Faculty of Electrical and Electronic Engineering Technology

DEVELOPMENT OF IoT BASED FLOOD MONITORING SYSTEM USING ESP32 AND NODE-RED FOR PREVENTIVE OF NATURAL DISASTER

SITI NUR LYANA KARMILA BINTI NOR AZMI

Bachelor of Electronics Engineering Technology (Telecommunications) with Honours

2022

DEVELOPMENT OF IoT BASED FLOOD MONITORING SYSTEM USING ESP32 AND NODE-RED FOR PREVENTIVE OF NATURAL DISASTER

SITI NUR LYANA KARMILA BINTI NOR AZMI

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electronics Engineering Technology (Telecommunications) with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2022

DECLARATION

I declare that this project report entitled "DEVELOPMENT OF IoT BASED FLOOD MONITORING SYSTEM USING ESP32 AND NODE-RED FOR PREVENTIVE OF NATURAL DISASTER" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Student Name : SITI NUR LYANA KARMILA BINTI NOR AZMI

Date : 17/01/2023

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electronics Engineering Technology (Telecommunications) with Honours.

Signature : ....

Supervisor Name : TS. MOHD FAIZAL BIN ZULKIFLI

Date : 11/01/2022

Signature : ....

Co-Supervisor : ....

Name (if any)

Date : ....

ABSTRACT

Natural disasters such as floods have major impacts such as loss of life and economic losses that cannot be avoided, but proper planning can decrease the terrible aftermath. A flood warning system generally incorporates data on telemetric precipitation and water level estimated at several locations around the local region. It is challenging to provide information regarding river conditions, flood kinds, and so on based on these observations. In the absence of a real-time monitoring system, it is impossible to warn authorities and provide protective measures in the event of a major incident. As a result, an enhanced flood forecasting system must be installed and developed. It is necessary to implement an end-to-end flood forecasting, warning, and response system capable of preparing for an emergency evacuation. It is suggested to construct a new flood monitoring and warning system that is simple, cost-effective, uses little electricity, and is simple to deploy and operate. To address such issues, a IoT based system which include Ultrasonic Sensor, Node-Red and ESP32 is presented.

ABSTRAK

Bencana alam seperti banjir memberi impak besar seperti kehilangan nyawa dan kerugian ekonomi yang tidak dapat dielakkan, tetapi perancangan yang betul boleh mengurangkan kesan buruk. Sistem amaran banjir secara amnya menggabungkan data tentang kerpasan telemetrik dan paras air yang dianggarkan di beberapa lokasi di sekitar wilayah tempatan. Adalah mencabar untuk memberikan maklumat mengenai keadaan sungai, jenis banjir, dan sebagainya berdasarkan pemerhatian ini. Sekiranya tiada sistem pemantauan masa nyata, adalah mustahil untuk memberi amaran kepada pihak berkuasa dan menyediakan langkah perlindungan sekiranya berlaku insiden besar. Akibatnya, sistem ramalan banjir yang dipertingkatkan mesti dipasang dan dibangunkan. Adalah perlu untuk melaksanakan sistem ramalan, amaran dan tindak balas banjir hujung ke hujung yang mampu menyediakan untuk pemindahan kecemasan. Adalah dicadangkan untuk membina sistem pemantauan dan amaran banjir baharu yang mudah, menjimatkan kos, menggunakan sedikit tenaga elektrik, dan mudah untuk digunakan dan dikendalikan. Untuk menangani isu tersebut, sistem berasaskan IoT untuk menjangka kejadian akan datang dan memberi amaran tepat pada masanya dibentangkan.

ACKNOWLEDGEMENT

First of all, thanks to my parent for giving encouragement, enthusiasm and invaluable assistance to me. Without all this, I might not be able to complete this subject properly.

Second, I would like to thanks to my project supervisor, TS. MOHD FAIZAL BIN ZULKIFLI for the guidance, inspiration, motivation, and encouragement he provided during the course of the project study. He has always been nice and patient enough to listen to the queries and solve them, in spite of a very busy schedule. I dedicate the project's success to him and his team which has been very supportive from the beginning of the project to the end.

My humble thanks to my family members, colleagues and last but not least, the board of UTeM's 2021/2022 Students Representative Council for the continuous support and idea in completing this project. I apologize to all the other unnamed who helped me and making this project, what it has come out to be.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

1.	V	Volt
2.	cm	Centimetre
3.	m	Metre
4.	€	Pound (Europe currency)
5.	Hz	Hertz
6.	mA	milli Ampere

LIST OF ABBREVIATIONS

1.	API	Application Programming Interface
2.	PVC	Centimetre
3.	API	polyvinyl chloride
4.	TRIG	Trigger on HC-SR04 Ultrasonic Sensor Pinout
5.	ECHO	Echo on HC-SR04 Ultrasonic Sensor Pinout
6.	VCC	Voltage Common Collector
7.	GND	Ground
8.	GPIO	General-Purpose Input/Output
9.	ESP32	Espressif Systems
10.	SMS	Short Messaging Service
11.	WiFi	Wireless Fidelity
12.	Tx	Transmitter
13.	Rx	Receiver
14.	GSM	Global System for Mobile
15.	IDE	Integrated Development Environment
16	DC	Direct Current
17.	C/C++	Programming Language C/C++
18.	JSON	JavaScript Object Notation
19	IoT	Internet of Things
20.	IBM	International Business Machines
21.	PWM	Pulse Width Modulation
22.	MCU	MicroController Unit
23.	PC	Personal Computer
24.	LCD	Liquid Crystal Display
25.	LED	Light-Emitting Diode
26.	GPS	Global Positioning System
27.	UI	User Interface
28.	USB	Universal Serial Bus
29.	CSV	Comma Separated Values

CHAPTER 1
INTRODUCTION

1.1 Background of Study

According to research that was published in 2018 by the Emergency Events Database (EM-DAT), Malaysia has been hit by at least 51 natural disasters over the course of the last 20 years. This nation is regularly struck by a variety of natural disasters, including earthquakes, tsunamis, floods, haze, and landslides. One of the most common types of natural disasters that strike a wide range of areas and populations throughout the nation on an annual basis is flood. It poses a direct risk to individuals, as well as to natural resources, the environment, and the economy, and it also causes harm. Figure 1.1 provides an overview of the significant disasters that have struck Malaysia between the years 1965 and 2016, broken down by the total number of persons impacted. (*"Dam Pre-Release as an Important Operation Strategy in Reducing Flood Impact in Malaysia,"* 2018).

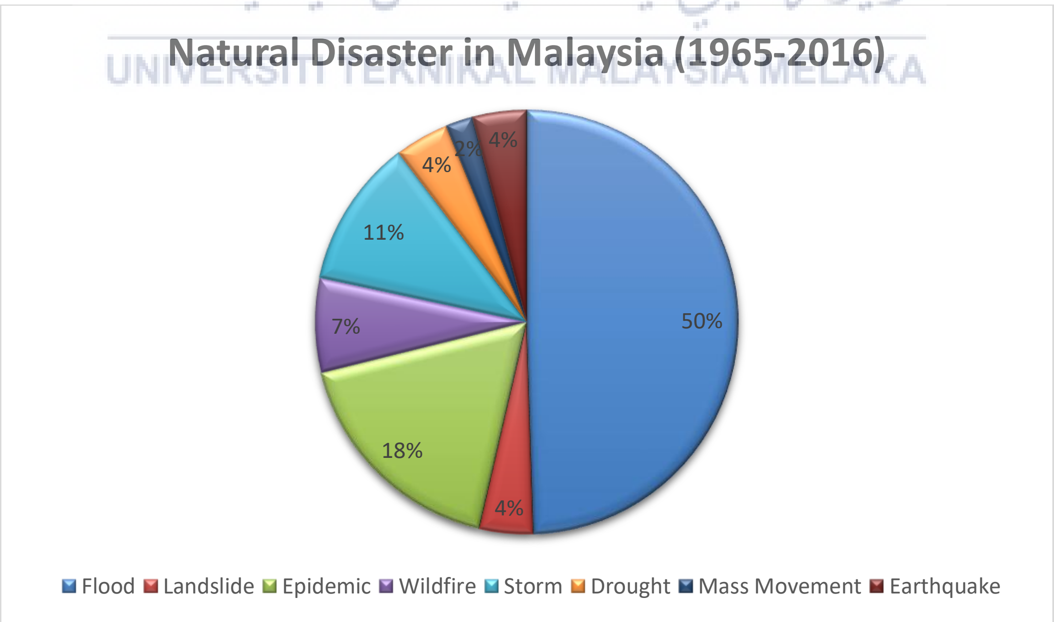


Figure 1.1: Natural Disaster in Malaysia (1965-2016)

In Malaysia, there are two types of floods which is flash floods and monsoonal floods. The fast river rises with shallow waters that exceed the bank of the river cause flash floods. It is most likely to happen near the river. It can sometimes happen even when no rain has occurred, such as when levees or dams fail or when debris causes a sudden discharge of water. Flash floods are widespread in desert locations with poor drainage systems.

While, monsoonal floods usually occur in the Northeast Monsoon Season. Monsoons are mainly caused by changes in temperature between land and water caused by radiation from the sun heating. The monsoon season happens in between of November and March, bringing heavy rains to the east coast states of Peninsular Malaysia and west of Sarawak. Monsoonal flood threatens people, natural resources, as well as creates economic damage. Sadly, this flooding claims more lives in Malaysia each year compared to other natural disasters which is why flood is a heavy concern of the country.

One of the technologies that can be utilised to reduce the number of fatalities caused by floods is the installation of flood monitoring and warning systems. These systems are particularly useful in the states along the east coast of Malaysia, such as Terengganu, Kelantan, and Pahang, as well as west of Sarawak. It is possible to connect the alarm system to the system so that it will inform both the local inhabitants and the higher authorities. An Internet of Things-based flood monitoring system is the name given to this kind of apparatus. The purpose of this system is to keep track of how high the water level is in the river and notify the user if it reaches a level that is considered to be dangerous. It is anticipated that national rescue teams such as PDRM, JPAM, and BOMBA would make use of this technology.

1.2 Problem Statement

The vast majority of flood monitoring systems are based on telemetry systems, which need data to be sent to a central terminal through transmitters, repeaters, and other associated hardware. When the equipment in a portion of the detected zone fails to perform properly, this approach is both expensive and unreliable. Other methods are unreliable because they depend on the communication infrastructure provided by other vendors. As a consequence of all this, a system that is trustworthy and reasonable in cost that is based on a wireless sensor network is needed.

1.3 Project Objective

This project's major objective is to develop an effective flood monitoring and warning system with IoT-based applications and built-in major flood mitigation. The following below are the specific objectives:

1. To study the flood monitoring and warning alert system.
2. To develop a Malaysian model of IoT based Flood Monitoring System using ESP32 and Node-RED for Preventive Natural Disaster.
3. To analyze a prototype for the proposed project.

1.4 Scope of Project

To eliminate any doubt regarding the project's scope caused by numerous limitations and constraints, the hardware of this project will be focusing more on in-house testing in which only indoor components and devices will be utilized. Besides that, a mock measurement for the threshold will be used in the wireless water level detection sensor system before triggering the warning system devices. This project is a system to monitor the water level and to give an alert to local netizens about the flood status so then, they could evacuate before the flood hit them. Equipped with ultrasonic sensors and ESP32, the water level may be able to detect and Node-Red's Dashboard will be used to display the result of the water level and the status of the danger of the water level. If the water level passed the Danger level, the Node-Red then sends a warning to the local netizen.

1.5 Expected Result

This project will achieve the project's objectives which are to design an IoT-based flood monitoring system and able to give an alert to the authorities or local netizens to evacuate the place if the water level passed the danger threshold. This system is intended to give an early warning via the Node-Red platform. The ultrasonic sensor will connect to the ESP32 and wirelessly transport the gained data of the water level to the Node-Red before Node-Red could send an alert message to the local authorities.

1.6 Thesis Outlines

There are five chapters in this thesis including of introduction of the project, the literature review which is the works of the others related to this project, and lastly the method used to implement the knowledge into the project.

Chapter 1: In Chapter 1, briefing about general ideas of the project which are an introduction, problem statement, target of the project, scope of the project, project significant, and thesis outlines.

Chapter 2: In this project, basically study the literature review which is work that is related to the project. It is important in order to obtain some knowledge about the project.

Chapter 3: Discusses the methodology, which is consists flowchart of the whole project and the description of the component that will be used to solve the problem statement. Furthermore, this chapter includes some explanation about software and hardware development and also the main component of the project.

Chapter 4: Discusses in detail the outcome of the project in a logical way. The analysis of the outcome is also presented within this chapter.

Chapter 5: Summarizes the contribution of all studies to the project and provides concluding remarks. In addition, there are some recommendations for future works.

1.7 Summary of Chapter 1

In a conclusion, Malaysia was severely affected by the flood, especially during the monsoon season. As the natural phenomenon is untouchable, a flood monitoring and warning system are required to help the authorities and local netizens to evacuate sooner. With the display of the water level on the Node-Red Dashboard, the authorities will be able to do make an evacuation preparation.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

2.1.1 What is Flood ?

A flood is an outpouring of water that submerges normally dry ground. Floods are the most common form of natural disaster, and they are frequently caused by heavy rain, fast snowmelt, or storm surge from a tropical cyclone or tsunami that occurs in coastal regions. Floods can be divided into 3 categories. These are storm surges, dam failures, and flash floods. Storm surges are most likely caused due to the rapid rise in water levels along with the high speed of water flow. Besides that, dam failures are caused by the volume of the flood exceeding the capability of the dam could hold. Lastly, storm floods are caused due to the natural phenomenal contribution such as severe onshore winds that came along with low atmospheric pressure and high tides.

During the monsoon season in Malaysia, which runs from October to March every year, flooding is a typical occurrence, especially along the eastern coast of the nation. In most cases, it takes place in towns or metropolitan regions like Kuala Lumpur, Johor Bahru, and Pulau Pinang despite the fact that these locations have insufficient drainage systems that are unable to manage the excessive amount of precipitation. The year 2021 came to a close with a tropical storm making landfall on the eastern coast of Peninsular Malaysia. This event was responsible for three days' worth of intense rainfall over the peninsula. There were eight states on the peninsula that were affected as a direct consequence of the floods. Over the course of its most severe phase, it impacted a total of more than 125,000 people (Wikipedia Contributors, 2022) and resulted in the relocation of around 71,000 people at the same time.

It is the worst flood the nation has seen since the Malaysia floods of 2014–2015. Damage was widespread throughout the states of Selangor and Pahang, notably in the region of Hulu Langat and the city of Shah Alam. The rainfall levels reached all-time highs at the meteorological stations in Selangor and Kuala Lumpur. P. Prem Kumar, a reporter for Nikkei Asia, claims that the government of Malaysia has been criticised for its delayed reaction and disinterest in the wake of the tragedy.

