming.

1.6 Project Contribution

Project Contribution defines the expected output from this project. This project can help community to become more alert with the flood in Malaysia. The main goal of this project is to monitor and detect the increasing of the water level at certain threshold.

The alert system will notify the community that flood is coming, and the data of the certain parameter will be kept in the cloud so it can be used for future used to predict the natural disaster. The project contribution can be referred to the table 1.4 below.

Table 1.4 Summary of Project Contribution

PS	**PQ**	**PO**	**PC**	**Project Contribution**
PS1	PQ1	PO1	PC1	Finalize the problems that been occur and method that been used in the existing system
	PQ2	PO2	PC2	Finalize the correct software and hardware that can help in detection and monitoring flood
	PQ3	PO3	PC3	Finalize the data that been obtain in certain parameter is effective to used

1.7 Report Organization

Chapter 1: Introduction

This chapter discusses the purpose of developing the IoT Smart-Based Flood Detection System includes project background, problem statement, project question, and project objective to clarify the intention of the system.

Chapter 2: Literature Review

This chapter discusses other topics that have similar fields and can correlated to this project. Previous work or related work that using different tools and methods been compared. In this chapter will make changes to the existing work and identify the improvements needed in this project.

Chapter 3: Methodology

This chapter discusses the method used to perform this project. Every stage of methodology will be recorded in this chapter. Milestones of this project will be determined in this chapter to make well-plan time to complete a certain task or phase of this project.

Chapter 4: Design

This chapter will cover all the requirements that need to be analyzed for the project which include the Data Requirement, Functional Requirement, and Non-functional requirement. The results of the analysis and detailed design will be described. This chapter can be considered important where it is necessary to formulate a good analysis and design to produce a successful project.

Chapter 5: Implementation

In this chapter contain the implementation of the project design that hasbeen determined previously. The system development begins using the required software and hardware. The implementation status and progress for each of the components or module have been described.

Chapter 6: Testing

Throughout this chapter, the testing will be made based on test plan which are test organization, test environment and test schedule. All the results of the test will be recorded in this chapter and analyzed. This is where the system will be analyzed and decide whether this project is successful.

Chapter 7: Project Conclusion

In this chapter, the conclusion for this project by explaining how the objective has been achieved based on the implantation and testing phase. All the results obtained will be concluded including the weakness and strength of this project.

1.8 Summary

IoT Smart-Based Flood Detection Systems offer significant advantages in detecting and responding to flood events. These systems utilize sensors, data analytics and responding to flood events. These systems utilize sensors, data analytics and connectivity to provide real-time monitoring of water levels, weather conditions and potential flood risks. By collecting and analyzing data, these systems can promptly detect flood situations, alert the event to the user, and trigger appropriate actions to mitigate the damage caused by floods.

Overall, IoT Smart-Based flood detection system has the potential to enhance early warning capabilities, improve emergency response times and ultimately reduce the impact of floods on communities and infrastructure. For the next chapter, the literature review chapter provides a comprehensive summary of the existing knowledge and research related to the project topic. It serves to establish a solid foundation by reviewing relevant studies, scholarly articles, books, and other sources.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

A literature review is a citation that refers to a topic relevant to the selected research subject. A literature review is a collection of ideas, mathematical methods, and methodological contributions on a particular subject. This chapter will focus on developing a development plan for IoT Smart-Based Flood Detection System. This document includes all the sensor and application descriptions and data. We will also discuss the earlier research on the flood sensor in this chapter. It will examine their implementation methods, constraints, and strategies and draw comparisons between the present and previous projects. Many crucial factors may be used to determine the appropriateness of an existing flood detection system for the research material while doing this evaluation. The system or hardware must be built using Internet of Things (IoT) technologies.

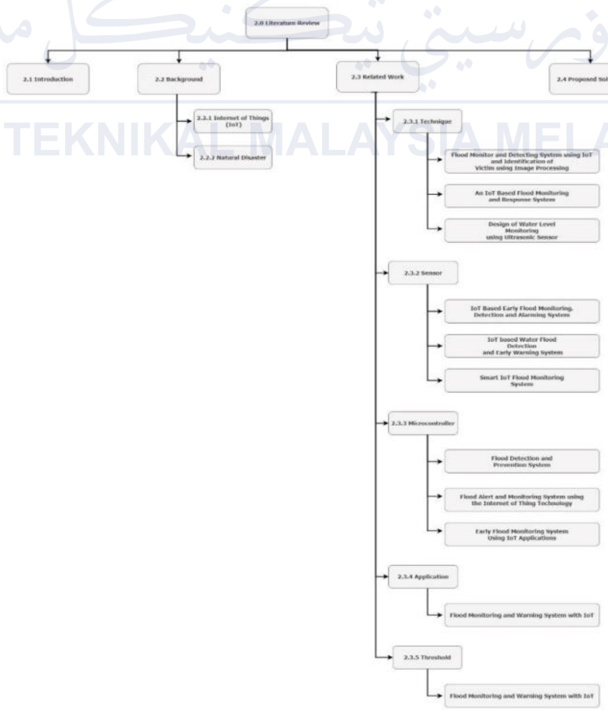


Figure 2.1 Outline of Literature Review

2.2 Background

2.2.1 Internet Of Things (IoT)

Internet of things is an innovation in Malaysia that have embedded software, hardware, sensors, and other technologies that exchanging data between the internet. There are three characteristics of internet of things which is sensing devices that help to enhance human sensory ability, dual communication and intelligence that can react with any interaction. Flood in Malaysia is a significant issue that occur in Malaysia and Malaysian government has taken measures to lessen the effects of floods, including installing early warning systems and developing flood mitigation infrastructure. However, the measure that has been taken has not been seen successfully until now. IOT devices like sensors could be utilized by rescue crews to monitor the whereabouts of flood victims. These interactions can be facilitated by IOT, which makes it possible for several devices to be linked to the Internet at once. Natural disasters that happen due to floods can be avoided with the help of IoT.

The IoT architecture consists of three layers which is perception layer which consists of sensor or actuators, next is network layer which consists of routers or gateway and last is application layer which involve cloud and database. These three layers is important and needed between each other. The lowest level is perception layer that contains embedded devices such as sensors that are needed to detect and response towards the physical input such as temperature, humidity, motion, etc. This physical input next will be converted to the significant data that involve in the next layer which is network layer. Network layer helps in traverse the sensed data through wired or wireless communication to transferring it toward the database or clouds. Finally, the application layer is the last layer that helps in specialized data storage. This application layer can be mobile application, web-based application, or cloud application. This layer transforms the data into the form of data analytic that help for future used that been kept in the cloud.

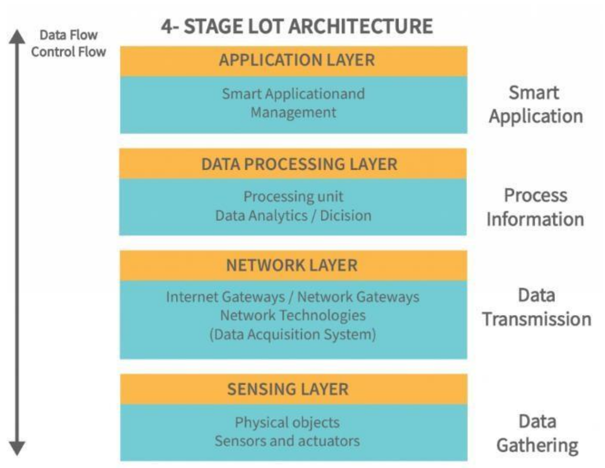


Figure 2.2 IoT Architecture

2.2.2 Natural Disaster Management

Natural disaster management entails a series of connected steps. Using a range of sensors and techniques, detection and monitoring comprises detecting early warning signs of impending disasters, continuously monitoring changing conditions, and alerting individuals in danger. While mitigation approaches, such as building codes and hazard reduction, aim to diminish the impact of disasters, prevention strategies aim to reduce the risk of disasters occurring. Planning for preparation, exercising, and gathering supplies are all examples of preparedness. When a crisis strikes, the priority is to save lives and provide basic services, followed by efforts to rebuild and restore normalcy. Finally, a critical review of preparedness and mitigation approaches informs future developments in disaster management tactics. Together, these stages contribute to lessening the risks and effect of natural disasters.

In Malaysia, floods are a frequent natural crisis, especially during the monsoon season, which normally lasts from November to February. Floods have the potential to kill people and seriously harm crops, houses, and infrastructure. In 2021, Malaysia saw its worst floods in decades, resulting in 54 fatalities and the call-up of the army. Eight states were affected by the extensive flooding that year, which put a strain on emergency services nationwide and led to criticism of the government's reaction to the crisis. The second-most populous state in Malaysia, Johor, has a population of 4 million people and has been hardest damaged by this season's floods. According to officials, tens of thousands of its citizens have now relocated to relief centres at schools and community centres. The Malaysian Meteorological Department's experts have issued a warning that the rainy weather could last into April.

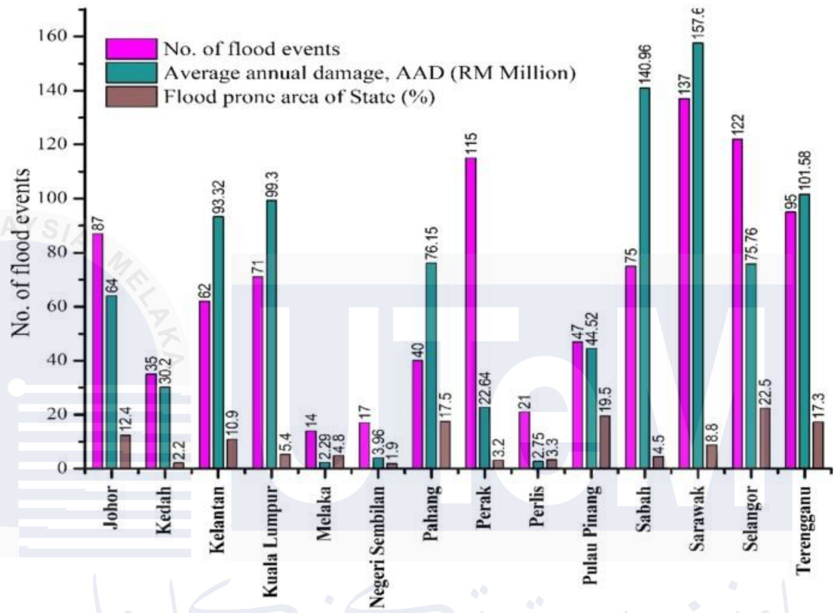


Figure 2.3 Flood Events

2.3 Related Work

2.3.1 **Network** Architecture

The flood detection system is made up of data processing and data transmission. The technique that is usually used is by using sensor that can detect the increasing and decreasing of the water level in real time such as ultrasonic sensor and water level sensor. Next is processing data, the data that has been captured is sent to the cloud by using microcontrollers such as ESP32, ESP8266 and Arduino UNO. After the data that has been captured is sent to the cloud server, the use of the android application is needed to keep the data for future use. The android applications that are usually used are Blynk application and Thing Speak. Lasty, an alert system is needed to notify the community about the water level. There are types of alert systems that been used such as buzzer, led, GSM and android application. The use of android applications is also helping to notify the community when the flood is coming. The article below shows how a different technique is used in flood detection systems.

A comprehensive system for early flood monitoring is presented. This system leverages Internet of Things (IoT) technology and consists of several key components, including the Wi-Fi ESP module, various sensors (rainfall, water level, and flow sensors), a power supply, and an Arduino as the central hub for device connectivity. The power supply ensures that the sensors receive the necessary electricity to function effectively. The Arduino collects and processes data from these sensors and facilitates interconnectivity among all system components. Notably, the system provides consistent updates through message alerts and displays real-time findings on a liquid crystal display (LCD). This research is valuable for addressing the critical issue of flood monitoring and early detection, Keerthan et al. [2]

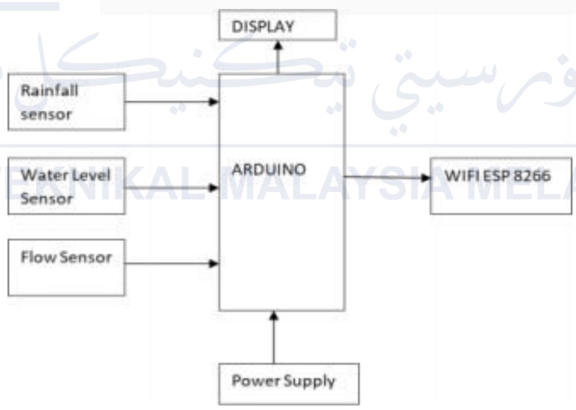


Figure 2.4 Figure 2.4 Flood Monitor and Detecting System using IoT and Identification of Victim using Image Processing (Keerthan et al., 2020)

Yiiong et al. [3] present a flood monitoring and response system that utilizes Internet of Things (IoT) technology. This system employs two key sensors: an ultrasonic sensor and a raindrop sensor. The raindrop sensor provides three signals, which are used to calculate the severity of the downpour. To facilitate understanding, the system uses color-coded indicators, with green representing safe conditions, yellow indicating a warning, and red denoting hazardous levels. The microcontroller used in this system is the Arduino UNO, which is connected to a Wi-Fi module for data transmission. An alarm buzzer is activated only when the alert level reaches the 'danger' stage, signifying a critical flood situation. The system reads sensor inputs and utilizes a web interface to store and display real-time flood data.

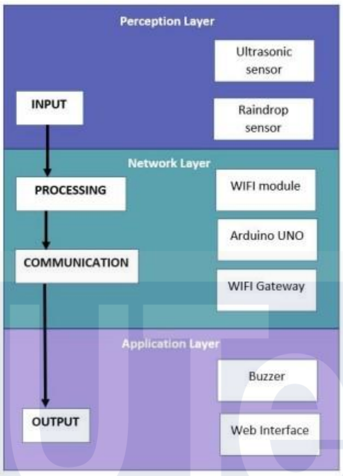


Figure 2.5 An IoT Based Flood Monitoring and Response System (Yiiong, Bundan, & Chew)

Baskhara et al. [4] present a system for early flood monitoring that utilizes Internet of Things (IoT) technology. This system relies on an ultrasonic sensor capable of detecting ultrasonic waves and transmitting the data as analog signals to a Raspberry Pi. To assess water movement, the system utilizes a servo motor tool. The data captured by the sensor is then displayed on an LCD screen, facilitating real-time monitoring of water levels, and enabling early flood detection.

2.3.2 Sensor

In internet of things, sensor is part of the important role that is really needed in order the development system works well. A sensor is a tool that can detect any external information that has been received from the environment and replace it with a signal that humans and machines can recognize. IoT smart-based flood detection system using ultrasonic sensor is the most crucial part to make sure this system works perfectly. An ultrasonic sensor is a piece of technology that uses ultrasonic sound waves to detect a target object's distance and then turns the sound that is reflected into an electrical signal. Ultrasonic waves move more quickly than audible sound that people can hear. The ultrasonic sensor has two main components which are transmitter and receiver. The transmitter will produce the sound, and the receiver will encounter the sound after it has gone to and from the target. The article below shows the use of different sensors in flood detection systems.

P et al. [5] present a system for early flood monitoring that harnesses Internet of Things (IoT) technology. The system's primary objective is to detect floods by monitoring various natural factors, including humidity, temperature, water level, and flow level. To gather data on these factors, the system incorporates different sensors. For humidity and temperature monitoring, a DHT11 Digital Temperature Humidity Sensor is used, which includes resistive humidity and temperature detection components. To track water levels, a float sensor is employed, operating on the principle of circuit completion when water levels change. For measuring distances, an HC-SR04 Ultrasonic Range Finder Distance Sensor is utilized, working based on SONAR principles. All these sensors are connected to an Arduino UNO, which processes and stores the collected data.

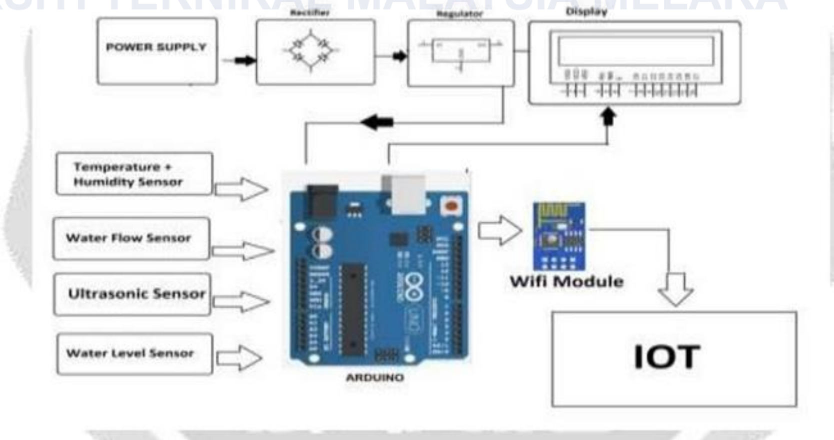


Figure 2.6 IoT Based Early Flood Monitoring Detection and Alarming System (P et al., 2020)

Chitra et al. [6] present a system for early flood monitoring that relies on Internet of Things (IoT) technology. This system extracts data from a water level sensor positioned near the water body, employing several sensors to collect data on various natural elements. For humidity and temperature monitoring, the system utilizes a DHT11 Digital Temperature Humidity Sensor, which comprises resistive humidity and temperature sensing components. To continuously monitor water levels, a float sensor is employed, functioning by opening and closing circuits (dry contacts) in response to changes in the water level.

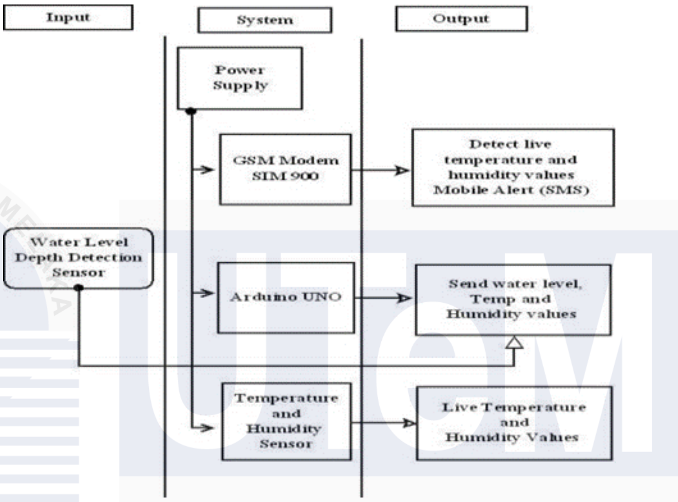


Figure 2.7 IoT based Water Flood Detection and Early Warning System (Chitra et al., 2020)

Zahir et al. [7] a system for early flood monitoring that utilizes Internet of Things (IoT) technology is presented. Arduino serves as the central controller of the system and is interfaced with a GSM modem and an ultrasonic sensor. The ultrasonic sensor, connected through the microcontroller, plays a crucial role in detecting changes in water levels, providing valuable data for flood monitoring.

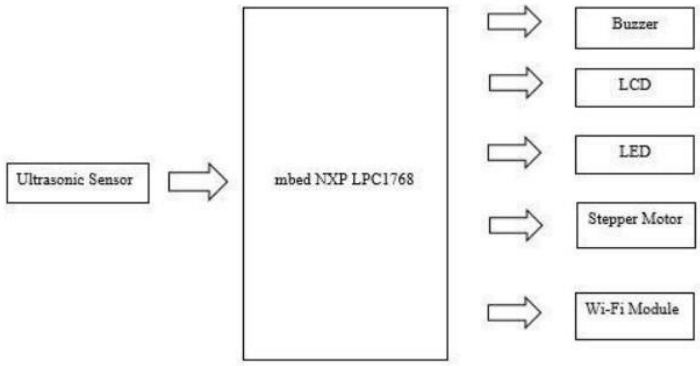


Figure 2.8 Smart IoT Flood Monitoring System monitoring (Zahir et al., 2019)

2.3.3 Microcontroller

In building an internet of things project, microcontroller is a crucial part in order the system can be successfully work. There are 3 types of microcontrollers that are usually used in IoT projects such as Arduino Uno, ESP32 a Raspberry Pi. These three types of microcontrollers have the same function, but the usage is different. For raspberry pi, the features a more powerful processor, more RAM, and onboard storage, allowing it to handle complex tasks and run various applications. Raspberry Pi has general-purpose input/output (GPIO) pins that enable you to connect and control external electronic components. For Arduino Uno, it has simpler architecture compared to raspberry pi. To use Arduino uno as microcontroller, wi-fi module is needed to connect with it. Lastly is ESP32, this microcontroller is an innovation that has built in wi-fi module that make it easier to use compared to Arduino uno and raspberry pi. These three types of microcontrollers have different capabilities in functioning, so the above article shows how it is used in some of the flood detection systems.

Ahmad and Suhaimi [8], a system for early flood monitoring utilizing Internet of Things (IoT) technology is introduced. This system employs real-time sensors as a flood control measure, providing both flood monitoring and alert systems. The project utilizes the Node MCU ESP32 as a microcontroller connected to two sensors to achieve its objectives.

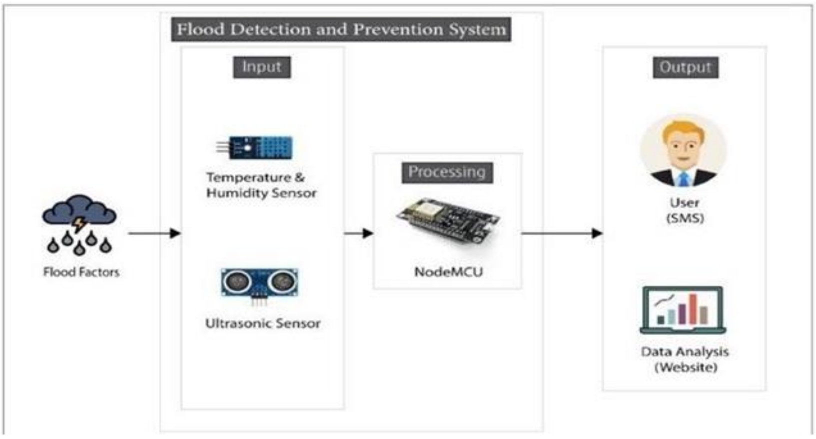


Figure 2.9 Flood Detection and Prevention System (Ahmad & Suhaimi, 2022)

Hamid et al. [9] a system for early flood monitoring employing Internet of Things (IoT) technology is presented. The system comprises a power source, a rain intensity sensor, and a water level sensor. These sensors, including the water level measurement sensor node and the rain volume sensor node, are strategically located away from the control unit. Data from raindrop and water level measurements are collected using Arduino Uno and Node MCU, and then transmitted to an IoT platform like ThingSpeak for monitoring.

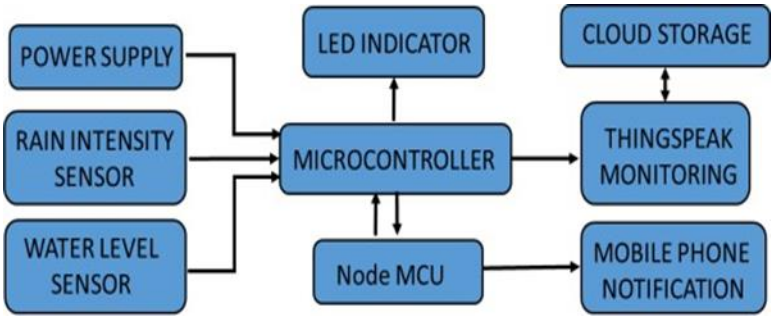


Figure 2.10 Flood Alert and Monitoring System using the Internet of Thing Technology (Hamid et al., 2022)

Suresh et al. [10] a system for early flood monitoring employing Internet of Things (IoT) technology is introduced. The system utilizes an Ultrasonic sensor connected to an Arduino Uno to detect water levels. Further, the Arduino Uno is linked to both an LED and a Buzzer. Serial communication is established between the

Arduino Uno and a Raspberry Pi, which is in turn connected to the cloud for data transmission and monitoring.

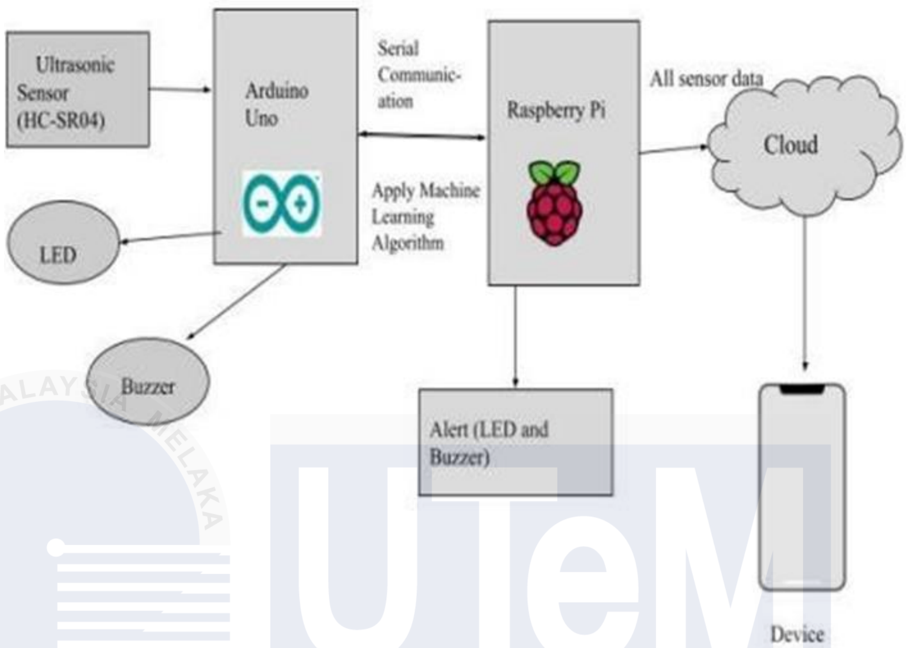


Figure 2.11 Early Flood Monitoring System Using IoT Applications (Suresh et al., 2020)

2.3.4 Application

Web dashboards and mobile apps play crucial roles on the Internet of Things (IoT) ecosystem. Users of mobile apps can view data in an accessible format, receive real-time alerts, and remotely operate, monitor, and adjust their IoT devices. They are appropriate for on-the-go, brief conversations. Web dashboards, on the other hand, provide cross-device control, remote management of bigger IoT deployments, and more thorough and detailed data analysis capabilities. They offer scalability and strong security measures and are accessible from a variety of devices using web browsers. Both mobile apps and web dashboards are essential for controlling and successfully interacting with IoT devices and data, depending on the individual demands of IoT users.

Haslina et al. [11] demonstrated the capability of the IoT platform Blynk. This platform enables users of both iOS and Android smartphones to effectively operate

Arduino and Raspberry Pi devices without requiring any additional hardware. By connecting to the internet via Wi-Fi, Blynk offers users the flexibility to craft unique dashboards with features like buttons, sliders, charts, and other widgets, tailored to the management of their specific IoT projects. Importantly, users have the option to control the visibility of sensor data as needed. Blynk further simplifies the development of hardware project applications by providing developers with an intuitive drag-and-drop virtual dashboard for building visual interfaces. Additionally, it facilitates template creation for diverse objects, streamlining management and monitoring tasks through the Blynk software.

2.3.5 Threshold

Setting proper thresholds is essential for efficient flood monitoring in an IoT flood detection system. When certain metrics or sensor readings cross these thresholds, which are specified values or conditions, alerts or actions are initiated. Water level thresholds, which signal rising water levels and the risk of flooding, and rainfall thresholds, which track rainfall volumes and patterns, are important thresholds. While sensor health thresholds provide sensor dependability, rate of change thresholds can detect sudden changes in water levels.

Haslina et al. [11] demonstrated the system has an LCD display and an LED indication for keeping track of the water level. The water level is displayed on the LCD and is divided into four states: low, normal, high, and too high. The LED switches from green for low water levels (0 cm to 2 cm), yellow for moderate levels (3 cm to 7 cm), and red for high levels (8 cm to 10 cm), depending on the water level condition. Importantly, when the measurement goes over 10 cm, the water level is considered to be "Water Level too High," and an alert is sent. This configuration makes it possible to monitor water levels clearly and visually, with the alarm acting as a safety safeguard when water levels become dangerously high.

Water level (cm)	Display	LED Colour	Alert
0-2	Water Level Low	Green	Off
3-7	Water Level Normal	Yellow	Off
8-10	Water Level High	Red	Off
11	Water Level Too High	Red	On

Figure 2.12 Flood Monitoring and Warning System with IoT (Awang Sufa, Yusof, & Afira Sani, 2019)

2.3.6 File Management

Choosing the best file management system is essential for efficient flood monitoring in an IoT flood detection system. This includes creating directories or folders to categorize different types of data such as sensor readings, system logs, and configuration files for thresholds and alerts. Each type of data should be stored in a logical and easily navigable structure to facilitate quick access and retrieval during analysis or troubleshooting. Similar to sensor thresholds, file management involves defining criteria for data storage and archiving. For instance, setting thresholds for file size or storage duration ensures that data remains manageable and relevant over time.

[17] In the context of the flood detection system described in the article from MDPI, the cloud server plays a pivotal role in facilitating efficient data management and real-time monitoring capabilities. The cloud server serves as a centralized repository where sensor data, including measurements from ultrasonic and water level sensors, is securely transmitted via microcontrollers such as Arduino UNO and ESP32. This data is processed and stored systematically, enabling continuous monitoring of critical parameters such as water levels and rainfall intensity. The cloud infrastructure supports seamless integration with IoT platforms like Blynk and ThingSpeak, allowing stakeholders to access comprehensive data visualizations and receive timely alerts through web and mobile interfaces. By leveraging the scalability and flexibility of cloud computing, the flood detection system not only enhances its responsiveness to changing environmental conditions but also ensures robust data accessibility and reliability for effective flood monitoring and early warning dissemination.