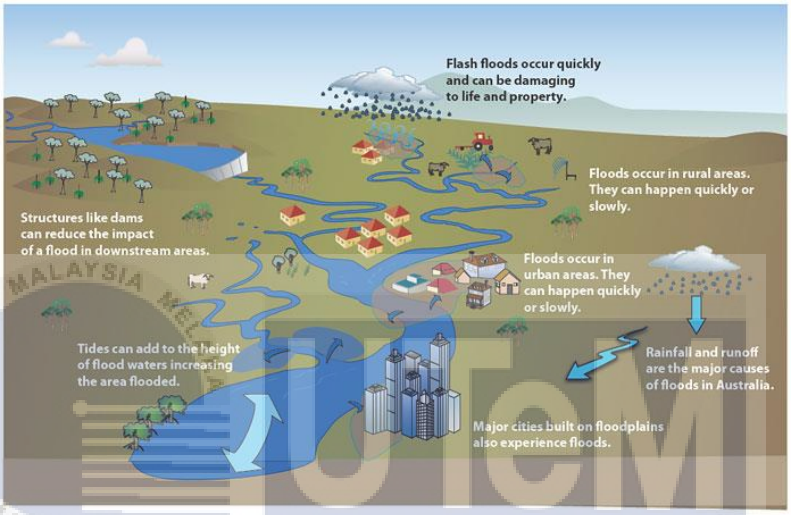


Figure 2.1: Flood Flow

Figure 2.1 illustrates the variables that lead to floods, one of which is the flow of water from a high point to a low point. This implies that low-lying regions may be drowned fast before reaching higher land (Ramizu, 2015). These factors differ across locations and dates, resulting in no two floods being the same.

2.1.2 Type of Flood That Occurs in Malaysia

According to the authors of "Performance Analysis of IoT-based Flood Monitoring Framework in Suburban" (2021), the disaster has been divided into two main categories by Malaysia's Department of Irrigation and Drainage (DID), which are flash floods and monsoon floods. Both of these types of floods are extremely dangerous. The length of time it takes for the water to return to its regular place is what allows meteorologists to differentiate between monsoon floods and flash floods. For the sake of designing technologies that would prevent hasty mass evacuations, it is essential to have a solid understanding of the behaviour of the floods that occur in Malaysia.

A flash flood is a kind of flood that happens within six to seven hours following heavy rain, and it often happens during the first two hours of the heavy downpour. It is distinguished by a rapid increase in the river's level, with water depths that extend beyond the river's banks. Near the river is the most probable location for it to take place. There is a risk of flash flooding if thunderstorms hit the same location at the same time. On the other hand, the likelihood of sudden flooding is reduced when storms move at a faster pace since the rain is spread out over a larger region. It is possible for it to occur even when there has been no rain at all, such as when spillway or dams fail or when debris causes a rapid flow of water from a dam or reservoir.

In Malaysian cities, flash floods are the hydro-meteorological occurrence that occurs most often and causes the greatest damage. The nation's capital city is seeing an increase in the frequency of sudden floods compared to previous 10 years. Even while flash floods may not always take place during monsoon seasons, this is the period of year in the city when they are most likely to take place. Even though there are various ways to prepare for and adapt to flash floods, they continue to be a significant risk, especially in urban areas. Not only are flash floods hazardous and potentially disastrous due of the intensity of the water itself, but they may also be catastrophic because of the flying debris that can be swept up in the rush of the water.The illustration of the flash flood may be observed in Figure 3 below.

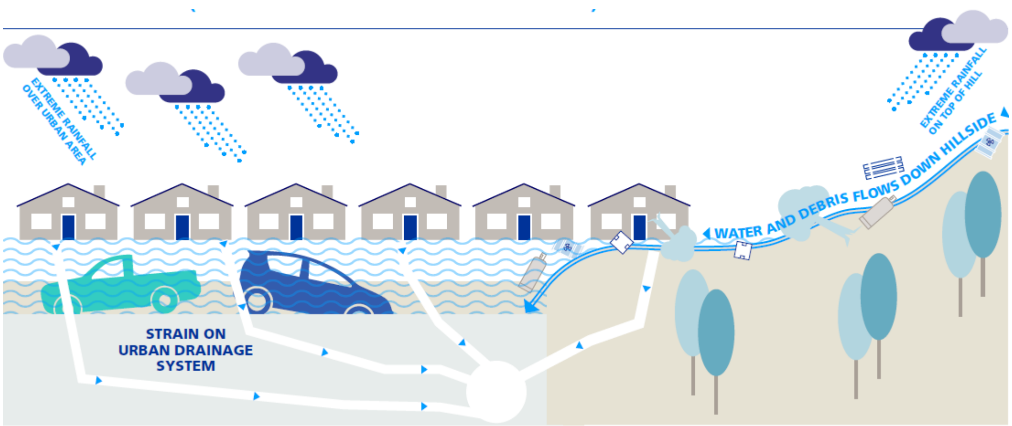


Figure 2.2: The illustration of the flash floods

Northeast monsoon flooding is the cause of monsoon floods. Monsoons are mainly caused by changes in temperature between land and water caused by radiation from the sun heating. The monsoon season begins in early November and lasts until March, bringing significant rains to the east coast states of Peninsular Malaysia and western Sarawak.

The Northeast Monsoon occurs as cold air moves out of Siberia's lowlands. While atmospheric pressure rises, resulting in the formation of a significant high pressure system. The blast of cold air combines with low pressure systems or storms that occur near the equator, causing strong winds and waves in the South China Sea as well as heavy rains on Peninsular Malaysia's east and west coasts. During the North-East monsoon, the wind blows heavy rain to Peninsular Malaysia, which frequently causes floods. The floods in December 2021 were among the worst in Peninsular Malaysia, impacting many states, and Pahang got it worst. This tragedy has caused damage on people, property, crops, animals, and infrastructure.

2.2 Previous Works

There are two types of flood systems that are often used which is flood monitoring and warning systems and flood monitoring alone. Flood monitoring operates in conjunction with flood forecasting to determine if a flood warning should be given to the public or whether previous warnings should be withdrawn. Figure 9 below shows the basic block diagram of the flood monitoring system which briefly explains how the flood monitoring system operates.

2.2.1 Flood Monitoring and Warning System

Flood Monitoring and Warning System (2015) written by Muhammad Ramizu Ab Halim. In this paper, Ramizu uses the solar system to supply energy in this project. GSM shield is used by the system. The ultrasonic sensor will transmit a signal to the GSM shield, which will analyze the data and send a brief message to the authorities or flood-affected areas. When an ultrasonic sensor is used as an input. In this system, ultrasonic sensors and a GSM shield are combined in a microcontroller. Based on the figure below, this project is utilizing a wireless-based system in which the GSM module operates for both the sensor network and SMS network system.

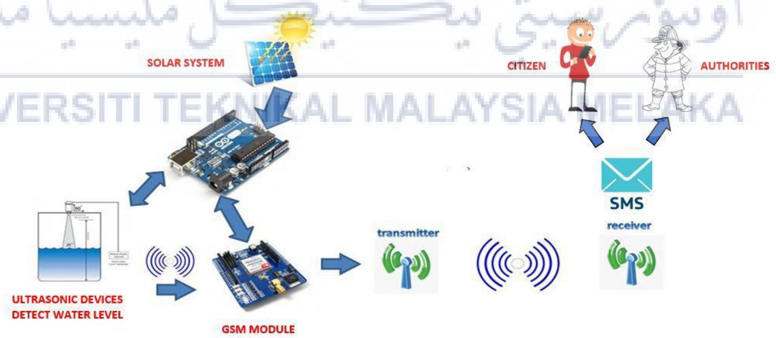


Figure 2.3: The system architecture of the project

The project is separated into two installations for the installation process which is in-house testing and outdoor testing. With the consent and collaboration of the Department of Irrigation and Drainage (JPS) Perak Tengah, the planned system was installed and tested in

Rumah Pam Sungai Perak. To monitor the changes in the water level, water level data was gathered and updated on a regular basis.

During field testing, some of the items that are used include PVC pipe, an ultrasonic sensor, and a water float. In the pump building, the required PVC pipe is installed in the suitable area. The actual water level measurement taken at the Trenches pump house is used to calibrate the ultrasonic sensors used in this system. It has a maximum range of measurement of 400 centimetres. The volume of water present in the pump house will inform the configuration choices that are made for this system. This technology might be used if the typical height of the water in the pump house is lower than 400 centimetres. The pump building in Perak Tengah that will be used for field testing may be seen shown in the image below.

Figure 2.4: The field-testing pump house in Sungai Perak.

According to Ramizu, this project follows two flowcharts, which is one is for the Sensing Device and the other one is the flowchart of the GSM Shield. The flow chart below demonstrates how the ultrasonic sensor functions and sends signals to the GSM shield. Ultrasonic sensors will be installed and tested. When the water level in the set increases, a signal is sent to the GSM shield, and when the water level is normal, no action is being taken.

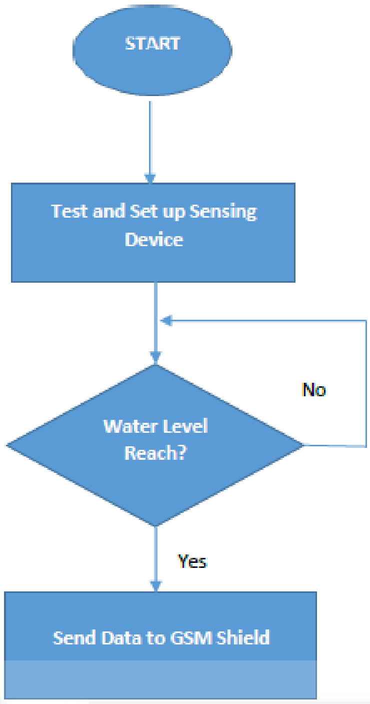


Figure 2.5: The flowchart of the Sensing Device

This information will be sent to the GSM shield system to complete the SMS send procedure. This configuration can communicate with one another. For example, following the GSM shield, it will send an SMS warning to acquire fresh signals from sensing devices. If the sensing system receives ultrasonic sensor data on the status of the water in the river, the sensing system will communicate with GSM configuration for updating the system.

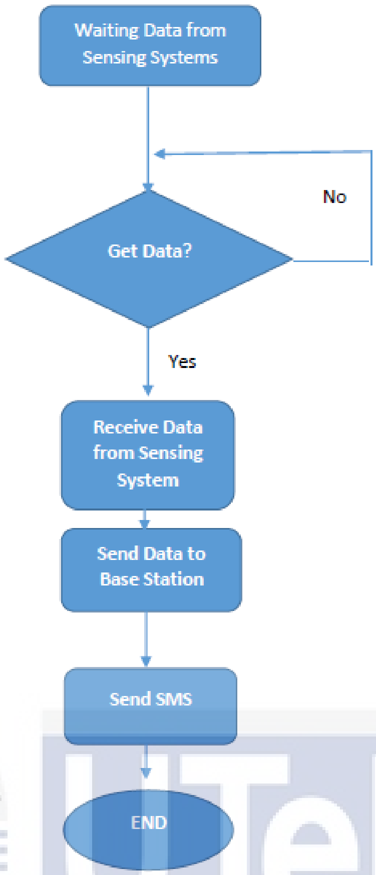


Figure 2.6: The flowchart of GSM Shield.

The GSM Shield, together with the ultrasonic sensor, is inserted within the microcontroller in the GSM setup. GSM Shield's main function is to process data in the form of SMS and transmit SMS to flood-affected people or authorities. GSM Shield will send SMS notifications to the base station before sending them to the population's authorities. It will interact with the base station for the SMS system if it receives data from sensing systems. This system requires programming to function in order to send an SMS.

As for the conclusion, Ramizu suggest that for the future recommendation, instead of using the SMS system, a web application may be developed, and the number of sensor nodes can be increased to provide a more comprehensive system and the result data would be much more accurate.

2.2.2 Low-Cost IoT-based Flood Monitoring System Using Machine Learning and Neural Networks: Flood Alerting and Rainfall Prediction

This project is written by Dola Shiba Rani, Dr. Jayalakshimi G N and Dr. Vishwanath P Baligar on 2020. Based on their writings, this project is basically focuses on rainfall prediction system in which utilizing Linear Regression Model, Support Vector Machine, and also Artificial Neural Networks. Both Linear Regression Model and Support Vector Machine models uses the rainfall data gained in the last three months to make a prediction of the rainfall in the next consecutive month.

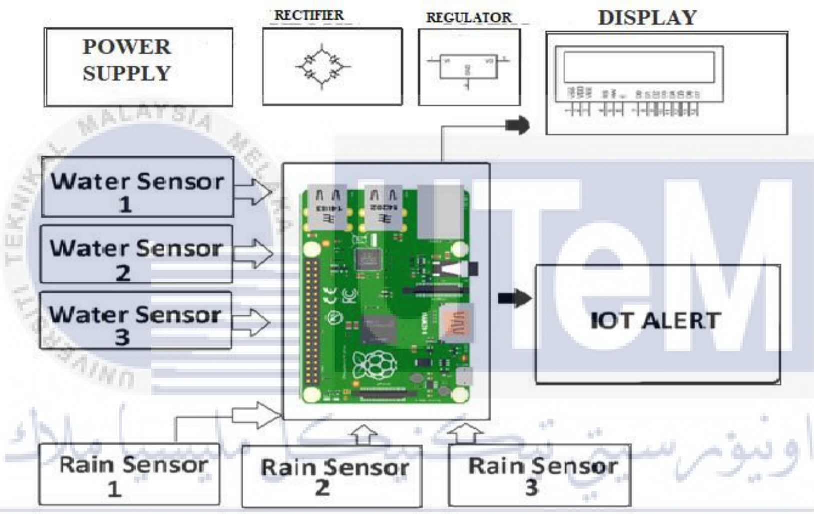


Figure 2.7: Block diagram of the IoT architecture

Based on the block diagram shown above, the main components are 3 water float sensors, 3 rain sensors, a single Raspberry Pie, and an LCD display. The sensors will be located at three different places. This is because is to gain an accurate data before a decision been made. When the water level in the body of water rises, the buzzer beeps and the system sends an alarm message, including the amount of time until the region floods.

Engineering Tools	Specification
Rain sensor x3	• Pins : 4 • Consist of GND and VCC
Water Flow Sensor x3	• Consists of GND and VCC • Average Life : 15 years
Rasberry Pi	• Uses DC supply, keyboard and mouse • Operating System : NOOBS

Table 2.1: Table of specification of the devices used in this study

Based on the table above, the rain sensors is to measure the intensity of rainfall in millimeters while the water sensors main purpose is to measure the water level and actuates the pump or initiates an alarm by a buzzer. The NOOBS is implemented to automatically generated by installing from local software or from a remote repository as NOOBS is operated by the Raspberry Pie. IoT Gecko is an open platform with API support and used to display the data of the rainfall readings.