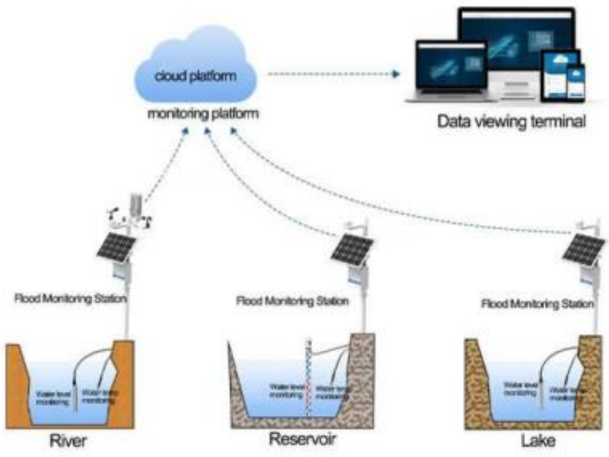


Figure 2.13 Flood Monitoring and Warning System with IoT (Renkeer, 2019)

Table 2.1 Statistics Past Project

Source/Paper (Author)	Sensor			Microcontroller			Comm tech	Alert system			Database system	
	ultrasonic	humidity	water flow	Arduino	Node MCU	Raspberry Pi	Wi-Fi Module	Circuit	GSM	Buzzer / LCD	APP	Web int
Neha Suresh et. Al (2020)	/			/		/	/			/	/	
Keerthan et. Al (2020)	/		/	/			/			/		
Soubhagya et. Al (2020)	/	/		/			/	/		/		/
N. A Hamid1 et. Al (2022)	/			/	/		/			/	/	/
Siew-Ping Yiiong et. Al (2022)	/		/	/			/			/		/
Shahirah Binti Zahir et. Al (2019)	/			/			/			/		/
Hedy Aditya Baskhara et. Al (2021)	/					/		/		/		
Mohamad Luqman Ahmad et. Al (2017)	/	/			/							/
M. Chitra et. Al (2020)	/	/		/					/			/

2.4 Proposed Solution

Based on the related work on the IoT smart-based flood detection system, there a few aspects specification that needed to be developed of this project. From the literature review that have been proposed, the most important aspect of the hardware and software in their project will be develop using the same method but in different implementation.

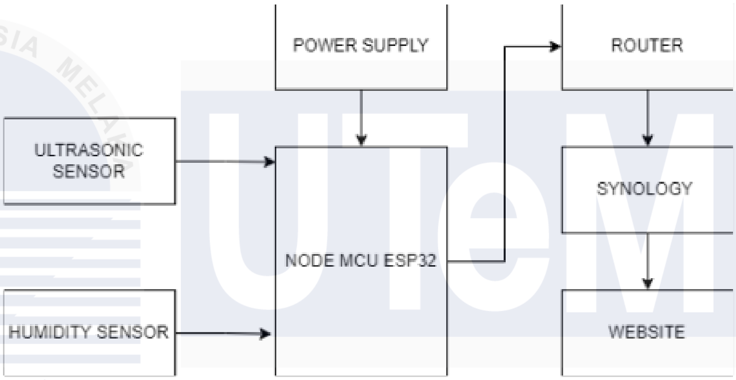


Figure 2.14 Proposed Solution Architecture

Based on the figure above, the microcontroller that been chosen is NodeMCU ESP32 because it has built in Wi-Fi in there compared with Arduino uno and raspberry pi that need Wi-Fi module to communicate. NodeMCU ESP32 and Arduino uno have same method to power on by using USB connection and external power supply but for NodeMCU ESP32, it has supporting battery operation which allow it to be portable application. Next is the choice of sensor, ultrasonic sensor is chosen between water level sensors because ultrasonic sensor has wide measure range compared to water level sensor that designed to measure water level in specific container. It also provides high accuracy and resolution while water level sensor has the same accuracy, but the resolution is limited. Humidity sensor is used to detect changes in humidity levels caused by water leaks or flooding. It serves as an early warning mechanism to detect the increasing of water level reaches the surface. Lastly, android application is chosen because it is easier compared with web-based application. Android applications are developed through android operating system so it easier to access through our own phone while web-based application needs to access through the server, so it takes time to access.

Table 2.2 Data Taken

Data Condition	**Water Distance (cm)**	**Reading's Taken (s)**	**Alert Message**
Humidity >80	< 4cm	2s	Water Level Is at Risk
	Between 6cm and 8cm	6s	Water Level Is Normal
Humidity < 80	Between 6cm and 8cm	6s	Water Level Is Normal

2.5 Summary

To generalize, this chapter is very important for the developer in terms of gaining more information and understanding for implementing the process of the project. All the information acquired will assist developer in constructing and planning the project. In relation to this project, the information regarding hardware and software being implemented will be re-implemented with a different scope in this project.

In the next chapter of project methodology, will be described the methodology will be described to develop this project that explain in detail including the project timeline which will be displayed in a Gantt chart.

CHAPTER 3: METHODOLOGY

3.1 Introduction

This chapter is important to comprehend the project's flow and development. The phases of the methodology contributed to the smooth running of the project process. This section also covers the research methodology to achieve the project's goals, as well as the project's philosophy, tools, and resources.

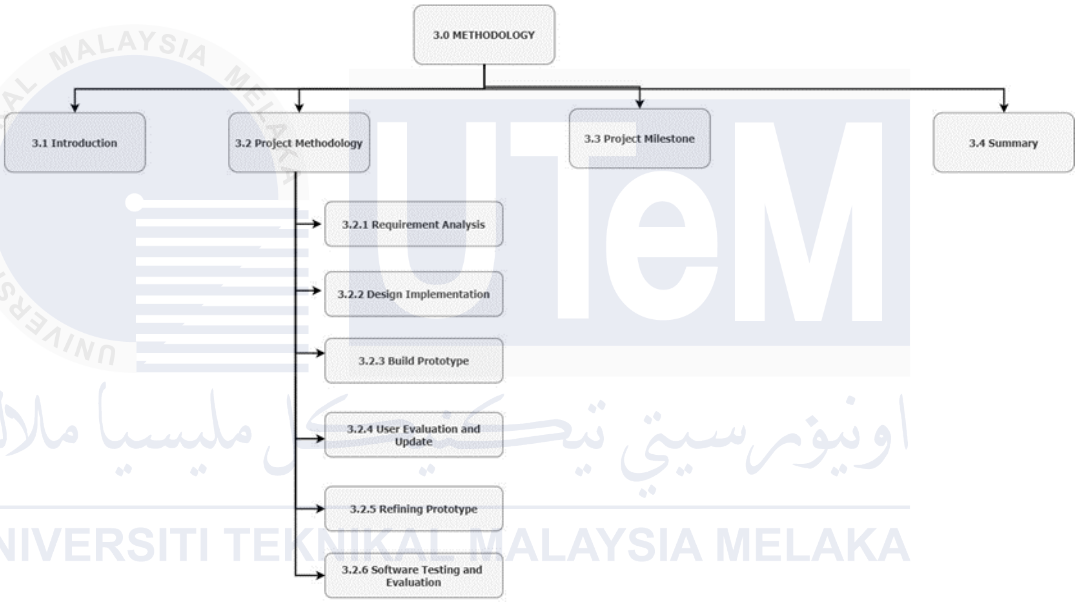


Figure 3.1 Outline of Methodology

3.2 Project Methodology

The methodology can be described as a sequence of phases that were utilized to detail the project development process. The actions that were carried out during the project's operation were covered in further detail in the methodology section. The Prototype model was chosen as the best methodology. This prototype model allows for flexibility and adaptation for evolving requirements while continuing improvement.

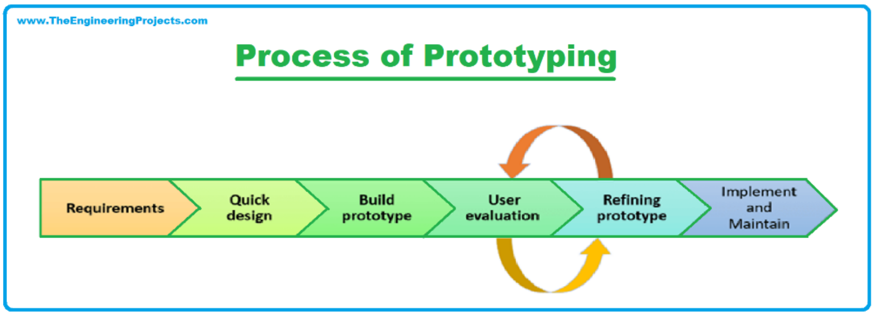


Figure 3.2 Prototype Model

3.2.1 Phase 1: Requirement Analysis

In phase 1, start by gathering the initial requirements for flood detection system. It is needed to understand the desired features, sensors, data collection, data processing and notification mechanisms. However, since requirements may change the project progress, we need to focus on capturing essential functionalities needed. We also assess the effort required to implement the monitoring system, considering factors such as sensor integration, data transmission, and analytics. This meticulous planning ensures a solid foundation for the project as we move forward.

3.2.2 Phase 2: Design Implementation

In phase 2, some literature review is happening to determine about the past system. A Feasibility study is needed to compare some of the software and hardware related to the system so from that we can know which part is needed to be improved in our system comparing with the past system. This design is not comprehensive but focuses on the essential system components, offering a high-level overview of how the system will function.

3.2.3 Phase 3: Build Prototype

In phase 3, we design how the architecture is designed with some schematic diagram that help to build how the prototyping looks like. This phase combines communication modules, IoT platforms, and sensors. Concentrate on key flood detection features like water level measurement. This involves connecting the humidity sensor and ultrasonic sensor to the NodeMCU esp 32 microcontroller. The NodeMCU will be linked with synology nas to save data processing and communication. In simpler terms, Synology, a tool that will handle the data collected by the NodeMCU. This integration allows for easier management of sensor data and device control, facilitating smooth communication and processing within the IoT system.

3.2.4 Phase 4: User Evaluation and Update

In phase 4, need to do some evaluation with the user about the software designing and prototyping whether it meets user requirement or not. This feedback is valuable for refining and enhancing the system in subsequent iterations. Based on the feedback received from the previous phase, necessary improvement and enhancement is needed to the flood detection system. This may involve adding new features, integrating additional sensors, or improving user interface. Each iteration should aim to add value and bring the system closer to the desired final state.

3.2.5 Phase 5: Refining Prototype

Once the flood detection system has reached a stable state, it can be deployed in the target environment. Deploy the system, configure it according to the deployment requirements and ensure proper integration with IoT infrastructure. To make sure the prototyping is functional, coding is much needed, so the output really meets the objectives.

3.2.6 Phase 6: Software Testing and Evaluation

In phase 6, conduct thorough testing of the developed functionalities to ensure they meet the desired requirements. Test the system with simulated or real flood scenarios and gather feedback from users. Some evaluations are needed to improve the system while testing the detection system. To make sure it runs smoothly and does not break down often, we check it thoroughly before doing regular maintenance.

3.3 Project Milestone

Table 3.1 Project Milestones

Phase	**Activity**	**Duration**
Requirement Analysis	• Gathering requirement • Research based on project	3 weeks
Design Implementation	• Build Architecture • Build System Flowchart • Build Connection	3 weeks
Build Prototype	• Build prototype. • Connection sensor • Writing the code	5 weeks
User Evaluation and Update	• Perform Testing • Collect Feedback	3 weeks
Refining Prototype	• Review Feedback • Improve based on feedback	3 weeks
Product Implementation	• Build Final Prototype • Evaluating the system	5 weeks

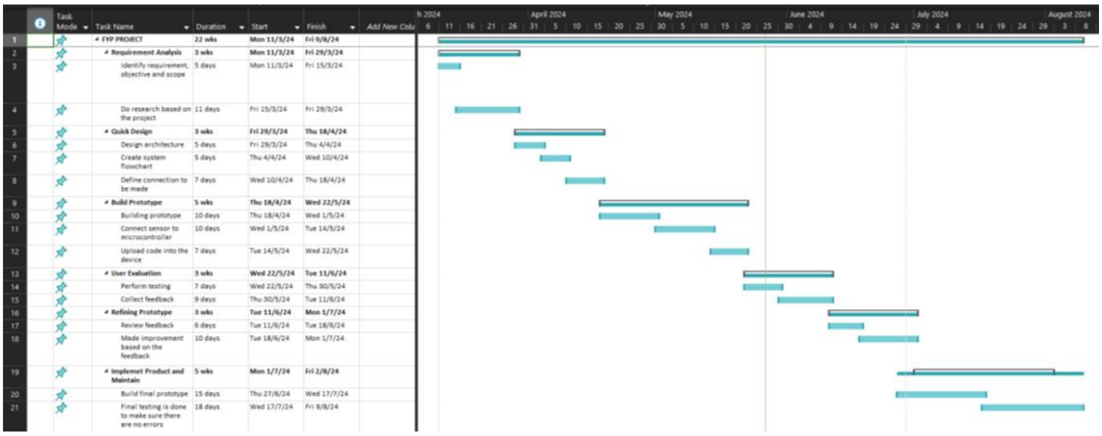


Figure 3.3 Gantt Chart

3.4 Summary

In conclusion, this chapter describes the approaches methodologies to create the project. The prototype model was applied for this research. This chapter will go over five phases that is planning and requirement gathering, design and development, implementation and testing, iterative enhancement and lastly is deployment and maintenance. Finally, all these procedures are necessary to determine the project's overall progress.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter is about how the system will be designed and explains how it will be implemented using problem analysis, system architecture and requirement analysis. The analysis and design phase are a critical stage in the software or product development process. It involves understanding and documenting requirements, and then transforming them into a detailed design that serves as the roadmap for implementation. This chapter will analyze and identify the software and hardware that has been used whether it meets the specific requirements, as well as the main problem of the analysis to achieve a better idea in completing the system.

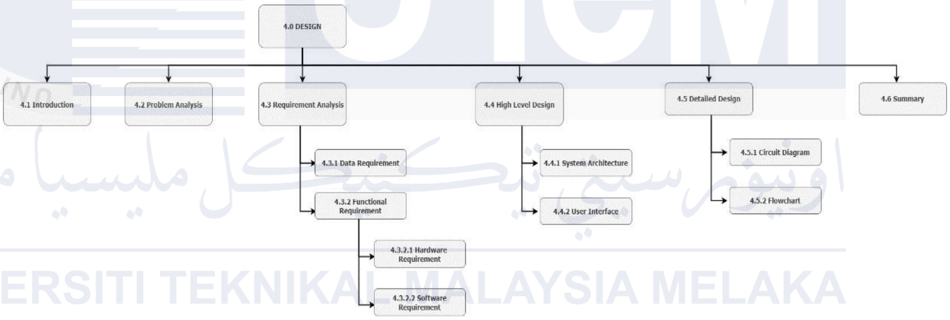


Figure 4.1 Outline of Design

4.2 Problem Analysis

Nowadays, one of the challenges faced by flood detection systems in Malaysia is inadequate sensor coverage. This results in delayed or incomplete detection of floods, making it difficult to issue timely warnings and implement effective mitigation measures. Next is effective communication and alert systems are essential for notifying the public about impending floods. However, existing systems in Malaysia may face challenges in spread it timely and accurate flood warnings to all community, especially in remote areas with limited connectivity.

By using IoT Smart-Based flood detection system, we can take more precautions about flood that happen in Malaysia. The system employs specific thresholds for determining the water level and alerting the community when flood is coming. Next, we are also adding an android application that can link with the system so it will be easier to notify them, and the application will keep the captured data in cloud for references in the future. By implementing this IoT smart-based flood detection system, we can alert the community to be prepared if a flood is coming.

4.3 Requirement Analysis

4.3.1 Data Requirement

Based on the purpose system, there are two values that are measured by humidity sensor which is humidity level and temperature while for ultrasonic sensor is measure the water level. Through the NodeMCU ESP32 microcontroller, data will be detected, transmitted, and recorded in the synology server. This system output is a web dashboard to monitor the flood by the authorities. As the future, the data can be monitored at the website for the user to see.

As for the water level, we used ultrasonic sensor to measure the distance between ultrasonic level and surface of the water. According to Carlson, C [12] Ultrasonic sensors use ultrasonic waves to measure distance. The sensor head sends out an ultrasonic wave and picks up the wave that the object reflects. To calculate the exact distance between ultrasonic and surface of the water is using this formula, distance = (duration / 2) * 29.1. Using the time recorded, determine the distance. Since the ultrasonic wave travels to the object and back, you must divide the total time by 2 to account for this. The speed of sound in air at room temperature is roughly calculated to be 29.1 cm/s. Depending on the surroundings, this value could change a little.

For humidity level and temperature, we used humidity sensor which is DHT11 to measure it. By detecting the electrical resistance between two electrodes, DHT11 determines relative humidity. The relative humidity has a direct relationship with the change in resistance between the two electrodes. Greater relative humidity causes the resistance between the electrodes to drop, whereas lower relative humidity causes the resistance to increase, RudraNarayanG [13]. According to Malaysian Meteorological Department, daily variations in relative humidity are larger than yearly variations. In dry months, the typical daily minimum might be as low as 42% and as high as 70% in wet months. The maximum daily average, which is over 94%, didn't vary much from location to location [14].

There are two thresholds that have been defined in the system, the table below shows the threshold that has been chosen which is normal level (6cm) and risk level (3cm). Dejan [15] stated that ultrasonic sensors can measure with a range of 2 cm to 400 cm. The threshold chosen for the prototype based on the container used for the prototype, but mostly river's depth in Malaysia is 10 kilometer and can take up to hundreds of kilometers.

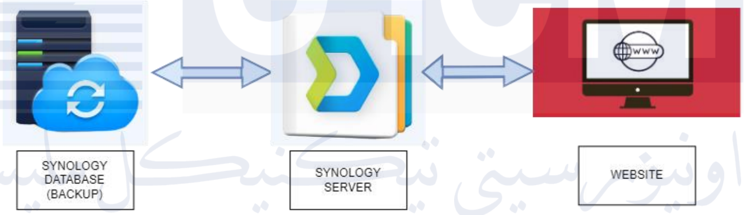


Figure 4.2 Project Data Requirements

4.3.2 Functional Requirement

4.3.2.1 Hardware Requirement

- NodeMCU ESP32

The NodeMCU is one of the low-cost IoT platforms which run with ESP32 Wi-Fi module. It is also the most robust and easy to use the boards since it is small and convenient.

Figure 4.3 NodeMCU ESP32

Table 4.1 Specification ESP32

Model	NodeMCU ESP32
Type	ESP32
Clock Frequency	240 MHz
Operating Voltage	2.7V to 3.6V
Wi-Fi	2.4 GHz

- Expansion Board for ESP32

A peripheral or accessory board created to increase the capability of the ESP32 microcontroller often serves as an expansion board for the device. The ESP32 is a flexible microcontroller that is frequently used in a variety of embedded and IoT (Internet of Things) applications. Expansion boards can be used to enhance the ESP32 platform with additional capabilities, sensors, or networking choices.

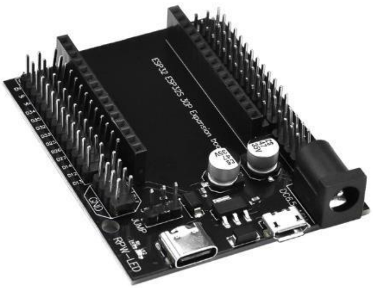


Figure 4.4 Expansion Board for ESP32

Table 4.2 Specification Expansion Board ESP32

Model	**ESP32 Expansion Base Board**
Type	**ESP32**
Operating Voltage	**7V to 12V**
Dimension	**65 x 55mm**

- Ultrasonic Sensor (HC-SR04)

The adaptability, usability, and dependability of ultrasonic sensors are well known in applications for object detection and distance measuring. They can deliver precise measurements in a variety of environmental circumstances and are appropriate for both short-range and medium-range applications.

Figure 4.5 Ultrasonic Sensor (HC-SR04)

Table 4.3 Ultrasonic Sensor (HC-SR04)

Model	HC-SR04
Type	Ultrasonic
Frequency	40 kHz
Operating Voltage	5V
Measurement Range	2cm to 400cm

- Humidity Sensor (DHT11)

A popular and affordable digital temperature and humidity sensor is the DHT11. The most popular sensor out there is the DHT11, which is typically utilized by students and makers. This sensor's primary job is to measure the ambient temperature and humidity. Additionally, factory calibrated, the sensor is simple to link with any microcontroller, including Arduino. The sensor can accurately measure humidity from 20% to 90% and temperature from 0°C to 50°C.

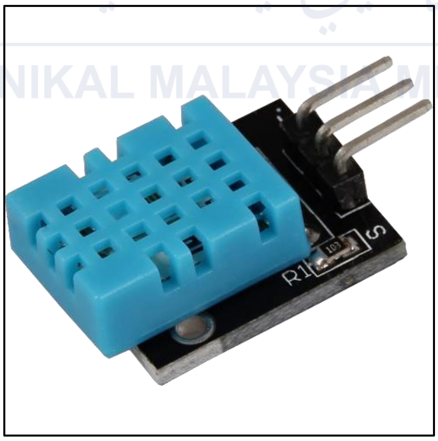


Figure 4.6 Humidity Sensor (DHT11)

Table 4.4 Specification of Humidity Sensor

Model	DHT11
Type	Humidity Sensor
Operating Voltage	3.5V to 5.5V
Humidity Range	20% - 90%
Temperature Range	0°C to 50°C

- Jumper Wire

Jumper wires are simply wiring with connector pins at each end. Jumper wires allowing to be used to connect two points to each other without soldering, and generally used with breadboards, solderless board, and devices to make it easy to modify circuits when required.

Figure 4.7 Jumper wire

- Synology NAS

A well-liked and approachable data storage solution, Synology NAS offers a robust operating system and cloud-based infrastructure to develop and manage data applications. It makes it simple for users, businesses, and developers to store and connect data to the internet, enabling remote access and management of various files and multimedia using a computer or mobile device.

Figure 4.8 Synology Server

- Router

In the flood detection system project, the router plays a crucial role in enabling seamless communication and data transmission between various components of the IoT architecture. The router acts as a central hub that connects the microcontrollers to the cloud server by providing a stable and secure Wi-Fi network. This connectivity allows the sensor data, which includes real-time measurements of water levels to be transmitted to the cloud for processing and storage.

Figure 4.9 Router

4.3.2.2 Software Requirement

- Arduino IDE

For the software requirement is Arduino IDE (Integrated Development Environment) and its open source provides free or almost low-cost, highly reliable and affordable technology. It is easy to write code and upload it to the board with the open-source Arduino Software (IDE). The software can be used with any Arduino development board. It is conveniently accessible for operating systems. These microcontrollers can be easily programmed using C or C++ language in the Arduino IDE. The model and port number must be selected to make it work during source code upload. To ensure the function of NodeMCU ESP32, this connection uses a cable adapter to connect between Arduino IDE and NodeMCU.

Figure 4.10 Arduino IDE

- Synology DSM

Synology DiskStation Manager (DSM) is the intuitive operating system that powers Synology NAS (Network Attached Storage) devices. It provides a comprehensive set of tools for managing files, backups, and media, as well as advanced features like virtualization, cloud integration, and robust security options. DSM is designed to be user-friendly, offering a web-based interface that allows users to easily configure and manage their NAS, making it ideal for both home users and businesses. Its versatility and wide range of applications make it a valuable asset for IoT projects like remote monitoring and data storage.

Figure 4.11 Synology DSM

4.4 High Level Design

4.4.1 System Architecture

System architecture is a system developed with synology nas server. This synolog will help the user to build the system server with efficient and easy ease. The user will be used the system developed with efficient and easy to conduct. System architecture is an important part of any product that is developed. It is the process of designing, developing, and implementing a system that can meet all requirements to achieve the goal. The figure below shows the system architectural of IoT Smart-Based Flood Detection System.

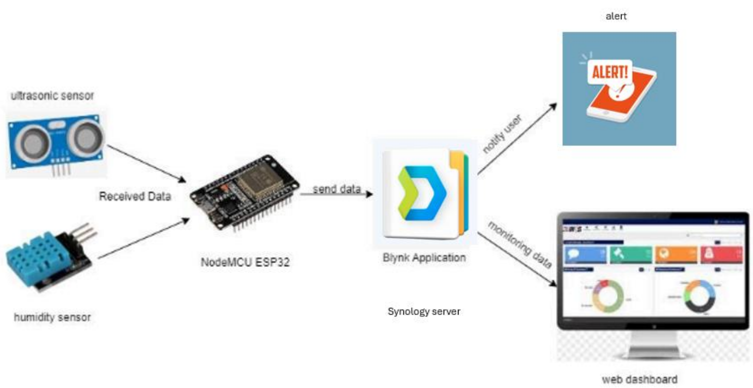


Figure 4.12 System Architecture

4.4.2 User Interface

For this user interface design there are used in the prototype, it will focus in on the website page that is used in the project. Next, the interface for website page that function for monitoring data to get alert when the water level is increasing. The main purpose of the interface is to provide information about a particular service. The interface should be easy to use and easy to understand for the user. This web dashboard is designed to help authorities in charge to monitor the surrounding environment. It displays the humidity level, temperature level and distance of the water from surface to sensor.

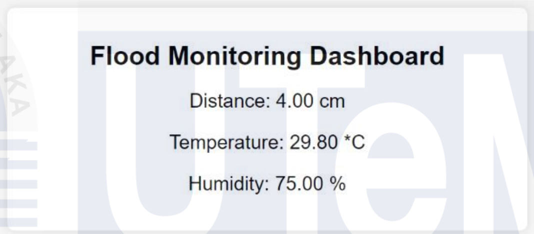


Figure 4.13 Web Dashboard

4.4.3 File Management

An efficient file management system is essential for a robust IoT-based flood detector. The system collects real-time sensor data on water levels, which needs to be stored and transmitted securely. Proper file naming conventions, directory structures, and data formats (like CSV or JSON) facilitate storage and analysis. Data compression techniques can optimize storage space, while secure protocols like HTTPS ensure safe transmission. Regular synchronization with a central server or cloud platform ensures data consistency, and error handling with backups protects against data loss. Additionally, security measures like encryption and access controls safeguard sensitive information. By implementing these strategies, file management becomes a critical 38 component in the overall reliability and effectiveness of the flood monitoring system.

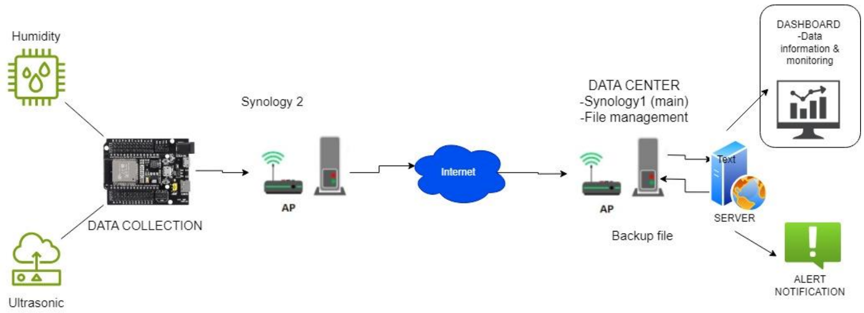


Figure 4.14 File Management

4.5 Detailed Design

This section shows the detailed design of the project where it will contain the structure view of the system to be developed which includes circuit diagram and flowchart of the system.

4.5.1 Circuit Diagram

Circuit design can keep going over things and fixing the connection mistakes. Circuit design is very important for a good project prototype.

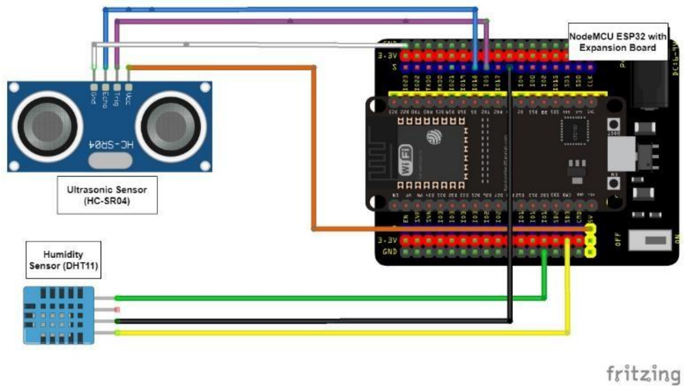


Figure 4.15 Circuit Diagram

4.5.2 Flowchart

Flowchart is a formalized graphic representation of a logic sequence of the system. In the design phase, flowchart is one of the project developments. This flowchart shows the flow of the system functionality from the device to the detection part of the system.