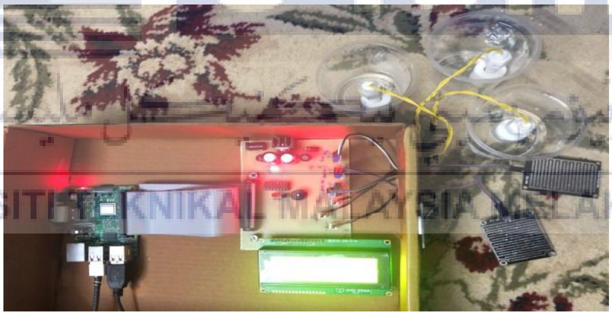


Figure 2.8: The prototype circuit of the project

This prototype shown in the figure above measures the water level in the fflood-pronearea, while three water sensors and raindrop sensors estimate the strength of the flood. The rainfall sensors will measure the raindrop in mm/hr and display the measurement on the LCD display.

In conclusion, this study is started just to prove that the machine learning model could be used for the prediction of rainfall in any state globally. However, to build a rainfall prediction system, meteorological data, history, and a large number of sensing devices are all necessary.

2.2.3 Prototype of Google Maps-Based Flood Monitoring System Using Arduino and GSM Module

This project was published in the International Research Journal of Engineering and Technology (IRJET) and written by Dedi Satria, Syaifuddin Yana, Rizal Munadi, and Saumi Syahreza. In this project, the main components are Arduino UNO, SIM9000 GSM Module, GPS U-Blox Neo 6M Module, Ultrasonic Senso r HC-SR04, and Wavecom USB Fast Track.

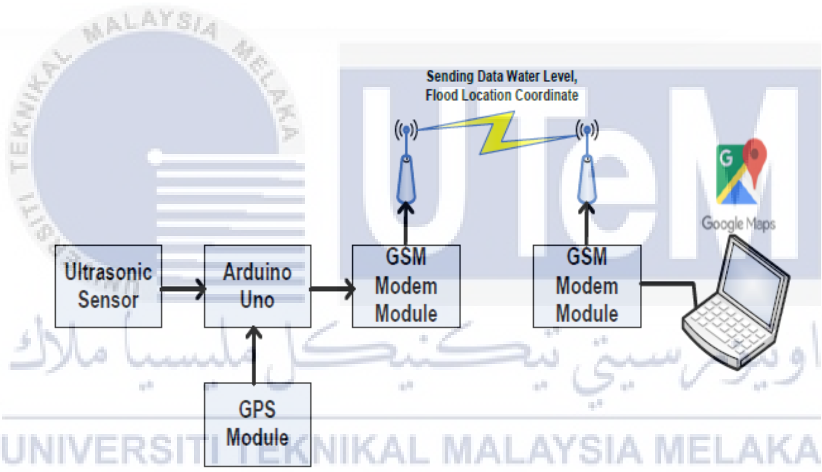


Figure 2.9: Simple block diagram of the project’s operation.

Based on the block diagram above, the ultrasonic sensor will be the input of the system where ultrasonic sensor will detect the level of the water. Beside that, GPS Module will also act as the input in which to detect the location of the ultrasonic sensor. Then, the gained data will be encoded by the Arduino Uno to transmit to the computer and processor to be a water level information system based on Google Maps by using GSM Module in the form of SMS data.

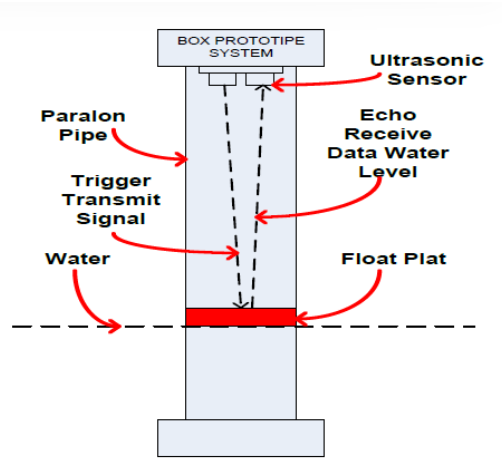


Figure 2.10: The architecture of the prototype of the Water Height Detection

The prototype shown in the figure above was built with a paralon pipe and a floating plate as a water level object The ultrasonic sensor is attached on top of the pipe, which emits signals from the trigger components, which are reflected by the ultrasonic sensor and collected by the echo component. The sensor will convert signals into water level data. The figure below demonstrates the construction of the Water Height Detection

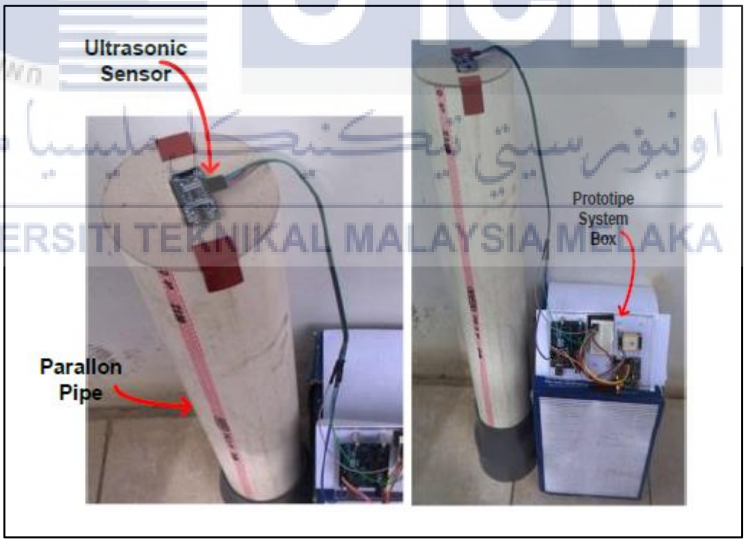


Figure 2.11: The construction of the Water Height Detection

In conclusion, This research project uses a different approach which is in this project includes GPS and Google Maps tools to detect the potential flood area. This system generates a flooded information system based on Google Maps with information of water level data and flood location coordinates.

2.2.4 Smart IoT Flood Monitoring System

Smart IoT Flood Monitoring System (2019) is written by Shahirah Binti Zahir, Phaklen Ehkan, Thennarasan Sabapathy, Muzammil Jusoh, Mohd Nasrun Osman, Mohd Najib Yasin, Yasmin Abdul Wahab, N.A.M Hambali, N.Ali, A.S.Bakhit,F.Husin, and M.K.Md.Kamil R.Jamaludin. This project is a flood detection and alert using website and wireless approach. The website will display the data of the water level measured. Besides that, clients can control the buzzer and stepper motor of the gate wirelessly as this project used WiFi Module. Other than alert the citizens using the alarm of the buzzer, this project also using stepper motor to open the gate to let the excess water flows out to decrease the water level.

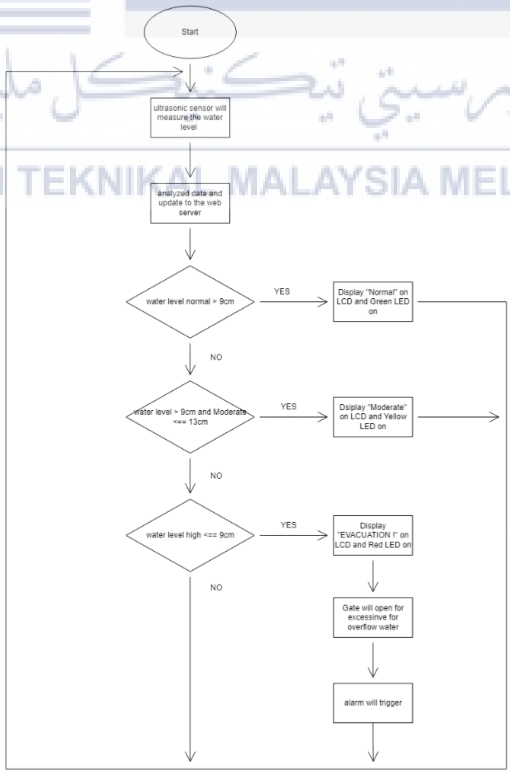


Figure 2.12: The flowchart of the project

According to the flowchart in Figure 4, the Smart IoT Flood Monitoring System is designed to notify the public in the area of an approaching flood. The procedure begins when an ultrasonic sensor detects the amount of water in the river. The sensor data is collected and transferred to the microcontroller, where it is displayed on the web server. Data will then be analyzed and compared. The stepper motor and buzzer could be controlled remotely by the user. Based on the information gathered, the flood status will be calculated. As a result, the water level status will be shown on the LCD and web server.

The water level will be indicated via an LED. Furthermore, when the stepper motor reaches the greatest threshold value, the alarm is activated to notify the public. As a result, residents will be well prepared to evacuate before the flood occurs.

Figure 2.13: The dashboard of the Smart IoT Flood Monitoring System Web Server

The water level is shown on the web server of the Smart IoT Flood Monitoring System, which may be seen in the figure referenced above (Figure 2.13). In addition, the user has the ability to control the buzzer, which serves as a warning signal to the general public, as well as the stepper motor, which serves as a gate to allow excess water that was caused by the flood out of the system. They are required to press the "send" button after selecting the radio buttons that control the digital out pins in the device. The data is then sent to the server, where it is read, and the server is given the ability to operate the motor and the buzzer.

2.2.5 Smart IoT Flood Monitoring System

This paper written by Annisa Jamali, and Jonathan Peter Giman. Based on this study, two distinct flood monitoring systems have been created in this system framework. These systems included hardware, software, and a database. Sensory data obtained by the system is employed to notify the public in advance before flooding. IoT-based floods monitoring systems may be placed at neighboring rivers by installing sensors in the hardware system to continually detect factors such as river water level and rainfall pattern. Continuously measuring such factors with sensors will offer up-to-date information on the state of river level. Using suitable wireless communication methods, the monitored data is transferred to the remote infrastructure service provider.

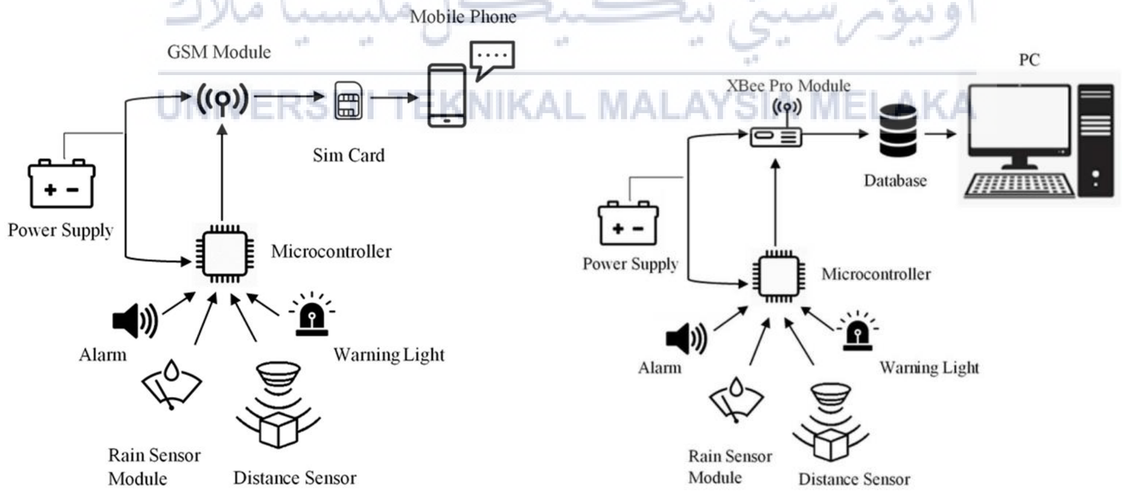


Figure 2.14: The System Architecture

Based on the Figure 2.14 above, the system using GSM SIM 900A module to send an alert message via SMS and XBee Pro module is used to transferring the data to the software database which is Node-Red. Node-Red will plotted a real-time graph of the water level while saved to the CSV files.

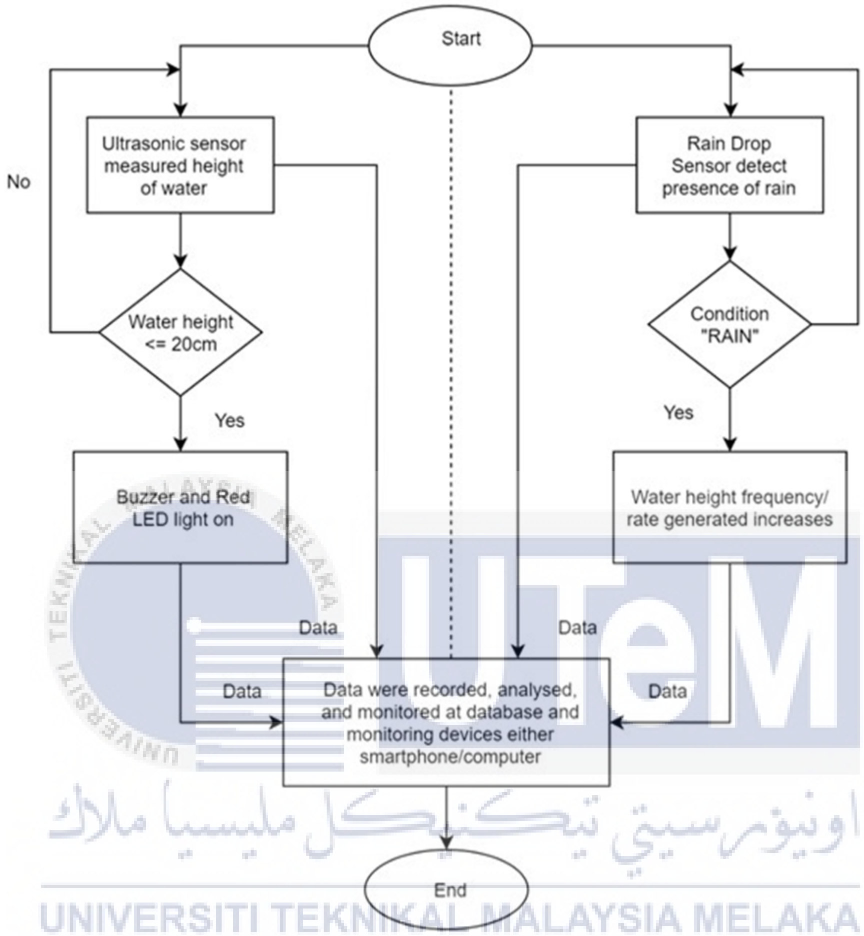


Figure 2.15: The Flowchart of this project

The flowchart is separated into two section which Section 1 addressing the water level control meanwhile Section 2 addressing is to detecting the rain and provide the rain rate information. The distance between the water's surface and the ultrasonic sensor was measured in the first section. When the distance was less than 20 cm, the situation was considered to be in an unsafe. The alert components would then be triggered.

Using a wireless IoT platform, the received data was transferred to the operator's PC or smartphone. Other than that, the gathered data also would be transferred to the smartphone over the GSM module. The information was shown via text message using the Short Messaging Service (SMS) application. The data gathered can be accessed from the sim card.

2.2.6 Design and Implementation of Internet of Things and Cloud Technology in Flood Risk Mitigation

This research case study is taking place in Jakarta. This system was implemented at a small scale to help mitigate flood risk in big cities such as Jakarta. The system was built using an IoT device that has a water level sensor. The data collected by the device can then be stored in DynamoDB. DynamoDB's water level time series data can be presented in a web or mobile app. This system can also notify the authorities and citizens in case of water level exceeds a predefined threshold.

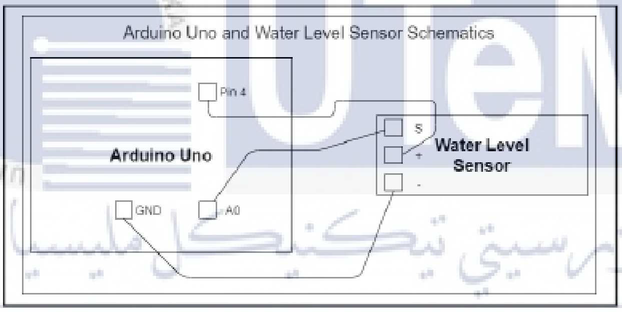


Figure 2.16: The IoT schematization with Arduino UNO and Water Level Sensor of the project

The hardware used in this project is as shown the Figure 2.16 above which is Arduino Uno, and Water Level Sensor with an additional of Raspberry Pi 3 Model B++. For the application, Ubidots, its server, and AWS DynamoDB are implemented. The communication API is REST API, and the communication protocols are MQTT. For the web service, AWS IoT Core and Ubidots are used.

Sensors will be installed above the surface at flood hotspots. When numerous sensors reach a specific water level, the sensors will communicate the data to the AWS cloud, which will subsequently deliver an event to the user. If the water level continues to fall and the requirements are not satisfied after a specific duration, another message will be sent.

DynamoDB is a key-value and document database that performs in single digit milliseconds at any size. The Nodejs program is used on the Raspberry Pi to gather sensor data from AWS, which is why the Raspberry Pi must be connected to the internet through ethernet or Wi-Fi. In this situation, Node RED will be used. If everything is in order, the sensor values should appear on the AWS IoT Core.

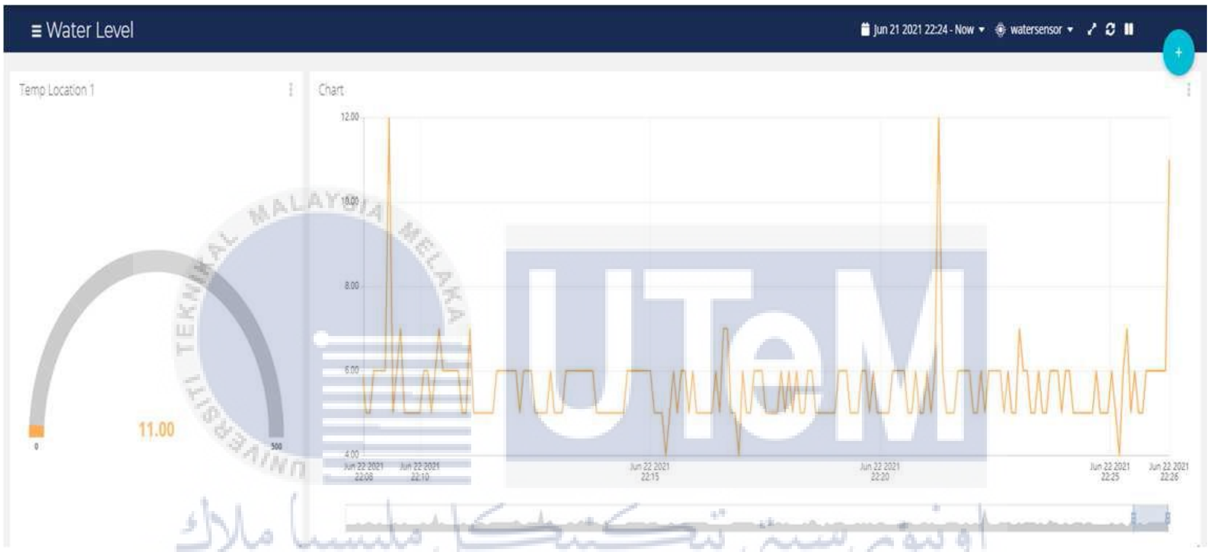


Figure 2.17: Dashboard for the Water Level

In the Dashboard, as shown in Figure 2.17, a gauge and a graph were displayed showing the behavior of the water level. A basic application visualizes data from a DynamoDB database using high charts and can also manage the pump remotely by pressing or clicking a button that uploads an event to the AWS IoT platform when pushed or clicked. After processing the event delivered to the IoT platform, the Raspberry Pi will act accordingly.

2.2.7 LoRa Based Real-Time Flood Detection and Monitoring System: A Brunei Darussalam Based Study

This study was take place in Brunei and written by Muhammad Ihsanuddin Haji Md Yassin, Au Thien Wan, S. H. Shah Newaz, and Saiful Omar. The software implementation for flood monitoring is Node-Red while, Node.js act as their back end server. The main result of the gained data was then viewed through the Grafana dashboard which is as shown as in the Figure 2.18 below.

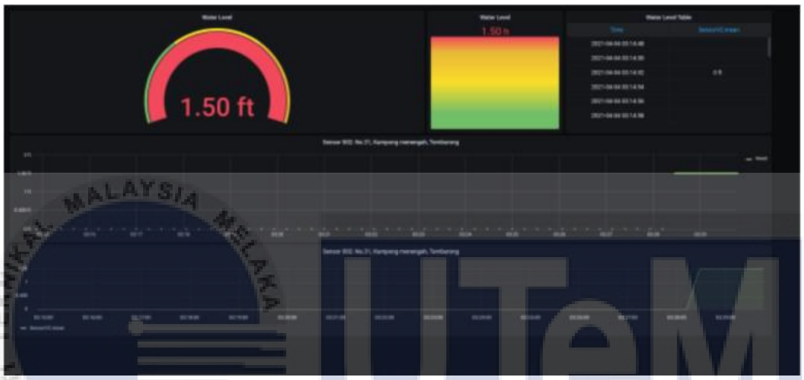


Figure 2.18: The Grafana dashboard

The project consisted of a roadside flood detector that interacted with a backend server through lightweight data transfer, ideally via a LoRa network, and a front end that let end users to view the dashboard. The backend might also store data, process data, and do activities like sending flood alerts and forecasting flooding.

The sensor nodes, which were effectively roadside water level sensors, transmitted data to the server in the system architecture, as shown in Figure 2.19. Node-RED was implemented as the server for the event-driven prototype. It might be used at the network's edge, on low-cost hardware like the Arduino board, and in the cloud. Node-RED was linked to InfluxDB, an open-source time series database (TSDB). InfluxDB analysed the real-time data obtained from sensor nodes and showed it on the dashboard. Grafana was employed as a real-time monitoring platform for the dashboard. Grafana may display numerous data sets at the same time in the form of a gauge, bar graph, or time graph.

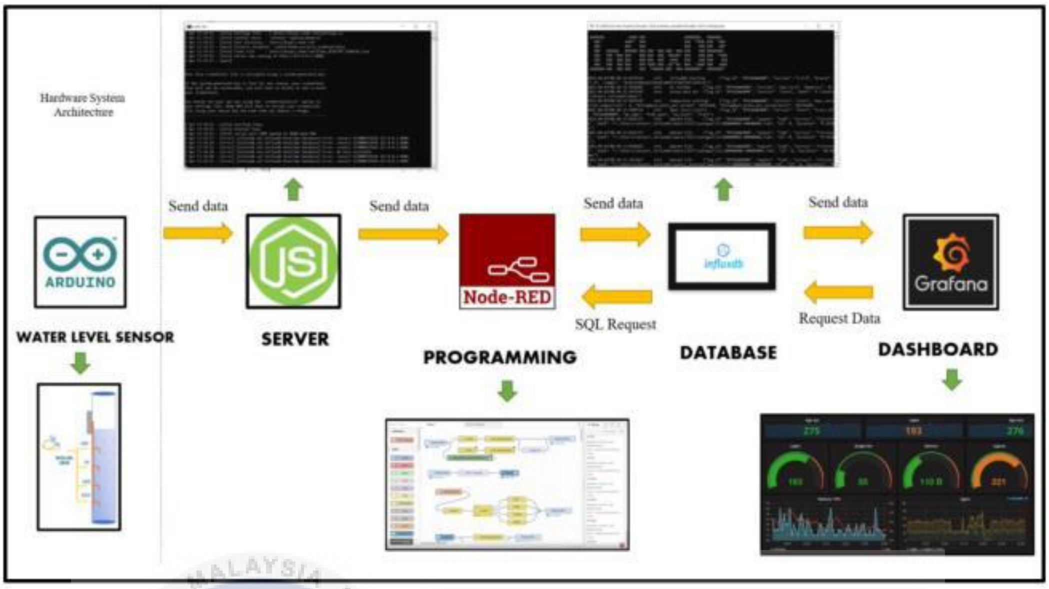


Figure 2.19: The architecture of the project

The roadside flood sensor was constructed up of water level sensors and was installed next to the road, as shown in Figure 2.20 below. When precipitation happened and water began to rise to a specific height, the sensors detected the rise and transmitted data to the backend for computation. The increase would be shown on the dashboard, alerting end customers. Four sensors were employed to detect four levels of water, including 0.1ft, 0.5ft, 1.0ft, and 1.5ft.

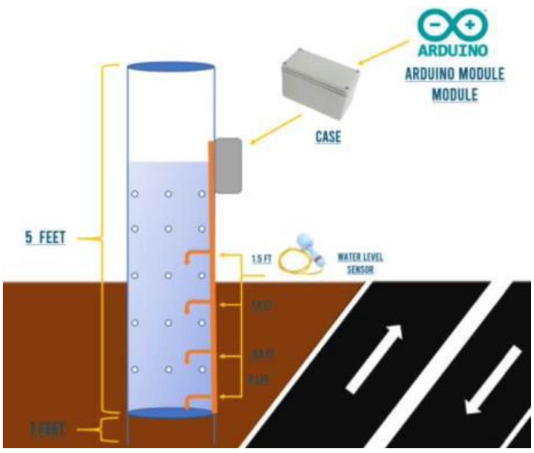


Figure 2.20: The roadside flood detector

2.2.8 A Prototype for Flood Warning and Management System using Mobile Networks

This paper is written by Syed Ahmad Ali, Fasih Ashfaq, Ehsan Nisar, Usama Azmat, and Jehan Zeb. In this study, the main components used are flow sensor, ultrasonic sensor, rain detector, LTE device, ESP32 and SIM800L GSM module. One of the sensors employed was the HC-SR04, an ultrasonic module that monitors water depth. When a signal is emitted by the transmitter, a high frequency back-reflected sound is produced when the signal detects an item.

The second sensor is a rain detector. When rain falls on the sensor, it acts as a switch. When rainfall is detected, it generates a digital output of 5 or 0 volts. The rotor of the water flow sensor rotates as the water runs through it, and the rotation speed increases as the water flow increases.

Then, every couple of seconds, the ESP-32 sends telemetry data to the MATLAB-based IOT-cloud platform. The IOT platform utilizes a MQTT protocol in addition to the TCP/IP standard. The sensor data from various units is analysed to average out the inaccuracy. This architecture could be observed based on the illustration of the Figure 2.21 below.

In the beginning, as shown in Figure 2.21, the data from each unit is collected by three sensors which is a rain sensor, a water flow sensor, and a water level sensor. This information is then sent to an ESP32 device. Following that, the MQTT protocol is used in order for the ESP32 module to transmit the data to the IoT platform. The IoT platform analyses incoming telemetry data to determine risk thresholds that have been specified for each sensor. These risk thresholds are then used by the IoT platform to evaluate the present flood danger level. On the Server PC is where a database holds all of this information is maintained. This includes telemetry data as well as the flood risk level.

Three steps are conducted based on the current flood risk level. First, the user receives an SMS message through a GSM module. Second, the user may sign in to the designated Android app to receive real-time updates on the current flood. Finally, the actuator, in this case, the dam's spillway, is controlled using the labeled data to lower the flood danger in the impacted region.

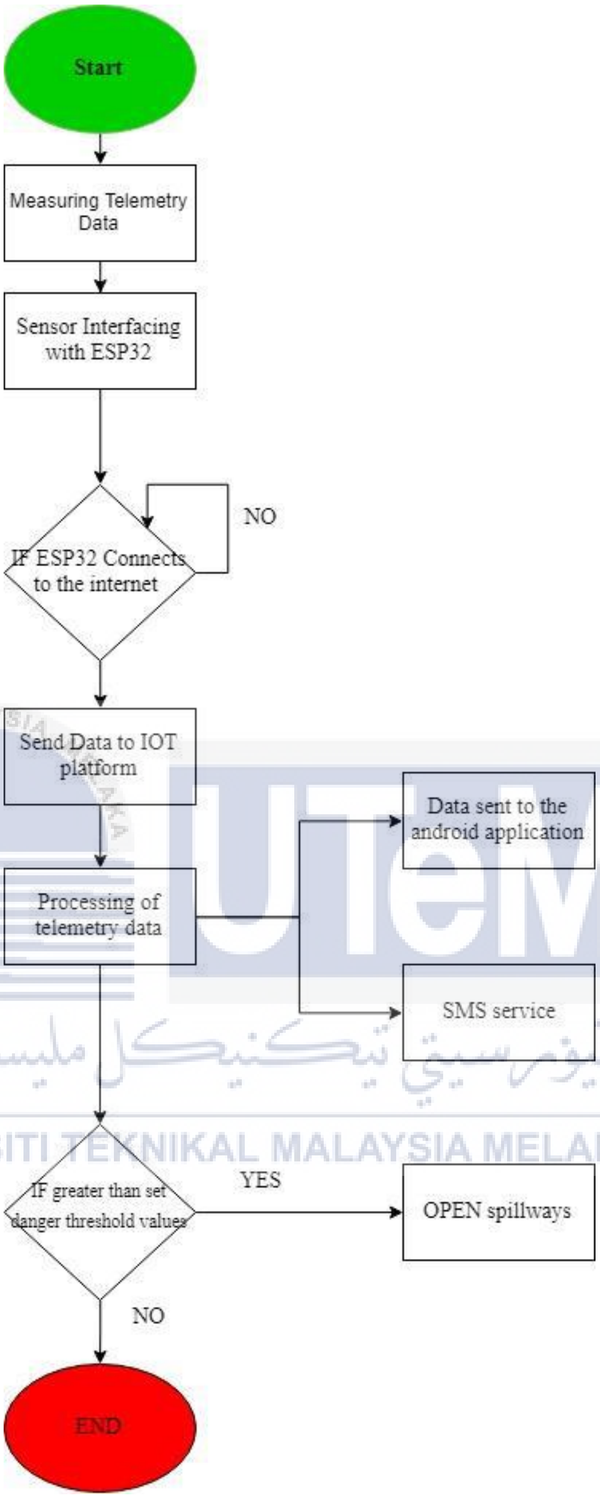


Figure 2.21: The flowchart of the project

Lastly, the authors suggested that a third warning notification system can be added using the LoRa WAN protocol with the ESP unit. This technology has the potential to bring more flexibility to the architecture and warning distribution strategies since it does not rely on data networks for data transfer and is completely independent of them. Another conceivable innovation is the use of a rain gauge sensor to measure rainfall, which will add complexity to the monitoring information.