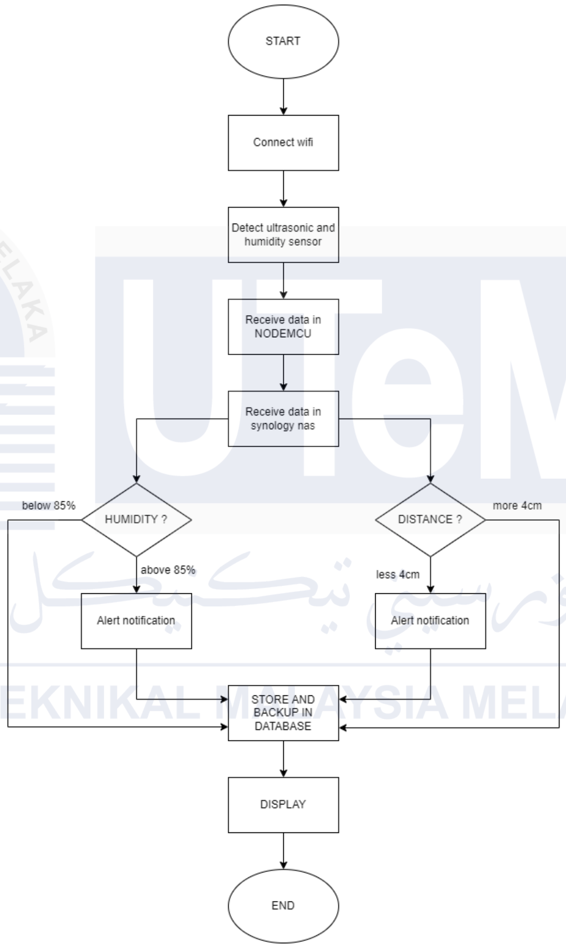


Figure 4.16 Flowchart System

4.6 Summary

Finally, the design is very important for the reference project. The prototype project will be completed through hardware, software, and system design. This design of hardware, software, and system requirements will help users ensure that the prototype is easy to install and functional, and it is important to understand the software and hardware to be used. Then, it is difficult to understand the process or steps for implementing the flood detection system to make the notification alert run smoothly and efficiently. In addition, this chapter also introduces the architecture and detailed circuit design to enable you to better understand the project process. The next chapter will focus on the implementation of a flood detection system based on this design stage.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter will go over how to use IoT to develop a Smart-Based Flood Detection System. The microcontroller in this project Is NodeMCU ESP32, and the sensor used is ultrasonic sensor and humidity sensor to detect the water level and humidity of the environment. Finally, to make this project function, it incorporates additional software and hardware into the IoT to develop a Smart-Based Flood Detection System.

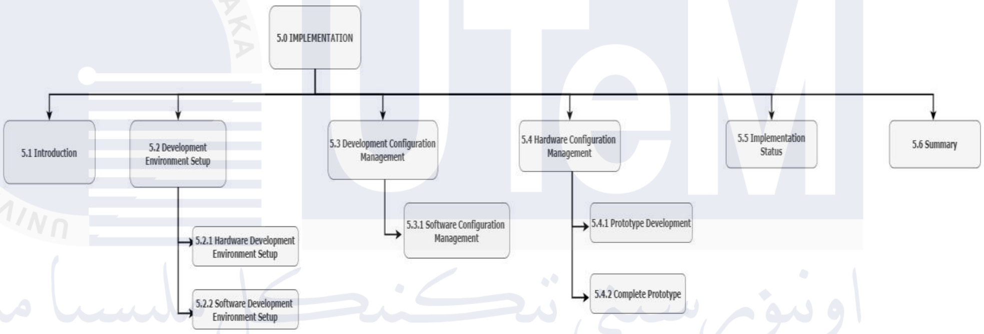


Figure 5.1 Outline for Implementation

5.2 Development Environment Setup

This section will discuss the project surrounding briefly. This project needs a prototype of the Smart-Based IoT Flood Detection System, which includes all necessary hardware and software. This chapter will provide further explanation regarding the connection of the hardware and software.

5.2.1 Hardware Development Environment Setup

The following hardware will be combined to build a functional detection system. Table 5.1 shows the details on how it connected using microcontroller Of NodeMCU ESP32.

Table 5.1 Hardware Details

Hardware	Wire	Pins
Ultrasonic Sensor HC-SR04	**VCC**	**5V**
	Trig	**D5**
	Echo	**D18**
	Gnd	**GND**
Humidity Sensor DHT11	**VCC**	**3V**
	Data	**D16**
	Gnd	**GND**

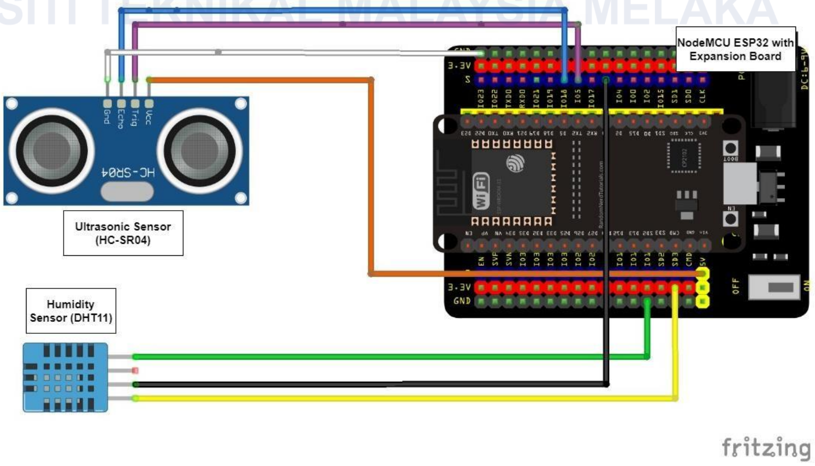


Figure 5.2 Hardware Details

5.2.2 Software Development Environment Setup

In software environment setup, the software that is needed will be installed and the installation is crucial to ensure the proper functionality of the system.

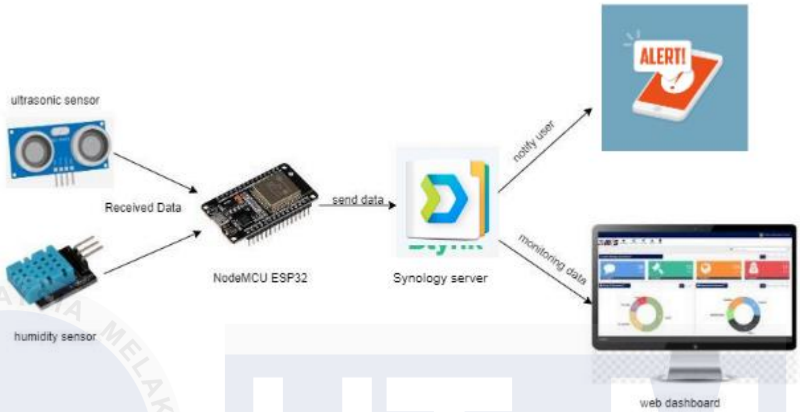


Figure 5.3 Software Details

5.3 Development Configuration Management

This section will discuss the software configuration control that was used to help the process of developing the project.

5.3.1 Software Configuration Management

5.3.1.1 Arduino IDE Configuration

- Installation board for NodeMCU ESP32

Step 1: Open Arduino IDE. Go to File and click Preferences.

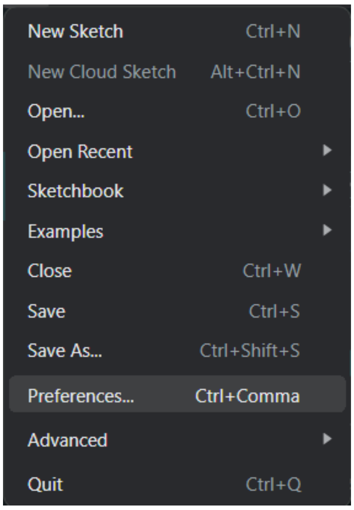


Figure 5.4 Preference in Arduino IDE

Step 2: In "Additional Board Manager URLs" field enter https://raw.githubusercontent.com/espressif/arduino-esp32/gh pages/package_esp32_index.json,http://arduino.esp8266.com/ stable/package_esp8266com_index.json" to add support for ESP32 board. Then, click 'OK'.

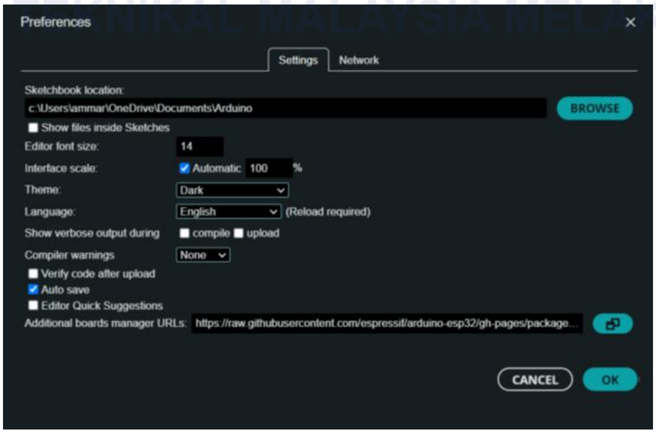


Figure 5.5 Preferences Window

Step 3: Go to Tools > Board > Board Manager, search ESP32 and install.

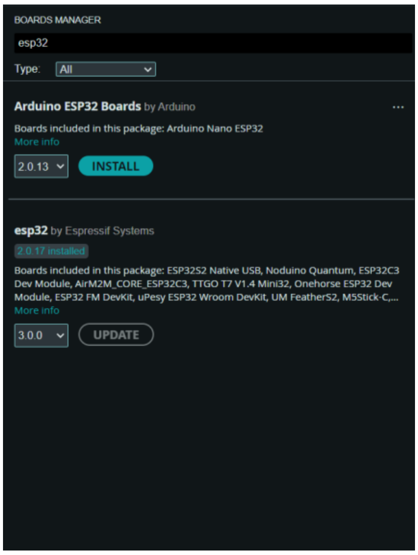


Figure 5.6 Board Manager Window

Step 4: Restart Arduino IDE. Go back to Tools > Board. ESP32 Arduino should be available. Choose Board "ESP32 DEV MODULE".

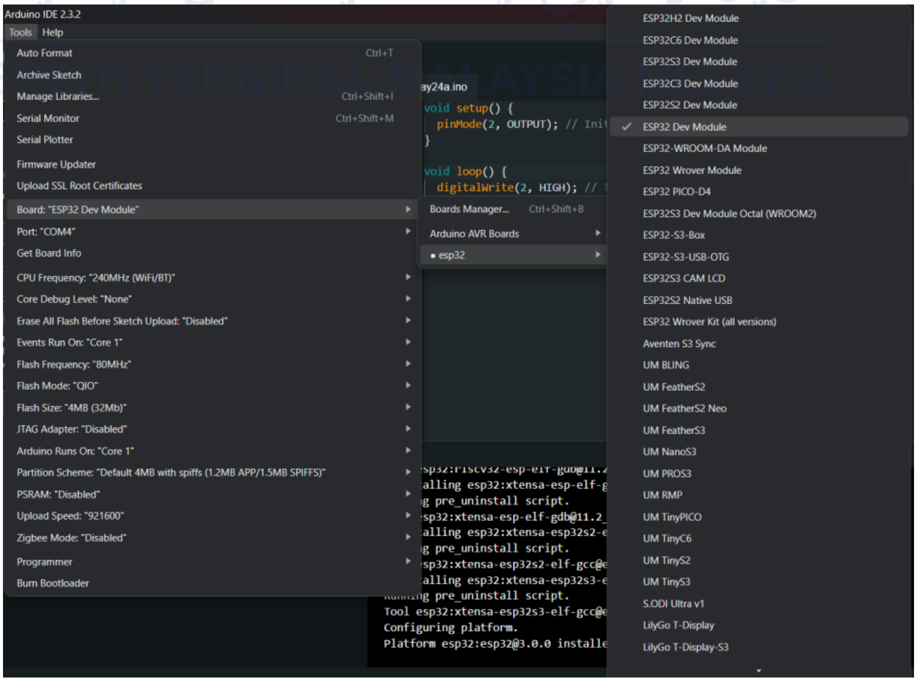


Figure 5.7 Set the board for ESP32

- Installation library for Humidity Sensor (DHT11)

Step 1: Go to sketch > Include Library > Manage Library. Search DHT Sensor library by Adafruit and install.

Figure 5.8 Library Manager Window

- Installation for Wifi (NewPing)

Step 1: Go to sketch > Include Library > Manage Library. Search NewPing library by Tim Eckel and install.

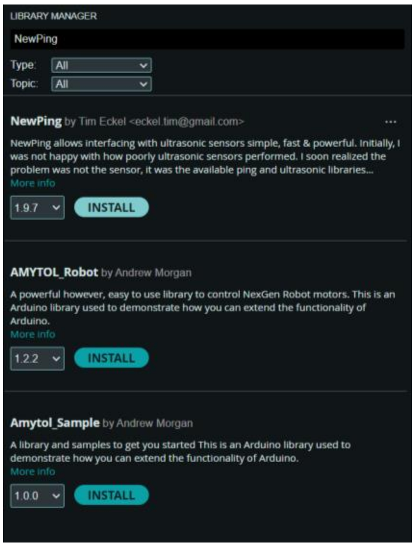


Figure 5.9 Library Install Manager Window

5.3.1.2 Web Dashboard Configuration

- Setup the web dashboard

Step 1: Log in to your Synology DSM. Go to Package Center. Search for Web Station and install it.

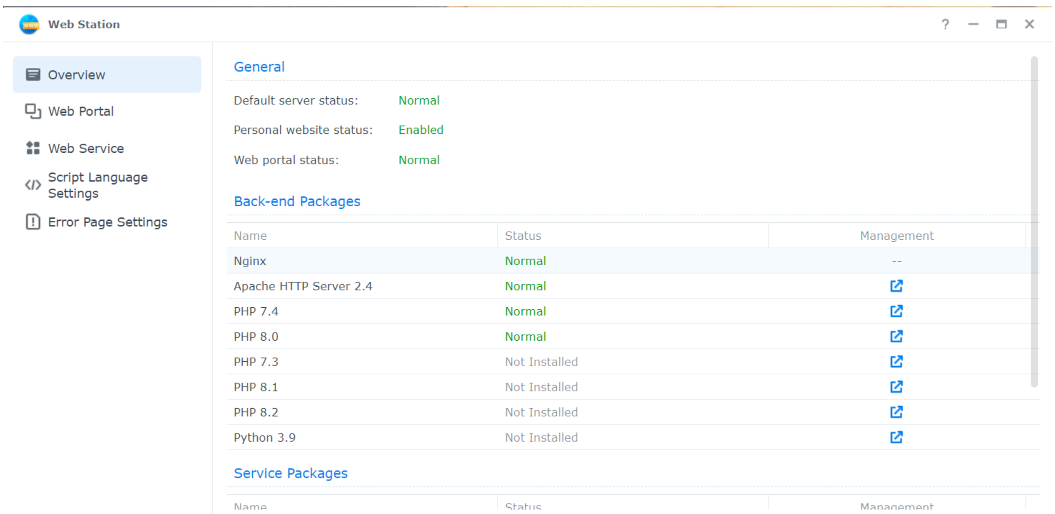


Figure 5.10 Web Station

Step 2: Use File Station to navigate to the web folder. Create web file and upload PHP files into this folder.

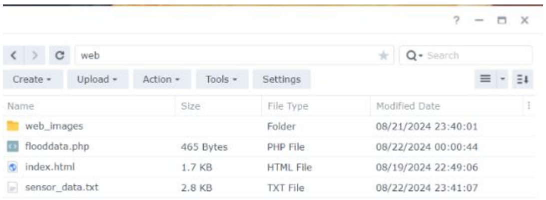


Figure 5.11 Web files

Step 3: Edit the PHP and index.html file for web interface.

```php
<?php
if (isset($_GET['distance']) && isset($_GET['temperature']) && isset($_GET['humidity'])) {
    $distance = $_GET['distance'];
    $temperature = $_GET['temperature'];
    $humidity = $_GET['humidity'];

    // Validate the data to ensure it is numeric and not empty
    if (is_numeric($distance) && is_numeric($temperature) && is_numeric($humidity) &&
        !empty($distance) && !empty($temperature) && !empty($humidity)) {

        // Get the current timestamp
        $timestamp = date("Y-m-d H:i:s");

        // Prepare the data string with timestamp
        $data = "Timestamp: " . $timestamp . " | Distance: " . $distance . " cm | Temperature: " . $temperature . " *C | Humidity: " . $humidity . " %\n";

        // Save the data to a file
        file_put_contents('sensor_data.txt', $data, FILE_APPEND);

        echo "Data saved!";
    } else {
        echo "Invalid data! Non-numeric or empty values detected.";
    }
} else {
    echo "Invalid data! Missing parameters.";
}
?>
```

Figure 5.12 PHP File

```html
<body>
    <h1>Flood Monitoring Dashboard</h1>
    <div class="dashboard">
        <div class="widget" id="timestamp-widget">
            <h2>Timestamp</h2>
            <p id="timestamp">--</p>
        </div>
        <div class="widget" id="distance-widget">
            <h2>Distance</h2>
            <img src="alert-icon.png" class="alert-icon" id="distance-alert">
            <p id="distance">-- cm</p>
        </div>
        <div class="widget" id="temperature-widget">
            <h2>Temperature</h2>
            <img src="alert-icon.png" class="alert-icon" id="temperature-alert">
            <p id="temperature">-- °C</p>
        </div>
        <div class="widget" id="humidity-widget">
            <h2>Humidity</h2>
            <img src="alert-icon.png" class="alert-icon" id="humidity-alert">
            <p id="humidity">-- %</p>
        </div>
    </div>

    <div id="chart-container">
        <canvas id="dataChart"></canvas>
    </div>

    <script>
        let distanceData = [];
        let temperatureData = [];
        let humidityData = [];
        let labels = [];

        function fetchSensorData() {
            fetch('sensor_data.txt')
                .then(response => response.text())
                .then(data => {
                    let lines = data.trim().split("\n");
                    let lastLine = lines[lines.length - 1];
```

Figure 5.13 Index File

5.3.1.3 Establish the code.

- Declare Library

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <NewPing.h>
#include <DHT.h>

#define TRIG_PIN 12
#define ECHO_PIN 13
#define MAX_DISTANCE 400
#define DHTPIN 16
#define DHTTYPE DHT11
```

Figure 5.14 Declare Library

- Declare Wi-Fi SSID and password

```
const char* ssid = "rumahsewabahagia-2.4GHz@unifi";
const char* password = "makandulu69";
```

Figure 5.15 Wi-Fi SSID and Password

- Declare Synology token

```
// Send data to server
if (WiFi.status() == WL_CONNECTED) {
  HTTPClient http;
  String serverPath = "http://192.168.0.102/flooddata.php?distance=" + String(distance) + "&temperature=" + String(temperature) + "&humidity=" + String

  http.begin(serverPath.c_str());
  int httpResponseCode = http.GET();

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

  http.end();
}
```

Figure 5.16 Synology Token

- Code to declare DHT11 and Ultrasonic pin.

```
#define TRIG_PIN 12
#define ECHO_PIN 13
#define MAX_DISTANCE 400
#define DHTPIN 16
#define DHTTYPE DHT11
```

Figure 5.17 Declare Pin

- NodeMCU Setup Coding.

```
void setup() {
  Serial.begin(115200);
  delay(1000);

  // Initialize DHT sensor
  dht.begin();

  // Connect to WiFi
  Serial.println("Connecting to WiFi...");
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }

  // Print ESP32 IP address
  Serial.print("Connected to WiFi. IP address: ");
  Serial.println(WiFi.localIP());
}
```

Figure 5.18 Setup Coding

- Code to display the reading.

```
void loop() {
  // Read distance from HC-SR04
  unsigned int uS = sonar.ping();
  float distance = uS / US_ROUNDTRIP_CM;

  // Read temperature and humidity from DHT11
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  // Print the results to Serial Monitor
  Serial.println("------- Sensor Readings -------");
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");
  Serial.println("-----------------------------");
```

Figure 5.19 Code to display the reading.Synology Application Configuration

5.3.1.4 Setup Web Dashboard Interface

Step 1: Login to the Synology and edit the index.html file for the web interface in the web file.

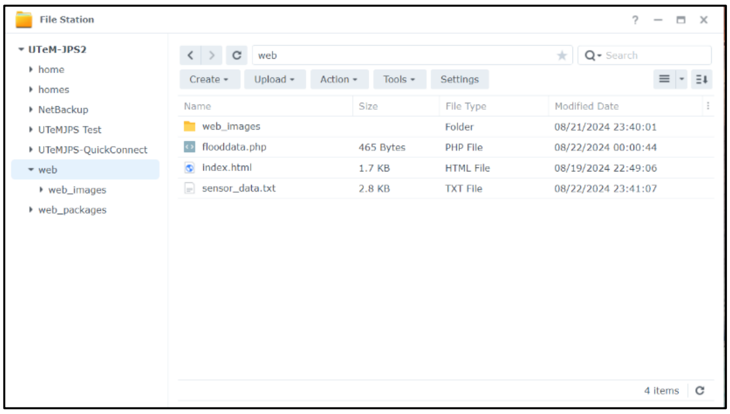


Figure 5.20 Web File

```
<body>
    <h1>Flood Monitoring Dashboard</h1>
    <div class="dashboard">
        <div class="widget" id="timestamp-widget">
            <h2>Timestamp</h2>
            <p id="timestamp">--</p>
        </div>
        <div class="widget" id="distance-widget">
            <h2>Distance</h2>
            <img src="alert-icon.png" class="alert-icon" id="distance-alert">
            <p id="distance">-- cm</p>
        </div>
        <div class="widget" id="temperature-widget">
            <h2>Temperature</h2>
            <img src="alert-icon.png" class="alert-icon" id="temperature-alert">
            <p id="temperature">-- °C</p>
        </div>
        <div class="widget" id="humidity-widget">
            <h2>Humidity</h2>
            <img src="alert-icon.png" class="alert-icon" id="humidity-alert">
            <p id="humidity">-- %</p>
        </div>
    </div>

    <div id="chart-container">
        <canvas id="dataChart"></canvas>
    </div>

    <script>
        let distanceData = [];
        let temperatureData = [];
        let humidityData = [];
        let labels = [];