2.2.9 Design of Automatic Water Flood Control and Monitoring Systems in Reservoirs Based on Internet of Things (IoT)

In this project, the main components used are Arduino Mega, Ultrasonic sensor and ESP32 module. Meanwhile, the software implementations are Arduino IDE and Web Host. This project is written by Yuliarman Saragih, Jandoni Horan Prima Silaban, Hasna Aliya Roostiani, and Agatha Elisabet S on the year 2020. According to the authors, the stated that a suitable approach is automated control and monitoring systems in reservoirs.

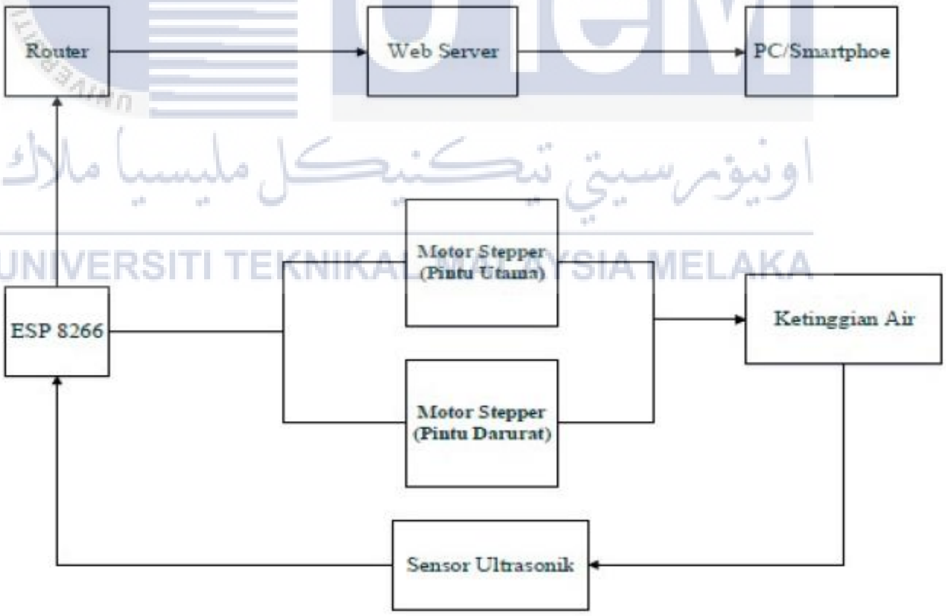


Figure 2.22: The block diagram of the work system of the project.

According to Figure 2.22, this prototype employs an HC-SR04 ultrasonic sensor to monitor water level and volume. The sensors have a detection range of 2 cm to 400 cm.

However, the component used for this experiment is to monitor changes in water level between 3 and 20 cm.

The electric actuator used in the prototype actuator is a stepper motor with the driver module. The floodgate is controlled by the motor, which opens and closes the flow of water. The main door remains open until the water condition returns to normal, at which point it shuts automatically. The emergency door remains closed until the water condition returns to normal, at which point it closes automatically.

As for monitoring system, a web server monitoring based is implemented. As shown in the Figure 2.23 below, the display results are the real-time height and the volume of the water, the status of the risk of the water level, and the status of the door way. Besides that, the dashboard also includes buttons of doorway to control the actuator wirelessly.

Figure 2.23: The dashboard of the project

2.2.10 Early Detection of Flood Monitoring and Alerting System to Save Human Lives

This study is written by Shivashankar, Jijesh J J, Dileep Reddy Bolla, Mahaveer Penna, Sruthi P V, and Alla Gowthami in the year 2020. In this paper, the study is mainly concerned with developing a system for notifying and monitoring floods in various regions. This study is conducted in India. To broadcast the alarm message, this project employs two systems which is Blynk and via SMS. If there is no access to Wi-Fi or internet in the designed region, a simple text message is sent to the registered phone numbers using the GSM module. This project also includes a weather forecast report that is updated every three hours.

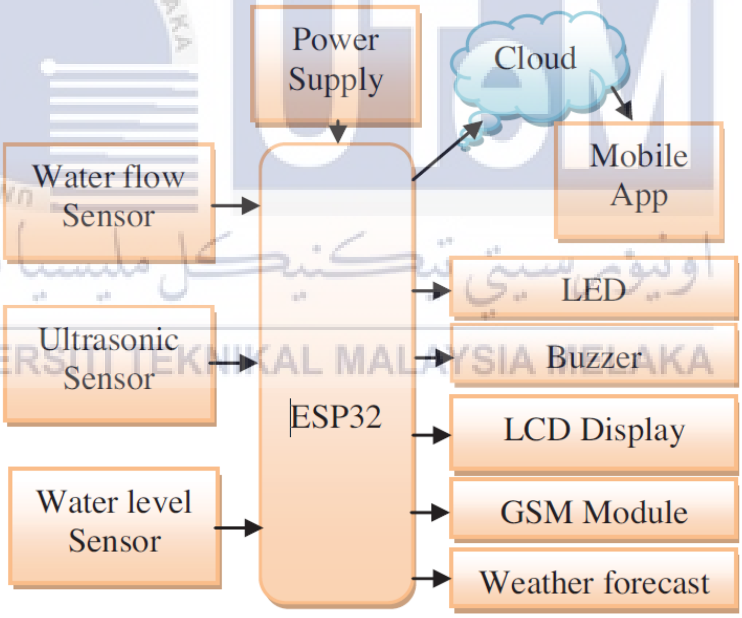


Figure 2.24: The block diagram of the study

This project's major components are a water flow sensor, an ultrasonic sensor, a rain sensor, a GSM module, and an ESP32 microcontroller. The water flow sensor is responsible for detecting the speed of the water flow. Meanwhile, the major purpose of the water level sensor is to determine the water level in dams or rivers. These outcomes will then be shown on the LCD display and shared via the Blynk application.

Once the threshold condition is met, where the value is set based on where the model is stored, the LED and buzzer are activated for dangerous situations, and an alarm message is delivered via the Blynk application as well as SMS via the GSM modem. Meanwhile, the weather forecast is utilized to keep track of the weather.

2.2.11 Comparison of Previous Works

Name of Study	Components/Hardware/Software used				Brief operations
	Sensing System	Monitoring System	Alerting System	Microcontroller	
Flood Monitoring and Warning	• Ultrasonic Sensor	Not applied	• GSM Shield	• Arduino ATMega 328	When the sensor transmit the data of the water level to the GSM shield, the GSM module then send the SMS to alert users.
Low-Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks: Flood Alerting and Rainfall Prediction	• Rain Sensor • Water Flow Sensor	• IoT Gecko • LCD Display	• Not applied	• Raspberry Pi	The sensor will collect the data and will be preprocessing in the Raspberry Pi. Then, the results will display on LCD and IoT Gecko
Prototype of Google Maps-Based Flood Monitoring System Using Arduino and GSM Module	• Ultrasonic Sensor • GPS Module	• Google Maps	• GSM Module	• Arduino Uno	The design of the system begins with an ultrasonic sensor, and the GPS module sends water level data and flood position coordinates to an Arduino Uno, which serves as a data processor. Both data are provided as SMS messages to the

					modem's receiving information system station. The data is received by a computer and processor, which is used to develop a water level information system based on Google Maps.
Smart IoT Flood Monitoring System	• Ultrasonic Sensor	• Web Server	• Buzzer • LCD • LED • Stepper Motor	• ARM Mbed NXP LPC1768 Microcontroller • ESP8266 Module	This project is a flood detection and alert using website and wireless approach. The website will display the data of the water level measured. Besides that, clients can control the buzzer and stepper motor of the gate wirelessly as this project used WiFi Module. Other than alert the citizens using the alarm of the buzzer, this project also using stepper motor to open the gate to let the excessive water flows out to decrease the water level.
Performance Analysis of IoT based Flood Monitoring Framework in Sub-urban	• Ultrasonic Sensor • Rain Sensor Module	• XBee Pro Module • PC	• GSM Module • Buzzer	• Arduino Uno	The distance between the water's surface and the ultrasonic sensor will be measured. The scenario was considered threatening when the distance was lesser than 20 cm. The alert components are then activated. The rain sensor is then built to detect the presence of rain and deliver information on the frequency or rate of precipitation. The gathered data was sent to the operator's PC or smartphone through a wireless IoT platform. The collected data will be sent to the smartphone over the GSM platform. The data was delivered by text message using the Short Messaging Service (SMS) application.

					The obtained data may be retrieved via the sim card. The gathered data will be sent to the XBee Pro platform.
Design and Implementation of Internet of Things and Cloud Technology in Flood Risk Mitigation	• Water Level Sensor	• Raspberry Pi • AWS IoT Core • Dynamo DB • Node.js • Ubidots Web	• AWS Iot Core	• Arduino UNO	The device's collected data may then be saved in DynamoDB. Water level time series data from DynamoDB may be shown in a web or mobile app. water sensor and a Raspberry Pi as the device that will send an alert when turned off and record the water level when turned on If the water level persists over a certain level for a long period of time, the system will send another notification.
LoRa Based Real-time Flood Detection and Monitoring System: A Brunei Darussalam Based Study	• Water Level Sensor	• InfluxDB • Grafana	• Pushbullett	• Arduino	This project used Node-RED and Node.js to monitor floods, and the main result was shown on the Grafana dashboard. When the water level sensor reaches a specified value, a PushBullet alert is delivered to the clients.
A Prototype for Flood Warning and Management System using Mobile Networks	• Flow Sensor • Ultrasonic Sensor • Rain Detector	•	• SIM800L GSM Module • Servo Motor	• ESP32	The ESP-32 transmits telemetry data to the IOT-cloud platform, which is based on MATLAB. In addition to the TCP/IP standard, the IOT platform uses the MQTT protocol. It is based on a publish-subscribe messaging technique designed for remote location connections with limited network bandwidth. The sensor data from several units is analysed in order to average out the inaccuracies. The processed data will then be

					labelled depending on the specified threshold. There are two ways to notify the user of the flood risks. An Android app that provides a real-time flood warning as well as a map-based representation of the level of flood risk in a person's location. When the flood danger level is high, the spillway gates open to a larger extent, and when the flood risk level is low, the gates open to a smaller extent. The greater the risk of flooding, the greater the opening of spillways and vice-versa.
Design of Automatic Water Flood Control and Monitoring Systems in Reservoirs Based on Internet of Things (IoT)	• Ultrasonic Sensor	• Web Host	• ESP32 • Stepper Motor • Actuator	• ATMega 8535 Microcontroller	Once the threshold condition is met, where the value is set based on where the model is stored, the LED and buzzer are activated for dangerous situations, and an alarm message is delivered via the Blynk application as well as SMS via GSM modem. Meanwhile, the weather forecast is utilized to keep track of the weather.
Early Detection of Flood Monitoring and Alerting System to Save Human Lives	• Water Flow Sensor • Ultrasonic Sensor • Water Level Sensor	• LCD Display • Weather Forecast	• LED • Buzzer • GSM Module • Blynk App	• ESP32	This project uses two systems which is Blynk and SMS to send the alert message. If there is no access to the Wi-Fi in a particular area the simple text message is sent via the GSM modem to the registered phone numbers. They also enhance the existing project by adding on a weather forecast report that is updated every 3 hours.

Table 2.2: The Table of Comparison

2.3 Hardwares

2.3.1 ESP32

The ESP32 is a well-known low-power system-on-chip microcontroller with built-in Wi-Fi and dual-mode Bluetooth, which makes it ideal for creating and prototyping IoT applications. The ESP32 may function as a stand-alone system or as a slave device to a host MCU, eliminating communication stack overhead on the primary application CPU. Through its SPI / SDIO or I2C / UART interfaces, the ESP32 could communicate with other systems to offer Wi-Fi and Bluetooth functionalities.

The ESP32 has 39 digital pins in total, 34 of which can be used as GPIO while the remainder are input-only pins. The microcontroller has 18 12-bit ADC channels and 2 8-bit DAC channels. It also includes 16 PWM signal generating channels and 10 GPIO pins that provide capacitive touch functionalities.

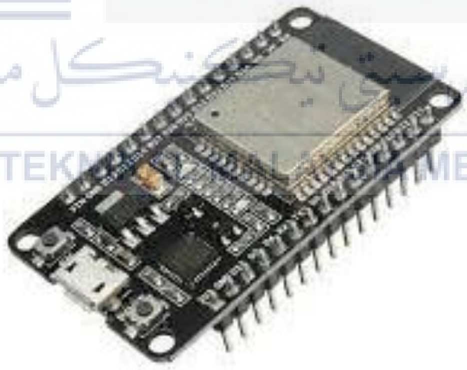


Figure 2.25: ESP32

ESP32 could be simply programmed using Arduino IDE with a proper library installed in the IDE. This microcontroller can be coded using various kinds of programming languages. As an example, C, C++, and even Phyton language. Furthermore, ESP32 is a low-cost component as it only costs not more than RM30 according to the Cytron Technologies Website.