        function fetchSensorData() {
            fetch('sensor_data.txt')
                .then(response => response.text())
                .then(data => {
                    let lines = data.trim().split("\n");
                    let lastLine = lines[lines.length - 1];
```

Figure 5.21 Index file

Step 2: Interface for web interface

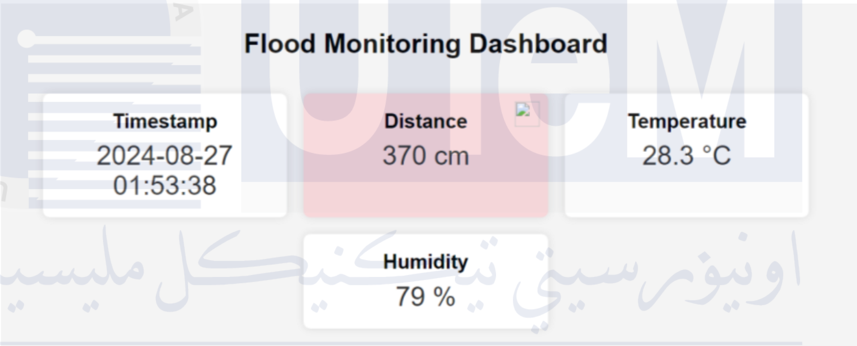


Figure 5.22 Widget Box

5.3.1.5 Setup for ShareSync

Step 1: Install Synology Drive Server on both Synology 1 and 2. On Synology 1, open Synology drive admin console. Go to Team Folder > Enable the folder want to sync the data from synology 2.

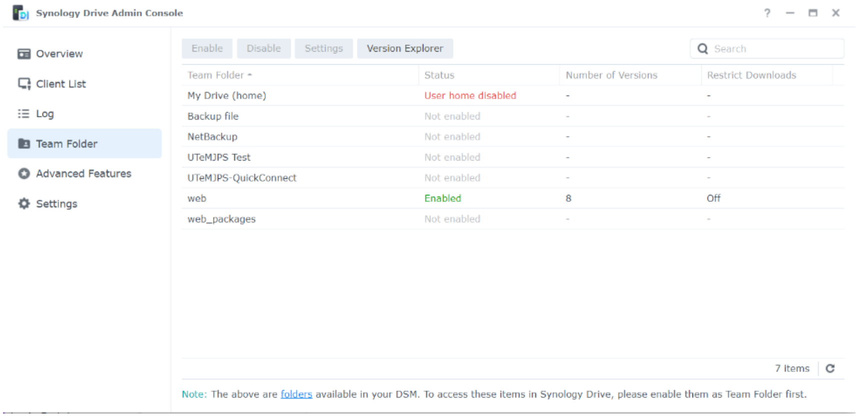


Figure 5.23 File sync

Step 3: On synology 2 and open sharing drive share sync. Click create to start a new sync task. Enter the Synology 1 ip address, username and password.

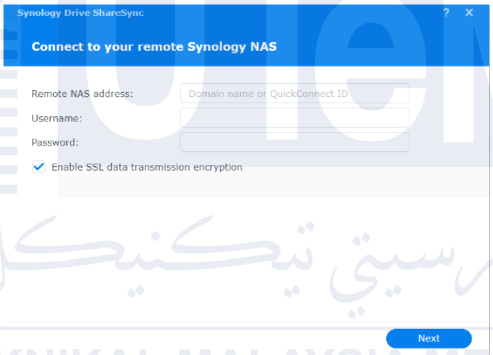


Figure 5.24 New Task Sync

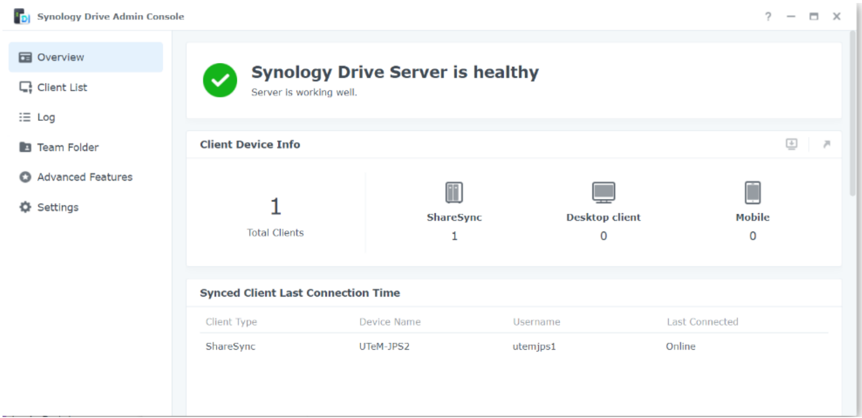


Figure 5.25 Synology Drive Status

5.3.1.6 Setup for Hyper Backup

Step 1: Install and open the Hyper backup application. Click +(add) button and select the Local shared Folder.

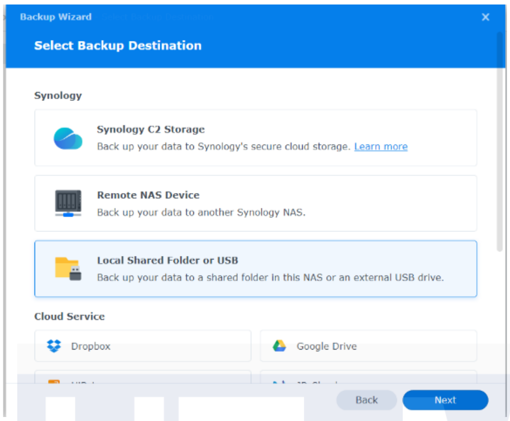


Figure 5.26 Hyper Backup Setup

Step 2: Select backup Version Type.

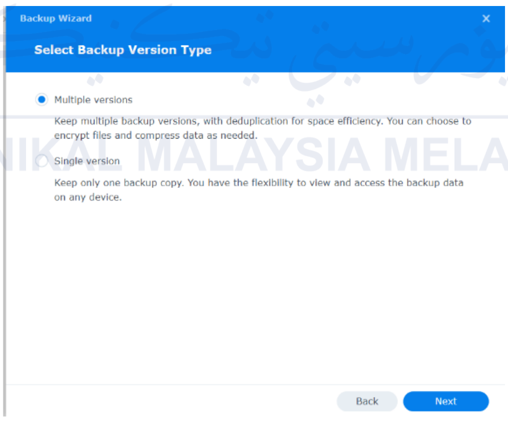


Figure 5.27 Backup Type

Step 3: Select backup destination.

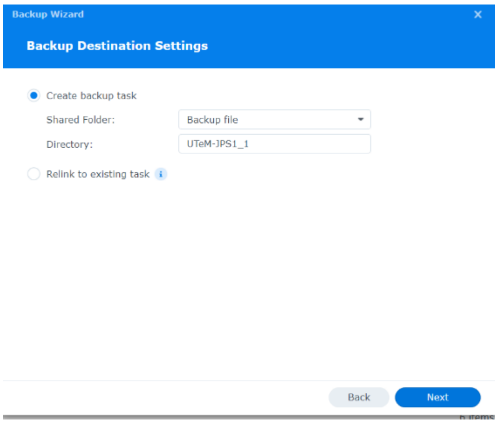


Figure 5.28 Backup Destination

Step 4: Select the data that needed to backup.

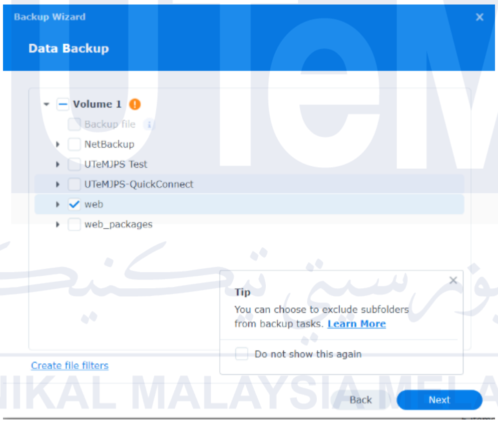


Figure 5.29 Backup Data

Step 5: Select the Backup Setting

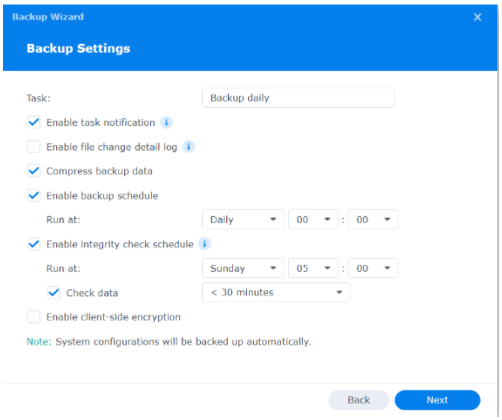


Figure 5.30 Backup Setting

Step 6: Final checkup setup

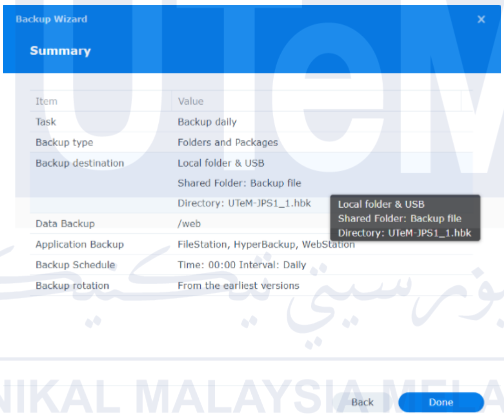


Figure 5.31 Complete Setup

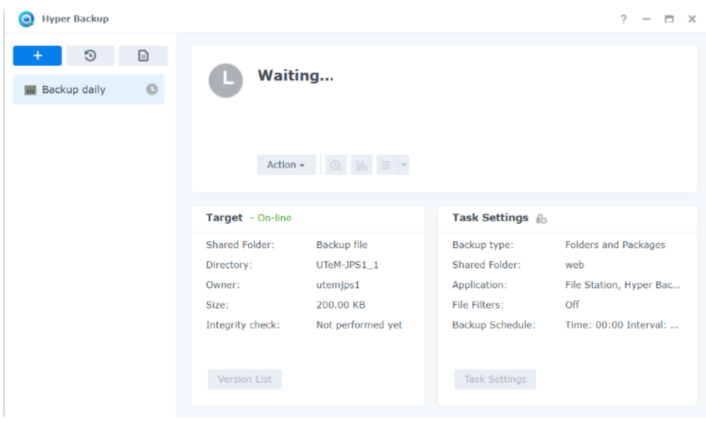


Figure 5.32 Done Setup Hyper Backup

5.4 Hardware Configuration Management

5.4.1 Prototype Development

This section will be described in detail in the setup device that used NodeMCU ESP32 and two others sensor which are ultrasonic and humidity. The hardware is outlined in chapter 4, and the connecting process is explained in further detail below.

5.4.2 Complete Prototype

The complete prototype contains a container that has 15 cm depth and two sensors located at the top of the container to easily detect the water distance and humidity level. The sensors connected to the ESP 32 through the jumper wire, and it is attached to the container.

Figure 5.33 Complete Prototype

5.5 Implementation Status

In this section, the process of creating each component will be discussed in detail, including the amount of time required for completion.

Table 5.2 Implementation Status

NO	COMPONENT	DESCRIPTION	DURATION
1.	Assembling Hardware	Obtaining the project’s required hardware and components	1 week
2.	Building Project Prototype	Using ultrasonic and humidity sensor with NodeMCU ESP32	1 week
3.	Software Configuration	Download and installation process for Arduino IDE, Synology NAS configuration and HyperBackup for file management	2 weeks
4.	Source Code Implementation	Upload and compile the source code to run the prototype	4 weeks

5.6 Summary

To recap, this chapter discusses the project’s implementation of the software and hardware for development environment for the project. The details of software or programming language is utilized to develop this project are given in software configuration management.

CHAPTER 6: TESTING

6.1 Introduction

For this chapter, it discusses the testing phase of the project, to ensure the product and system developed meets the requirement and functions properly. All components and products are tested to get the output and to ensure the objective of the project is successful and can be used for this project. In this project, a microcontroller is used which is NodeMCU ESP32, that is focus on the Flood Detection System that can help to monitor and detect the increasing of water level based on humidity changes. Aside from that, this chapter contains subtopics such as test plan, test setting, test design, test outcome, and interpretation.

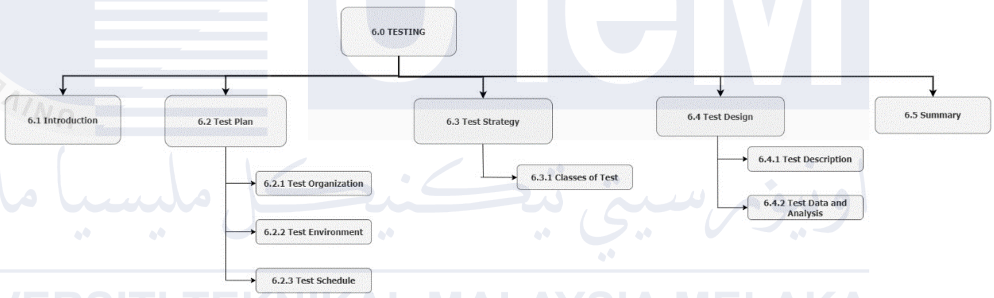


Figure 6.1 Outline for Testing

6.2 Test Plan

This section goes into the fundamentals of each method and product, as well as product testing. This segment also discusses who is involved in the research process and how the activities are carried out. This test plan assists in the detection of issues that may arise during device operation. The projects' goals are to ensure that the whole product and system plan can be used to solve the user's problem. The test plan is very important before the user uses the products and systems in the project.

6.2.1 Test Organization

The responsibility for conducting this test primarily falls upon the developer or the project team, as they possess an intricate understanding of the system's inner workings. It is crucial for a developer to spearhead this test group since their comprehensive knowledge ensures a thorough examination of the project's functionality. This phase involves meticulously assessing the system's performance, identifying potential errors, and improving the system.

6.2.2 Test Environment

Testing can be performed in an indoor environment, but that place must have a Wi-Fi network to retrieve the data and transfer it to a web interface because our system requires the internet. The NodeMCU ESP32 is the main controller of this prototype to capture the sensor's data and display the data to Synology. The humidity sensor is used to detect the moisture level and works as an early indicator for flood and ultrasonic sensor used to detect the increasing of water level. The data of the sensors will be sent to the ESP32 and link it to the Synology. As a result, the test environments will be developed during the project development process to ensure the testing phase is reliable and accurate.

6.2.3 Test Schedule

This testing section is to determine how long it takes to complete the project, create a test schedule. Several errors and problems occurred during this period, requiring additional time to resolve and verification during the implementation phase. This is a continuous testing phase that continues until the system's functionality is complete. The evaluation plan's duration is as follows. After the prototype has been set up, the user must upload the code to the hardware and test it to see whether it works or not. As a result, the prototype has worked properly and successfully.

I. Start the prototype.

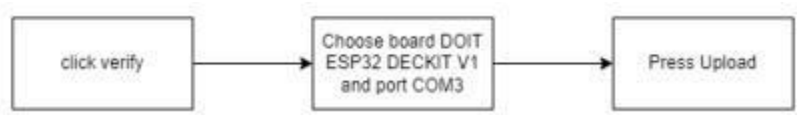


Figure 6.2 Shows how to switch on the prototype.

Users can use the Arduino IDE's verify button to ensure that the code is correct and free of errors. To ensure that the prototype will work, the users should first determine which board they want to use. The port is COM3 for the NodeMCU ESP32. The prototype will not be able to run if the users are unable to select this board and port. After that, to import the code into the prototype, press upload.

II. Functionality of the prototype

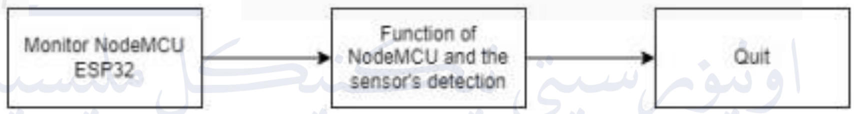


Figure 6.3 Shows the functionality of the prototype.

The key components are those that control the NodeMCU ESP32. The monitoring NodeMCU ESP32 will be used to connect all the components together to provide more functionality. Next, the sensor's job is to identify the humidity level and water level. The functionality components will assist the user in alerting them if the water level is at risk level.

III. Connect with Synology.

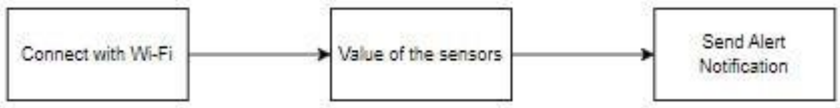


Figure 6.4 Connection with a Synology

The user must be connected to Wi-Fi in order to connect to the synology, which provides real-time data and sends a notification to the wweb dashboard. The data, which includes the sensor's value and stored in the file.

6.3 Test Strategy

In this section, we'll talk about how to choose how to test the system. White box testing is a way to check how well a system works. White box testing is a testing technique where testers examine the internal code, prototype, and software application to ensure it functions correctly, adheres to design specifications, and detects coding errors and vulnerabilities. Figure 6.4 shows a picture of how white box testing works.

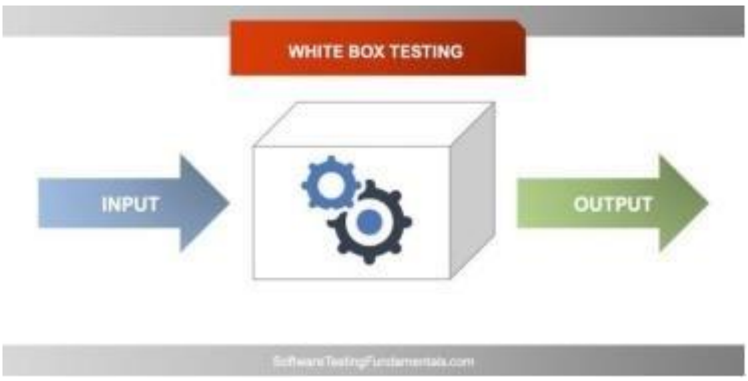


Figure 6.5 White box testing

White box testing includes path checking, output validation/functionality test, security test, loop test and data flow test. This project will focus on functional testing to make sure everything works as it should. The developer will use this method to figure out which requirements are important for the project. The test will start with the two sensors which are humidity sensor and ultrasonic sensor. The system will use the web dashboard on a web interface to monitor the sensor's value.

6.3.1 Classes of test

6.3.1.1 Functionality Test

The functionality test is done to make sure that all components worked. Testing can start when NodeMCU ESP32 is powered on. Then, the NodeMCU ESP32 can capture all the sensor's value. Next, the NodeMCU is able to receive sensor's data from humidity sensor and the ultrasonic sensor will act based on the threshold that was set on the NodeMCU. Finally, the sensor's data will be sent to Synology for monitoring process.

6.3.1.2 Application Notification Test

The notification received to the user's smartphone when the sensor's value exceeds the predetermined threshold. The smartphone and the NodeMCU must first be connected to the internet to receive the notification. As a result, users are able to know the increasing of water level.

6.4 Test Design

This is one of the most important parts of the testing process, where designed components or modules are checked for accuracy and effectiveness in the system. To validate the process, each part or module is checked. Testing is critical to ensure that the device meets the specifications and functions properly. By doing this analysis, the project's functionality can be improved, allowing it to meet its objectives. Furthermore, testing may aid in the improvement of system functionality in order to achieve system goals. To ensure flawless performance, all components and modules have been thoroughly checked.

6.4.1 Test Description

The aim of this section is to determine which parts of the project need testing. The test cases in this project are detailed in tables 6.1 to 6.5. This section covered all of the components and modules that were used to produce a precise and successful result. This prototype needs all of the knowledge about the components used in the project. The users must be aware of the project's procedures as well as the expected performance. In the table below, all of the test cases are mentioned.

6.4.1.1 Connectivity test between NodeMCU ESP32 and computer

Table 6.1 Connectivity of NodeMCU ESP32 and computer

Test Case ID	TC_01
Test Functionality	Connectivity test of NodeMCU ESP32 and computer
Precondition	NodeMCU and computer are connected using USB adapter. Arduino IDE installed on the computer
Execution Steps	Launch Arduino IDE and connect with Arduino by using USB cable. Upload the C++ script onto the Arduino board.
Expected Result	Arduino IDE can detect the connection of the NodeMCU board
Error Message	None
Result	Pass

6.4.1.2 Connectivity test between NodeMCU ESP32 and sensor

Table 6.2 Connectivity of NodeMCU ESP32 and Sensor

Test Case ID	TC_02
Test Functionality	Connectivity test of NodeMCU ESP32 and sensor (Humidity and Ultrasonic)
Precondition	NodeMCU and computer are connected using USB adapter. Arduino IDE installed on the computer. NodeMCU and sensor are connected using jumper wire
Execution Steps	Launch Arduino IDE and connect with Arduino by using USB cable. Connect the NodeMCU and sensor using jumper wire. Upload the C++ script onto the NodeMCU board. Open serials monitor to observe any input value from the sensor
Expected Result	-The C++ script is successfully uploaded onto NodeMCU board.

	The NodeMCU are able to receive all the sensor values. The Arduino IDE able to show the output value from the humidity and ultrasonic sensor using serial monitor.
Error Message	None
Result	Pass

6.4.1.3 Connectivity test between NodeMCU and Synology NAS

Table 6.3 Connectivity of NodeMCU ESP32 and Synology NAS

Test Case ID	TC_03
Test Functionality	Connectivity test of NodeMCU and Synology NAS.
Precondition	NodeMCU are connected to computer using USB cable. All the sensors are connected to NodeMCU using jumper wire. NodeMCU can receive sensor data. Synology NAS is opened in the browser using an ip address to setup the server.
Execution Steps	1. Launch Arduino IDE and connect with NodeMCU using USB cable.

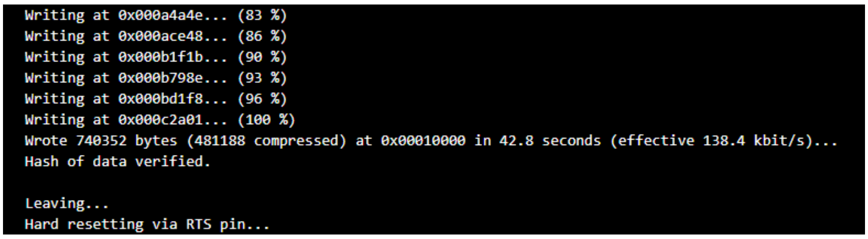
	2. Upload the C++ script onto the NodeMCU board.
Expected Result	NodeMCU are able to connect to Wi-Fi and display the connection using Arduino IDE's serial monitor. Synology server should be able to display the sensor data.
Error Message	None
Result	Pass

6.4.2 Test Data and Analysis

6.4.2.1 NodeMCU and computer Connectivity Test

The connection between the NodeMCU and the PC using USB cable. This connection is critical for the project to go forward to the next stage, uploading code.

Figure 6.6 Connectivity



```
Writing at 0x000a4a4e... (83 %)
Writing at 0x000ace48... (86 %)
Writing at 0x000b1f1b... (90 %)
Writing at 0x000b798e... (93 %)
Writing at 0x000bd1f8... (96 %)
Writing at 0x000c2a01... (100 %)
Wrote 740352 bytes (481188 compressed) at 0x00010000 in 42.8 seconds (effective 138.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Figure 6.7 Arduino IDE Monitor shows NodeMCU and Computer Connectivity

Based on figure 6.7 above, when the coding is uploaded at the NodeMCU ide board, the connectivity between the NodeMCU ESP32 and Arduino IDE is established, and it shows at the output console.

6.4.2.2 Sensor and NodeMCU Connectivity Test

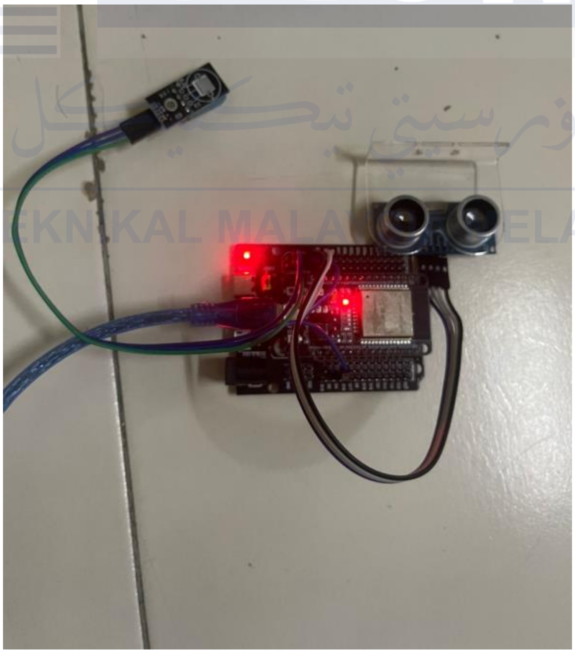
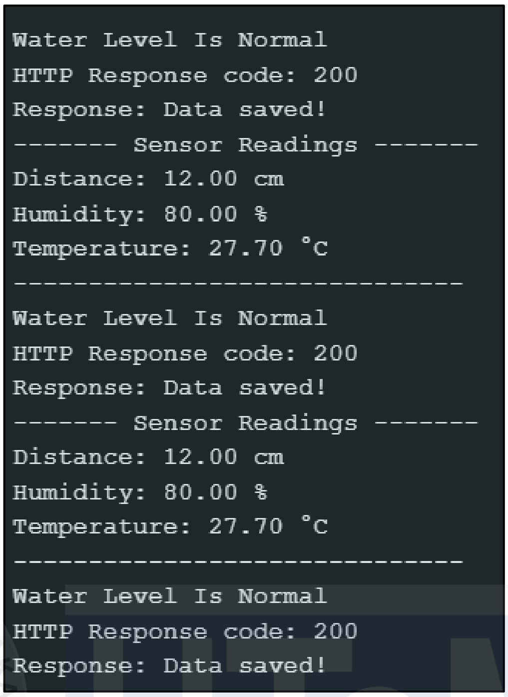


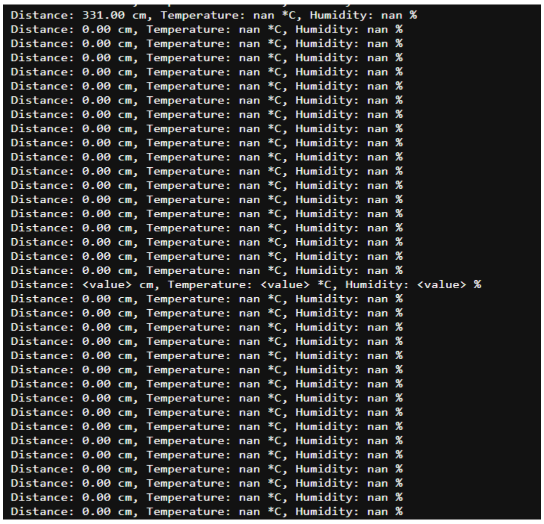
Figure 6.8 Connectivity



```
Water Level Is Normal
HTTP Response code: 200
Response: Data saved!
------- Sensor Readings -------
Distance: 12.00 cm
Humidity: 80.00 %
Temperature: 27.70 °C
------------------------------
Water Level Is Normal
HTTP Response code: 200
Response: Data saved!
------- Sensor Readings -------
Distance: 12.00 cm
Humidity: 80.00 %
Temperature: 27.70 °C
------------------------------
Water Level Is Normal
HTTP Response code: 200
Response: Data saved!
```

Figure 6.9 Serial Monitor shows the sensor connectivity.

Based on figure 6.9 above, the serial console displays the reading for the sensor. Its shows that the connectivity between sensor, NodeMCU ESP32 and Arduino IDE is established.



```
Distance: 331.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: <value> cm, Temperature: <value> *C, Humidity: <value> %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
Distance: 0.00 cm, Temperature: nan *C, Humidity: nan %
```

Figure 6.10 Sensor can connect with Web Dashboard

Based on figure 6.10 above, the web dashboard shows online events that shows the connectivity between Synology and NodeMCU ESP32 is established and the widget at web dashboard display the reading of the sensor, which means the connection between sensor and web dashboard is established.

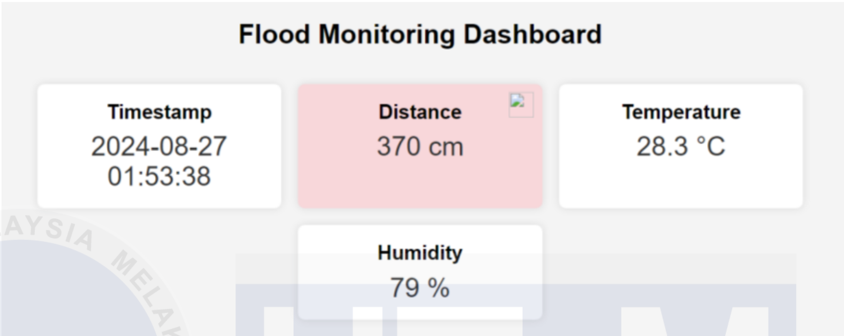


Figure 6.11 Sensor can connect with synology web dashboard

Based on figure 6.11 above, that shows the connectivity between Synology Server and NodeMCU ESP32 is established and the web monitoring display the reading of the sensor.

6.4.2.3 Synology Connectivity Test

```
HTTP Response code: 200
Response: Data saved!
```

Figure 6.12 Arduino IDE shows the synology connectivity.

Based on figure 6.12 above, it shows the connectivity between Blynk and the Wi-Fi. The Arduino IDE serial monitor displays the NodeMCU ESP32 and synology is connected to the Wi-Fi and its IP address.

6.4.2.4 Phone Notification Test

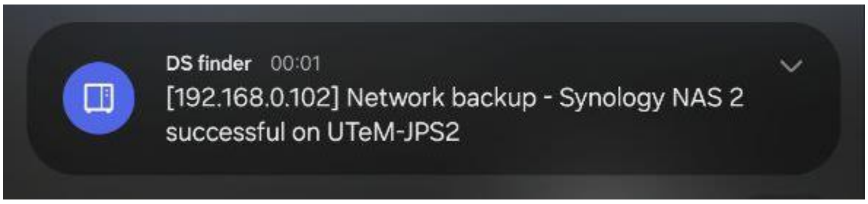


Figure 6.13 Synology Notification

Based on figure 6.12 above, it shows the notification that was sent to user. The notification is send to the user within 1s after the sensor display the reading of the water distance.

6.4.2.5 Synology Data Record Test


```
Timestamp: 2024-08-28 03:27:41 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:27:43 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:27:45 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:27:47 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:27:49 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:27:51 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:27:53 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:27:55 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:27:58 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:00 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:02 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:04 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:06 | Distance: 13.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:08 | Distance: 8.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:10 | Distance: 8.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:12 | Distance: 8.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:14 | Distance: 8.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:16 | Distance: 8.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:18 | Distance: 8.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:20 | Distance: 8.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:22 | Distance: 8.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:24 | Distance: 8.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
Timestamp: 2024-08-28 03:28:27 | Distance: 8.00 cm | Temperature: 27.00 *C | Humidity: 80.00 %
```

Figure 6.14 Synology Data Record Test

Based on figure 6.14 above, it shows the data record of each reading of the sensor within 1 minute. It can be downloaded at the web dashboard and displayed in TXT format. From the data above, we can see the summarization of the environment condition and its help for future use.

6.5 Conclusion

In summary, all the goals were met, including designing the IoT Flood Detection System with a clear and productive design, developing the tool prototype, and testing the prototype's effectiveness. The feature test is used to ensure that the prototype can work properly. The results were analyzed, and conclusions were reached based on different settings. In the next chapter, will discuss project summarization, contribution, limitation, and the entire project progress will be discussed in the next chapter.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

In this chapter project summarization, project contribution and project limitation will be covered. This chapter is critical to the project's efficiency and improvement.

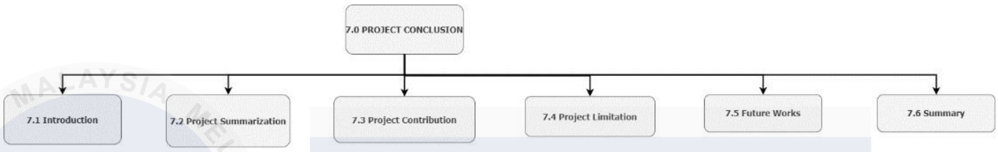


Figure 7.1 Outline of Project Conclusion

7.2 Project Summarization

IoT Smart-Based Flood Detection System is a project that aims to ensure that the user can detect flood situations efficiently. There is web dashboard that has been set up to make it easier for the authorities to keep on monitoring the environment condition that might cause flood to occur. For the user, the Blynk Application helps to notify the user when the water distance reaches a certain threshold that been set. This project uses NodeMCU ESP 32 as microcontroller and there are two sensors that involve which is Ultrasonic Sensor HC-SR04 to measure the water distance of the water and humidity sensor DHT11 to measure the humidity level. The microcontroller connected through Wi-Fi and the readings from sensor will display through Synology web interface and it store inside the file station.

The three objectives in this project are to study the problem that occur, and current solution used in the existing system, to develop the IoT Smart- Based Flood Detection System that could help in detection and monitoring flood and to evaluate the effectiveness of the proposed system based on the parameter that been involved. Every part serves a purpose in ensuring that the prototype runs smoothly. This system works well after integrating the hardware and software.

Table 7.1 Strength and Weakness Prototype

Strength	Weakness
Visualization of data. Alert Notification. Record Data. Increase user awareness. Low-cost prototype.	Power Source. Accuracy of sensor.

7.3 Project Contribution

I. The problems that been occur and method that been used in the existing system have been encountered.
II. The software and hardware used is help in detection and monitoring the flood.
III. The data that been obtain based on certain threshold is effective.

7.4 Project Limitation

I. Location

- This project needs to add the location of the user, so the water distance and alert notification pop out based on the exact place.

II. Internet Speed

- This project needs a stable internet connection and speed to ensure real-time data transmission.

7.5 Future Works

There are many more future works that can be done to increase the system functionality and efficiency. To improve the project, several suggestions have been put forward:

- IoT Smart-Based Flood Detection System can be improved by unlocking the extra features of Synology such as location.
- The rain sensor can accurately detect if any rain drops on board so it can detect more effectively compared to humidity sensor that used to predict weather conditions.

7.6 Summary

This project develops a simple IoT Smart-Based Flood Detection System to measure the increasing of water distance and humidity level. This project is powered by a NodeMCU microcontroller attached with two sensors which are ultrasonic sensor HC-SR04 and humidity sensor DHT11. Next, it provides a notification to the user if the water distance reaches a certain threshold. This project met all objectives stated in chapter 1 and the implementation and testing part is clearly shown in the previous chapter.

REFERENCES

[1] Davies, R. (2023) Malaysia – Floods in Johor and Pahang After 430mm of Rain in 24 Hours, Asia,News. Available at: https://floodlist.com/asia/malaysiafloods-johor-pahang-sabah-january-2023 (Accessed: 19 Meir 2023).

[2] Haskina Farhana Awang Sufa, Mohd Ismail Yusof and Mohd Aliff Afira Sani (2019) 'Flood Monitoring and Warning System With Iot', 3(2), pp. 7–13. Available at: www.mitec.unikl.edu.my/mjit.

[3] "An IoT Based Flood Monitoring and Response System" by Siew-Ping Yiiong, Nancy Bundan and Kim-Mey Chew

[4] Baskhara, H.A., Anwar, S. and Abdurahman (2021) 'Design of Water Level Monitoring using Ultrasonic Sensor', Journal of Physics: Conference Series, 1772(1). Available at: https://doi.org/10.1088/1742-6596/1772/1/012001.

[5] Prathaban, B.P., R, S.K. and M, J. (2023) 'IoT Based Early Flood Detection and Avoidance System', (2), pp. 555–563. Available at: https://doi.org/10.1007/978-3-031-35501-1_55.

[6] Chitra, M. et al. (2020) 'IoT based Water Flood Detection and Early Warning System', International Journal of Scientific Research in Research Paper. Computer Science and Engineering, 8(5), pp. 47–53. Available at: www.isroset.org.

[7] Binti Zahir, S. et al. (2019) 'Smart IoT Flood Monitoring System', Journal of Physics: Conference Series, 1339(1). Available at: https://doi.org/10.1088/1742-6596/1339/1/012043.

[8] Ahmad, M.L. and Suhaimi, S. (2022) 'Flood Detection and Prevention System', International Journal of Recent Technology and Applied Science, 4(2), pp. 97–111. Available at: https://doi.org/10.36079/lamintang.ijortas0402.398.

[9] Hamid, N.A. et al. (2022) 'Flood Alert and Monitoring System using the Internet of Thing Technology'.

[10] Fergus, P. and Al, A. (2020) 'E f m s i t a', (May), pp. 3348–3353.

[11] Haslina Farhana Awang Sufa, Mohd Ismail Yusof and Mohd Aliff Afira Sani (2019) 'Flood Monitoring and Warning System With Iot', 3(2), pp. 7–13. Available at: www.mitec.unikl.edu.my/mjit.

[12] Carlson, C. (2023) How Ultrasonic Sensors Work, 1 march. Available at: https://maxbotix.com/blogs/blog/how-ultrasonic-sensors-work#:~ :text=For presence detection%2C ultrasonic sensors, sensors are a reliable choice (Accessed: 13 Jun 2024).

[13] RudraNarayanG (no date) Measuring Humidity Using Sensor (DHT11). Available at: https://www.instructables.com/Measuring-Humidity-UsingSensor-DHT11/ (Accessed: 13 June 2023).

[14] DEPARTMENT, M.M. and MINISTRY OF NATURAL RESOURCES, E.& C.C. (no date) Malaysia's Climate. Available at: https://www.met.gov.my/en/pendidikan/iklim-malaysia/#:~ : text=The average monthly relative humidity,5%25 (Alor Star) (Accessed: 13 September 2023).

[15] Dejan (2019) Ultrasonic Sensor HC-SR04 and Arduino – Complete Guide. Available at: https://howtomechatronics.com/tutorials/arduino/ultrasonicsensor-hc-sr04 / (Accessed: 13 September 2023).

[16] https://electronicsworkshops.com/2024/01/28/iot-based-flood-monitoring-system-using-esp-32-and-ultrasonic-sensor/

[17] https://www.mdpi.com/2624-6511/6/3/68