2.3.2 Ultrasonic Sensor

Ultrasonic sensors operate by emitting a sound wave with a frequency higher than human hearing. The sensor's transducer functions as a medium to receive and transmit the ultrasonic sound waves. Like many others, our ultrasonic sensors employ a single transducer to emit a pulse and receive the echo. Table 2.3 below is the table of Ultrasonic Sensor's specification of the sensor:

Specifications	
Dimension	45mm x 20mm x 15mm
Power Supply	5V DC
Working Current	15mA
Working Frequency	40kHz
Maximum Range	4m
Minimum Range	2cm
Measuring Angle	15^0
Resolution	0.3cm
Trigger Input Signal	10us TTL Pulse
Echo Output Signal	TTL pulse proportional to the distance range

Table 2.3: Table of Specification of HC-SR04 Ultrasonic Sensor

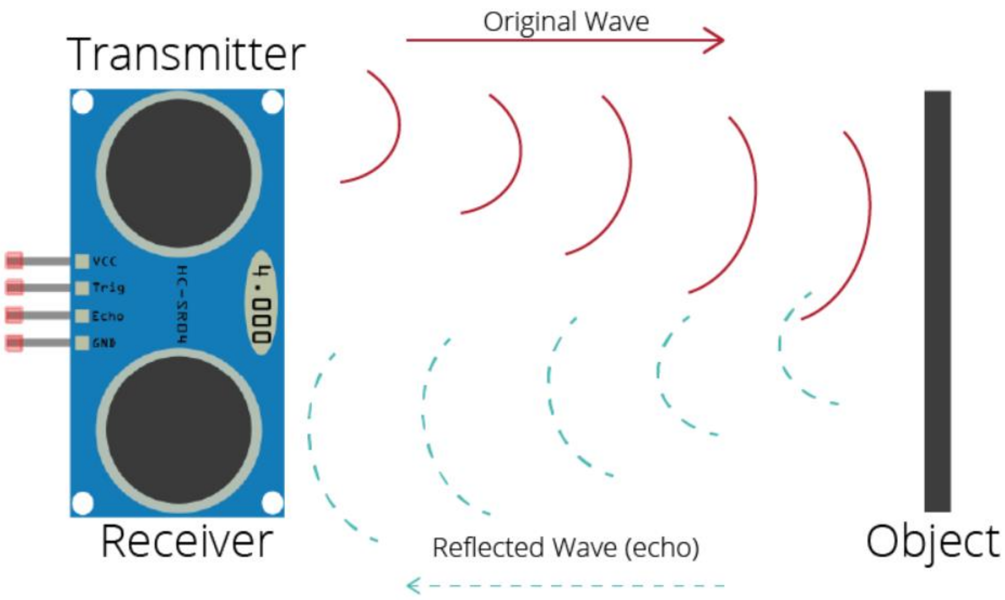


Figure 2.26: The behavior of the Ultrasonic Sensor

According to Figure 2.25 above, the transmitter terminal of the sensor emits a high frequency which is a 40kHz sound to the object. As the sound wave travels through the , the wave reflected back to the sensor and is received by the receiver terminal of the sensor. The velocity of the sound wave signal is calculated as the formula below :

$$Distance\ of\ the\ wave\ to\ the\ object = \frac{speed\ of\ sound\ in\ air\ x\ time\ (s)}{2}$$

The pin of the Ultrasonic Sensor are consist of 4 pins which is VCC pin, Trig pin, Echo pin and GND pin. Table 2.4 below shows the pinout of the sensor:

HC-SR04 Ultrasonic Sensor Pinout	
Pin	**Purpose**
VCC	Input voltage (5V)
Trig	Wave Receiver terminal
Echo	Wave Transmitter terminal
GND	Common ground

Table 2.4: HC-SR04 Ultrasonic Sensor Pinout

2.3.3 Buzzer

A buzzer is a tiny yet effective component for adding sound to our project/system. Buzzers are generally available in two varieties. The one basic buzzer that, when powered, produces a continuous beep sound, whereas the other type is a premade buzzer, which is bigger and produces a repetitive beep sound.

Figure 2.27: Buzzer 6-12V c/w Wire

This buzzer could be operated by simply connecting it to a 4V to 9V DC power supply. A basic 9V battery can also be used, however a controlled +5V or +6V DC supply is much more preferred. The buzzer is generally connected to a switching circuit that turns the buzzer on and off at the desired time and interval. The Table 2.5 below shows the specifications and features of the buzzer:

Specifications	
Parameters	**Value**
Rated Voltage	6V DC
Operating Voltage	4-8V DC
Rated Current	Less than 30mA
Resonant Frequency	Approximately 2300Hz

Table 2.5: Specification of Buzzer

2.4 Softwares

2.4.1 Node-Red

Node-Red is a Node.js-based Open-Source flow-based tool, IoT platform, and dashboard developed by IBM. Node-red is suitable for constructing IoT and home control dashboards and automation, as well as many other applications, due to its versatility and user friendly. Node-red could be installed directly on Windows, Linux, and, the Raspberry Pi.

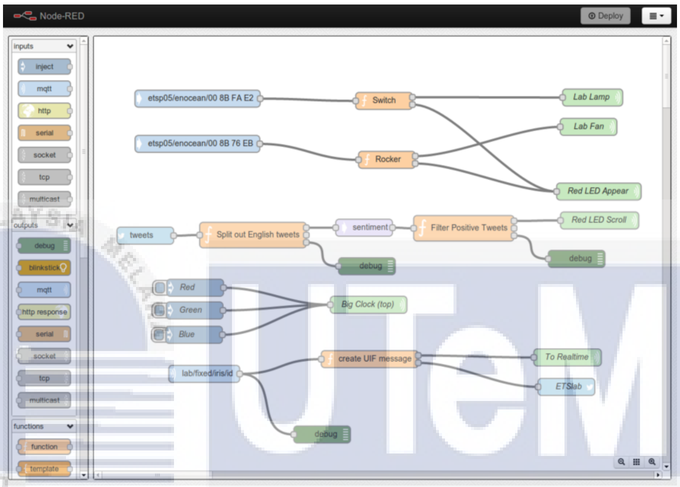


Figure 2.28: Node-RED Flow Editor

Node-Red provide a web browser-based flow editor. This application elements can be stored or shared for future use. Node.js is used to build the runtime and JSON is used to store the flows made in Node-RED. Table 2.6 below shows the project that include in Node-Red.

Name	**Description**
Node-RED	A visual tool for connecting the Internet of Things
Node-RED Dashboard	Node-Red Dashboard user interface
Node Generator	A command-line tool for generating Node-RED node modules from a variety of sources, including the Open API document and the code of a function node.
Node-RED Command Line Tool	People may remotely administrate a Node-RED instance using a command-line tool.

Table 2.6: Project of Node-Red

2.4.2 Arduino IDE

In contrast to Node-Red, Arduino IDE is an Open-Source software for writing and uploading code to Arduino boards. The IDE is compatible with a variety of operating systems, including Windows, Mac OS X, and Linux. It is compatible with the programming languages C and C++. It is not only able to program Arduino boards, but it also can code non-native boards with an appropriate installation of an Arduino can ore-based library. As example, ESP32 and ESP8266.

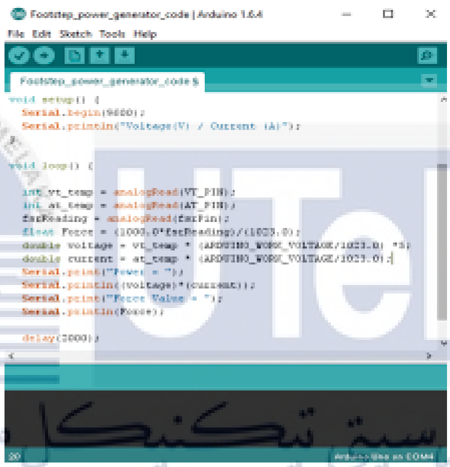


Figure 2.29: The Arduino IDE

The major text editing application used for Arduino programming is the Arduino Integrated Development Environment (IDE). It is where the user is able to write their code before uploading it to the board you're coding. The platform is shown in the Figure 2.29 above. Sketches are the term given to Arduino code. It is necessary to utilize the most recent version of the Arduino IDE to avoid any difficulties during the uploading process.

The IDE basically converts and compiles user sketches into code that Arduino can interpret. After the Arduino code has been compiled, it will upload to the memory of the board. If any mistakes are found in the Arduino code, a warning message will pop up, encouraging the user to make corrections before recompile the sketch.

CHAPTER 3
METHODOLOGY

3.1 Introduction

In order to successfully complete this project, implementation of both hardware and software are required. The ultrasonic sensor, the buzzer, and the ESP32 are going to be the three most important elements of the hardware design, and the project will be developed using Node-Red in conjunction with the MQTT protocol. The ultrasonic sensor will determine how much water is there and then transmit that information to the ESP32 so it can be decoded. The information will then be displayed on the Node-Red Dashboard when it has been processed. In the case that the water level rises over the dangerous threshold, the ESP32 will deliver a message of caution to the relevant authorities as well as the residents. The buzzer will sound an alarm to notify those who live in the area. When it comes to the actual implementation of software, Node-Red is mostly used to programme the whole system. Both local authorities and citizens will be informed of the situation by the message notification that is issued by Node-Red.

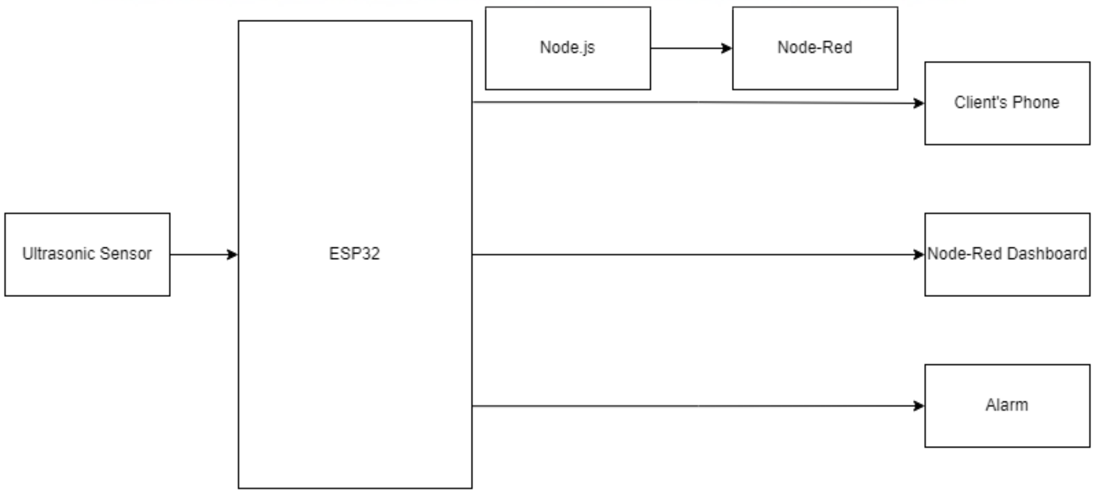


Figure 3.1: Block Diagram of the project

The Ultrasonic Sensor is shown as the input in the block diagram, thus this is what the prototype receives. In addition to this, the microcontroller that is being used is an ESP32, and the output involves of alert messages being sent to the customer's phone through SMS, the Node-Red Dashboard, and the alarm or buzzer. The system's back-end server is implemented using Node.js.

3.2 Flowchart

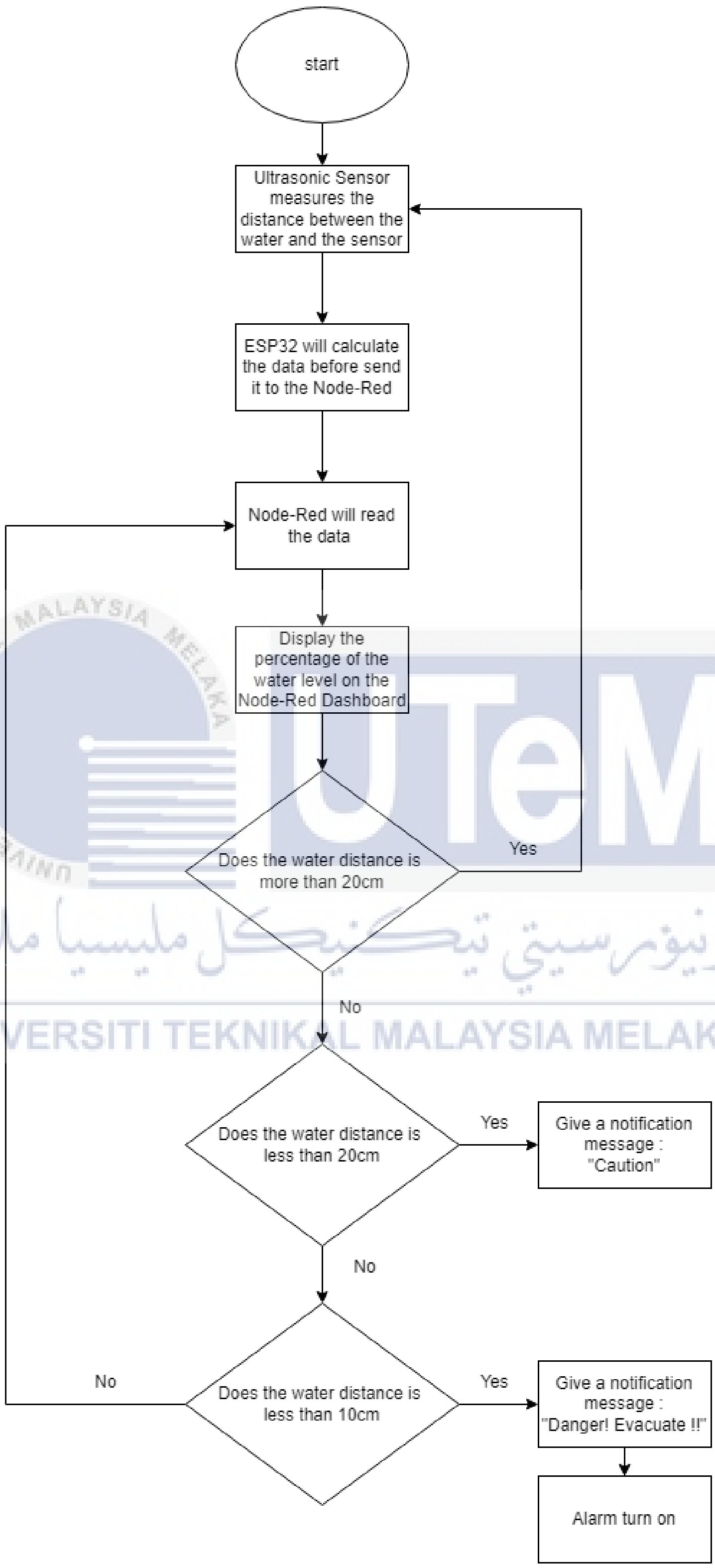


Figure 3.2: Flowchart of the project

The proposed project's flowchart may be seen in Figure 3.2, where it is shown as a flowchart. The distance between the Ultrasonic Sensor and the water will be measured by the Ultrasonic Sensor. As a result of the water level increasing, the centimetre measurement will become decreasing. Both the Node-Red and the ESP32 will be linked to a backend server that uses the open source Node.js server environment for data storage. The computation will be performed on the ESP32, and the results will then be sent to the Node-Red.

The Node-Red will display the percentage of the water level on the Node-Red Dashboard. The buzzer will sound when the distance between the sensor and the water is less than 10cm and Node-Red will send an "Danger ! Evacuate !" message to the flood-prone residents and to the authorities. Meanwhile, if the distance measured in between 19cm and 11cm, an alert message write "Caution !" will be sent to the residents and the authorities. As for more than 20cm, no action will be taken.

3.3 Hardware Implementation

3.3.1 5V DC HC-SR04 Ultrasonic Sensor

The primary objective of the ultrasonic sensor that is being used in this project is to determine the distance within the ultrasonic sensor from the surface of the water. Since the length of the UPVC Pipe is 1 foot, which is equal to 30.48 centimetres, the maximum value of the measurement should be more than 30 centimetres. Using the sensor's brief ultrasonic pulse, we are able to quantify the amount of time it took for that pulse or the echo to travel to and from the liquid.

Figure 3.3: HC-SR04 Ultrasonic Sensor

3.3.2 Node-MCU ESP32

The ESP32 is serving as the project's microcontroller. The communication stack overhead that would normally be performed by the main application CPU may be avoided by using the ESP32 either as a stand-alone system or as a slave device to a host MCU. The ESP32 is able to interface with other systems in order to provide Wi-Fi capabilities. This is particularly important considering that this project will design a wireless-based architecture. In addition to it, the Arduino Software Development Kit (IDE) is used to write code for the ESP32.

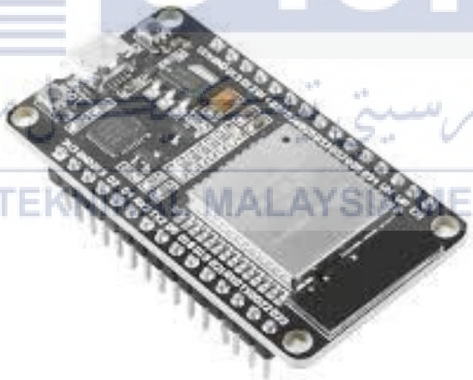


Figure 3.4: Node-MCU ESP32

3.3.3 Buzzer 6-12V c/w Wire

In this project, the buzzer will activate once the distance between the sensor and the top of the water is less than 10 centimetres. The buzzer is controlled by the signal that is sent by the ESP32, which is written using the Arduino IDE. The flexible ferromagnetic disc is attracted to the coil when the buzzer is engaged, and it stays in place there until the buzzer is turned off. The signal is oscillated via the coil, which causes the buzzer to produce a changing magnetic field, which in turn causes the disc to vibrate and collide. The buzzer will sound once this movement has been made.

Figure 3.5: Buzzer 6-12V c/w Wire

3.3.4 SIM900A GSM GPRS Module

SIM900A Modem is based on the Dual Band GSM/GPRS SIM900A modem from SIMCOM. It runs at 900 or 1800 MHz frequencies. SIM900A has the ability to automatically search these two bands. AT commands can also be used to switch between frequency bands. The AT command can be used to set the baud rate between 1200-115200. The GSM/GPRS Modem contains an integrated TCP/IP stack, allowing the connection of the internet through GPRS. The SIM900A wireless module is compact and reliable. This is a complete GSM/GPRS module in SMT form, with a very powerful single-chip CPU based on the AMR926EJ-S core, allowing users to benefit from small dimensions and low-cost solutions.

Figure 3.6: SIM900A GSM GPRS Module

3.4 Bill of Materials

Components	Supplier	Per Unit (RM)	Total Quantity Required	Unit Cost (RM)
Components				
Node-MCU ESP32	Cytron Technologies	29.00	1	29.00
5V DC HC-SR04 Ultrasonic Sensor	Cytron Technologies	2.90	1	2.90
Buzzer 6-12V c/w Wire	Cytron Technologies	3.00	1	3.00
Male to Female Wire	-	-	-	-
Battery + battery holder set	Shopee	6.05	1	6.05
Breadboard	-	-	-	-
Hardware				
UPVC Pipe 8" (1ft)	Shopee	16.50	1	16.50
ABBAWARE ACME Transparent Storage Box with Wheels 80L (63cm x 47 cm x 40cm)	Shopee	45.90	1	45.90
WIREMAN Weatherproof Enclosure Box IP56 (4" x 4" x 2")	Shopee	2.85	1	2.85
TOTAL				**RM 106.20**

Table 3.1: Bill of Materials

3.5 Software Implementation

3.5.1 Node-Red

In the prototype of this project, Node-Red is expected to send an alert message to the designated number. Node-RED is a programming tool for connecting physical devices, APIs, and web services in novel ways. It has a browser-based editor for easily connecting flows utilising the palette's vast range of nodes, which can be pushed to its runtime with a single click.

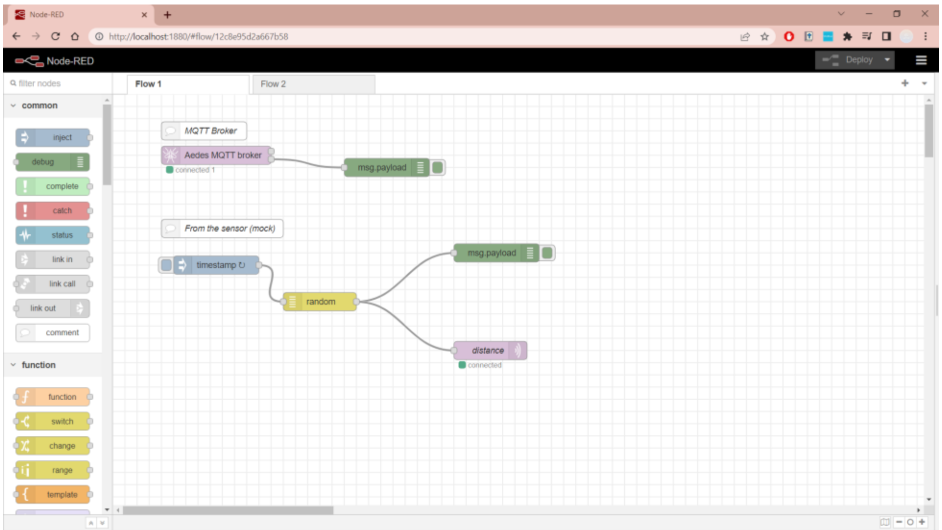


Figure 3.7: Node-Red Browser Based Editor (First Flow)

Figure 3.7 shows an example of the basic first flow of the water level detector. Aedes MQTT Broker is being used in this project to allow MQTT clients to interact with one another. An MQTT broker, in particular, takes messages published by clients, filters them by subject, and distributes them to subscribers as which the subscriber in particular is in the second flow of the Node-Red Editor.

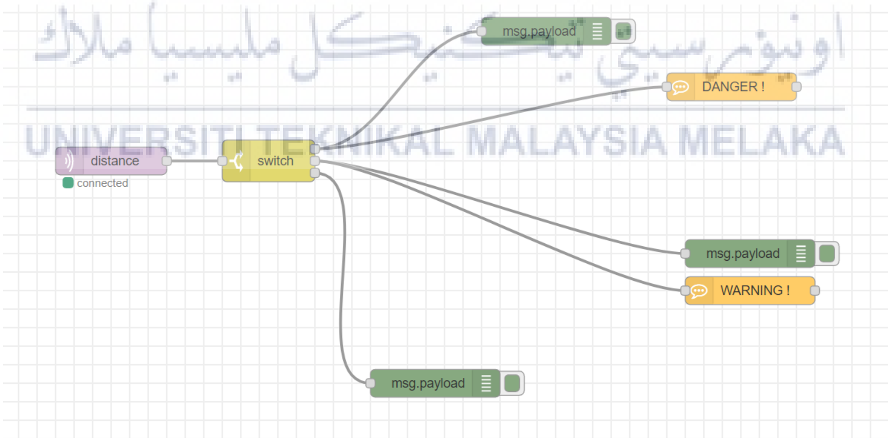


Figure 3.8: Node-Red Browser Based Editor (Second Flow)

Meanwhile, in the second flow of Node-Red as shown in Figure 3.8, it will include the sending messages nodes and display the gained data on the dashboard nodes. The system will received a signal through receiving MQTT nodes and afterwards will show the data as shown in Figure 3.9 below.

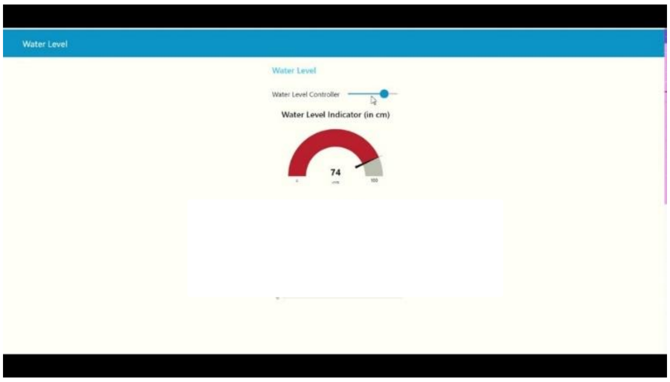


Figure 3.: Node-Red Dashboard

In the Figure 3.9 is the Node-Red based Dashboard. The main element of this project is to monitor the rises of the water level. In Node-Red, they provided a gauge button where it will display the percentage of the water level from its normal height.

3.5.2 Arduino IDE

The Arduino Integrated Development Environment (IDE), also known as the Arduino Software (IDE), has a code editor, a message area, a text terminal, a toolbar with buttons for basic functions, and a series of menus. It is able to interface with the Arduino hardware and upload applications to it as well. However, in order for Arduino IDE to be able to write code for ESP32, it was necessary to first install the appropriate library into the programme.

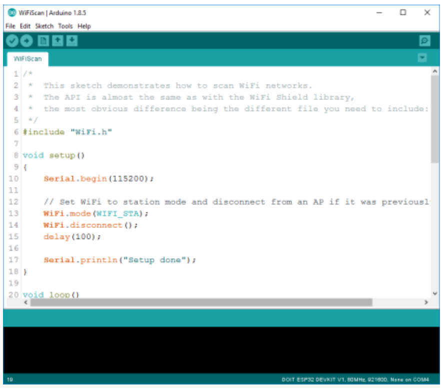


Figure 3.10: Arduino IDE

3.6 Project Design

3.6.1 Interfacing of Ultrasonic Sensor and ESP32

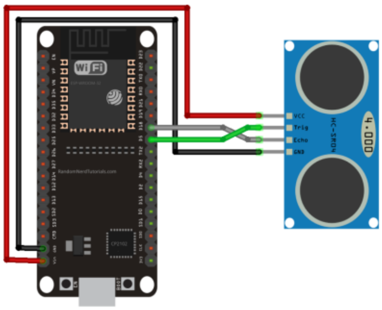


Figure 3.11: The Interfacing of Ultrasonic Sensor and ESP32

Connect the HC-SR04 ultrasonic sensor to the ESP32 as illustrated in the schematic the Figure 3.11 above. According to Table 3.2 below, connect the Trig and Echo pins to GPIO 5 and GPIO 18, respectively. While in the Arduino IDE, use C programming language to program the ESP32.

Pin of ESP32	**Pin of HC-SR04 Ultrasonic Sensor**
Vin	VCC
GND	GND
D5	Trig
D18	Echo

Table 3.2 : The Pin connection of ESP32 and the Ultrasonic Sensor

3.6.2 Interfacing of Buzzer and ESP32

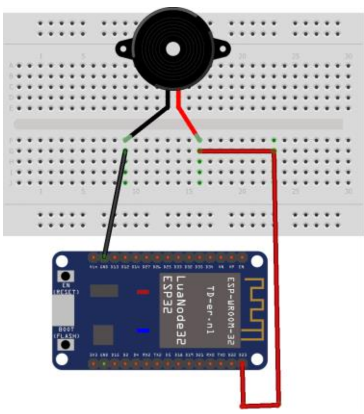


Figure 3.12: The Interfacing of Buzzer and ESP32

Based on Figure 3.12 shown above, the assembly consists of connecting the positive terminal of the buzzer to pin GPIO23 meanwhile the negative terminal to the ESP32 GND pin. In this project, the buzzer would act as the emergency alarm for the local residents to evacuate.

3.6.3 Interfacing of ESP32 and the GSM Module 900A

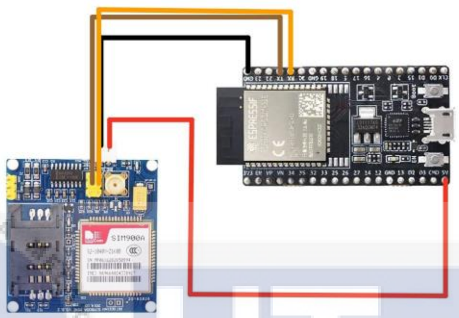


Figure 3.13: The Interfacing of GSM900A and ESP32

Based on figure 3.13 shown above is the circuit of the interfacing of the GSM Module900A and ESP32. The white connector where we connect 5V power supply on ESP32. The ESP32 Tx and Rx are connected in the middle of the male headers. The GSM Tx will be linked to the ESP32 Rx, while the GSM Rx will be connected to the ESP32 Tx.

3.6.4 Prototype Design

This project is built with a long UPVC pipe and the Ultrasonic Sensor will attach at the top of the pipe. The main components will be assembled inside a weatherproof enclosure box to avoid any damage during rain or overheating weather.

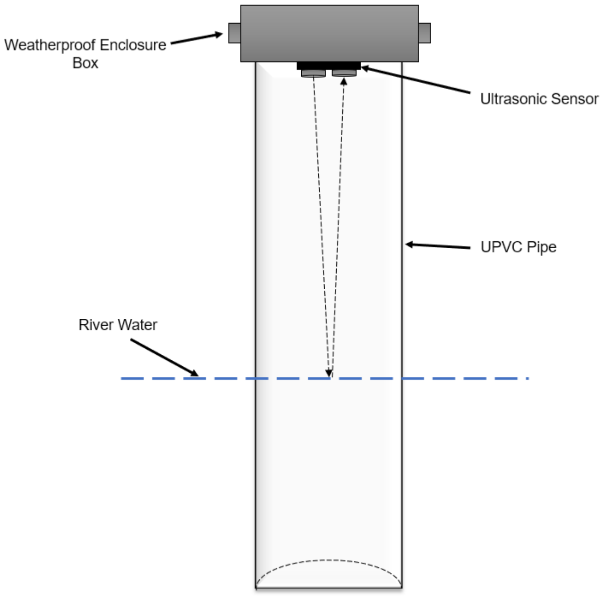


Figure 3.13: The prototype architecture of the project

3.7 Coding Explanation

‘PubSubClient.h’ for the MQTT protocol and ‘WiFi.h’ for ESP32 compatibility are the libraries that required to be downloaded. The code then begins with the definition of both Wifi credentials and MQTT server and protocol topics. The definition terms used in the coding are listed as in table below.

WiFi Credentials	
ssid	milaxx
password	milahaha
MQTT server and topics	
MQTT server name	iot.eclipse.org
Topic for MQTT	distanceCM

Table 3.3: The declaration code in the programming

The code was broken out into five voids which is void setup, void setup wifi, void reconnect, void sensorUS, and void loop. In void setup, is to declare the sensor and buzzer pinouts, as well as the MQTT protocol connection to Node-Red. The void setup wifi function is used to configure the ESP32's wifi. Then, void reconnect ensures that the system always reconnects to the established WiFi. The void sensorUS is responsible for configuring the sensor's data processing. Finally, the main void loop is where the computed data from the sensor is presented on the Node-Red dashboard, as well as the action that the buzzer and GSM will perform.

3.8 Summary of Chapter 3

A collection of procedures, techniques, processes, and rules that are used by people who work in a field is what is meant to be understood when one refers to the term "methodology." The purpose of project methodology is to exercise control over the whole of the management process by ensuring the efficacy of certain processes, approaches, techniques, methods, and technologies. This is accomplished via efficient decision making and the resolution of issues. In this Chapter 3, this project is implementing wireless-based system to alert residents and authorities and also to monitor the rises of the water level. This is a low cost project as the main components used are only ESP32, Ultrasonic Sensor and a buzzer.

CHAPTER 4
RESULT AND DISCUSSION

4.1 Introduction

This chapter will concentrate on the findings from the previous semester as well as the results from this semester. Furthermore, the performance and validation of the constructed system will be evaluated and explained in this chapter. The research is based on the construction of a water level monitoring system and the use of SMS alerts to notify warnings.

4.2 Hardware Development and Experimental Work

4.2.1 Circuit Design

Figure 4.1 below shows the circuit diagram and each component connect each other. The ultrasonic sensor and Buzzer connected to ESP32 microcontroller. All, ESP32, Ultrasonic sensor and Buzzer need at minimum 5V to generate data.

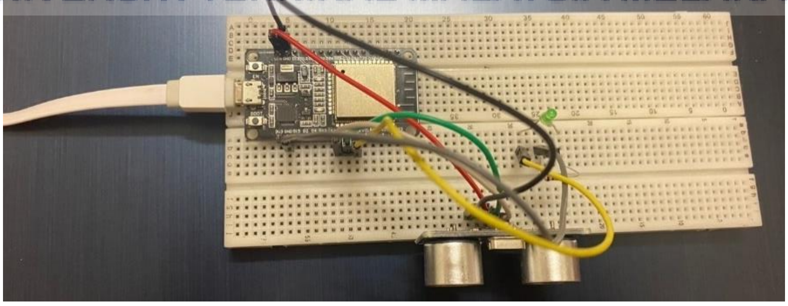


Figure 4.1: The circuit of project

An ultrasonic sensor was used to measure the depth of the water, and an alarm buzzer was used to notify the public. The system is controlled by an ESP32 microcontroller. All the gained data will be performed in chart and gauge on the Node-RED Dashboard ui platform. All the connection of the components are shown as in the List of Pinouts in the Table 4.1 below.

Components	Pinout	Components	Connected to
ESP32	GPIO19	Buzzer	Red
	GPIO18	Ultrasonic Sensor HCR-S04	TRIG
	GPIO5		ECHO
	VIN		VCC
	GND		GND
Ultrasonic Sensor HCR-S04	VCC	ESP32	VIN
	GND		GND
	TRIG		GPIO18
	ECHO		GPIO5
Buzzer	Red	ESP32	GPIO19
	Black		GND

Table 4.1 : The list of the pinouts and the connection of the component used

4.2.1 Prototype Design

The prototype design are using some appropriate tools such as water pipe PVC, tape, electronic container and transparent container. The Figure 4.2 below shows a set of prototype that has been installed.

Figure 4.2: Prototype Unit

For starters, the system uses ultrasonic sensors with a microcontroller board installed. Both of the components are installed in the electronic box and attached at the PVC pipe. The main task is the ultrasonic sensor to calculate the distance between the surfaces of the water and the ultrasonic sensor. For analysis testing, the attached PVC pipe will be placed in a transparent container. The maximum distance that could be measure is 30cm.

The gained data will be sent to the Node-Red system through the ESP32 before make the send SMS process. This configuration can communicate with each other. For example, if the Ultrasonic sensor detects the water level rising or the distance between the surface and the sensor is less than 5cm, the ESP32 will trigger the Node-Red to send an SMS notifications to notify the danger.

4.3 Software Development and Experimental Work

4.3.1 Node Flow Design in Node-Red

The key components of Node-Red are nodes and flow. The system flow is separated into two primary flows, the first of which includes the connection between the ESP32 and the Node-Red, as well as the parameters shown on the dashboard. Meanwhile, the second flow contains the node that conducts the SMS notification procedure.

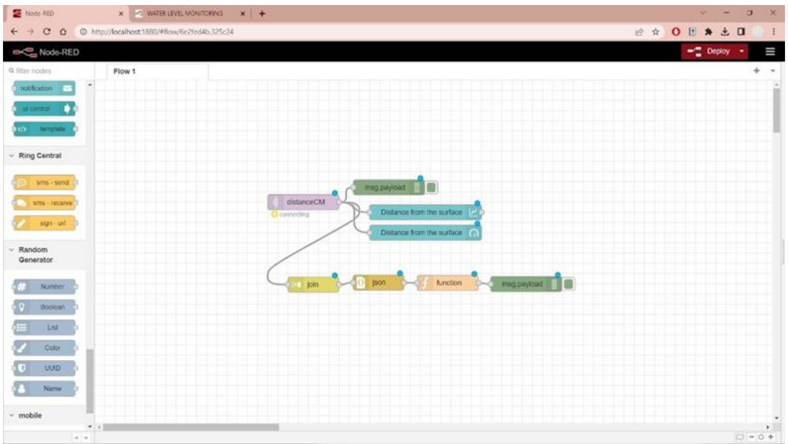


Figure 4.4: The Node Flow Design in Node-Red

The first main flow which as shown in the Figure 4.44, the “distanceCM” is the first node represents data receiver from the ESP32. Using a MQTT library installed in the Node-Red, the connection setup are shown as in Figure 4.5 below.

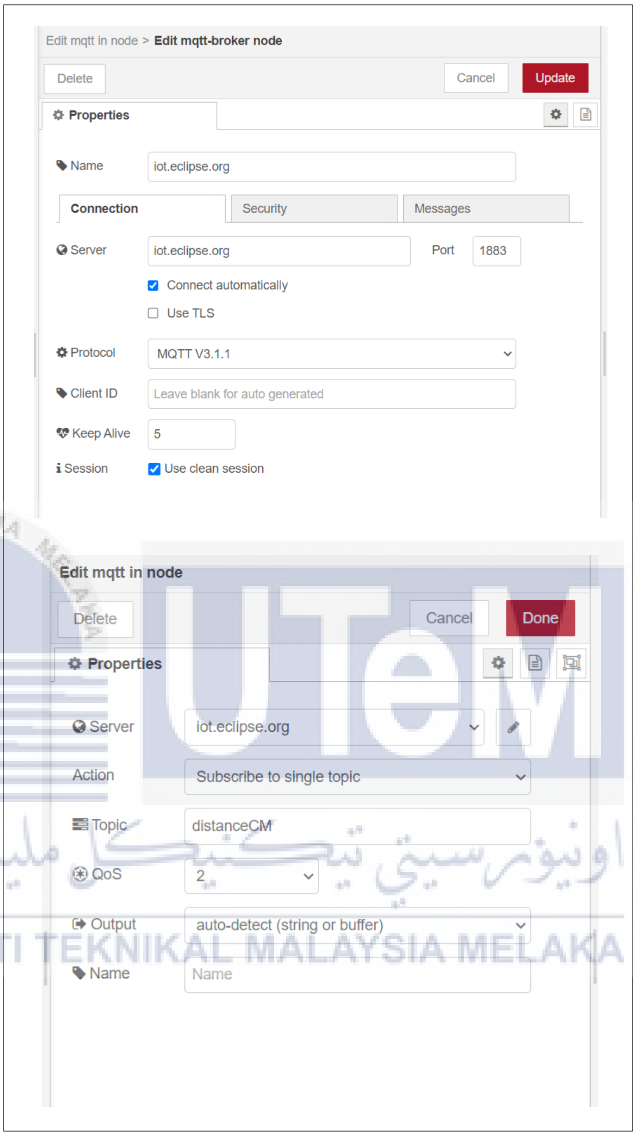


Figure 4.5: The parameters setup for MQTT node.

The server's name is "iot.eclipse.org," which is also defined in the project's programming. MQTT V3.1.1 is the protocol version that must be installed initially in Node-Red. To create a connection between the ESP32 and the Node-Red, the system topic, which is set to "distanceCM," must match. "distanceCM" is also specified in the ESP32's programming.

The second node is “json” which is a node to converts between JSON String and Object obtained from the ESP32. Meanwhile, the “function” node in this system is set to select an array as the message payload. The command is written in Javascript, as illustrated in Figure 4.6.

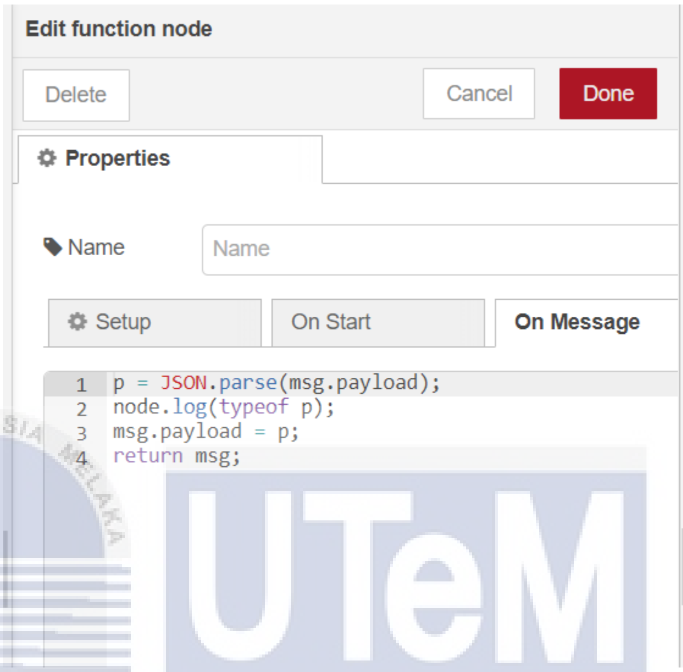


Figure 4.6: The command in the “function” node

As shown in the Figure 4.7 below, the second main flow is using a “switch” node to connected to the first main flow. The “switch” node allows messages to be routed to different branches of a flow by evaluating a set of rules against each message.

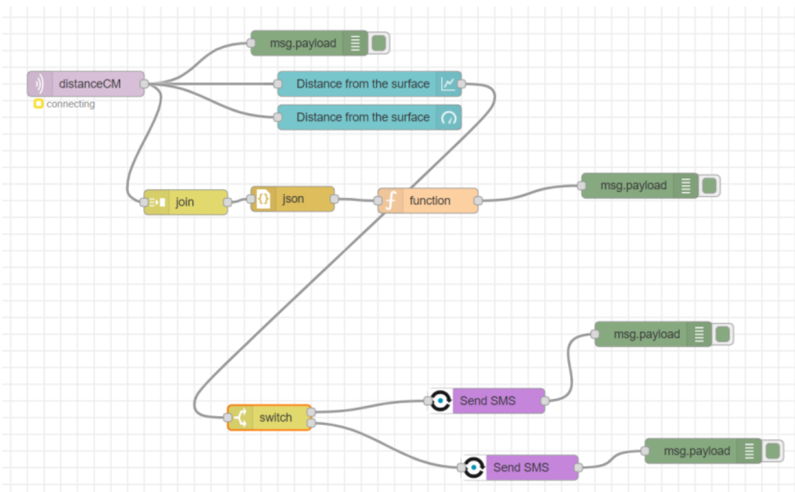


Figure 4.7: The second primary flow

The property to test is set on the node. The "switch" node is configured as shown in Figure 4.7. The node is given two paths, the first of which is triggered if the read data is between 0 and 6, and the second route is triggered if the read data is between 7 and 15.

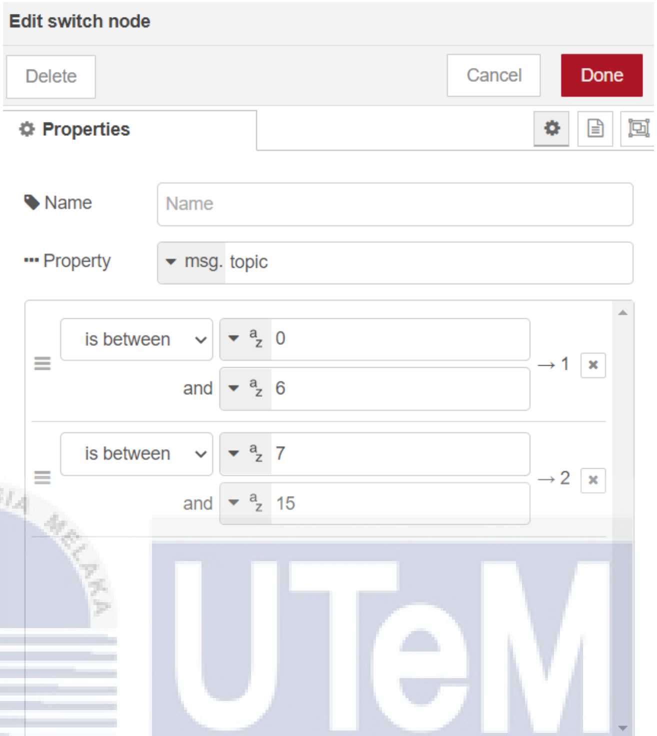


Figure 4.8: The configuration of the “switch” node

The Nexmo "Send SMS" node uses the API key issued by an API key provider to deliver SMS notifications. The first "Send SMS" node sends an evacuation SMS message, while the second node warns users that the water level is rising rapidly and they should prepare to evacuate. The configuration of this node is shown in Figure 4.9. The recipient number is set in E.164 format which is 60185711756. The SMS notification is received from the Vonage APIs.

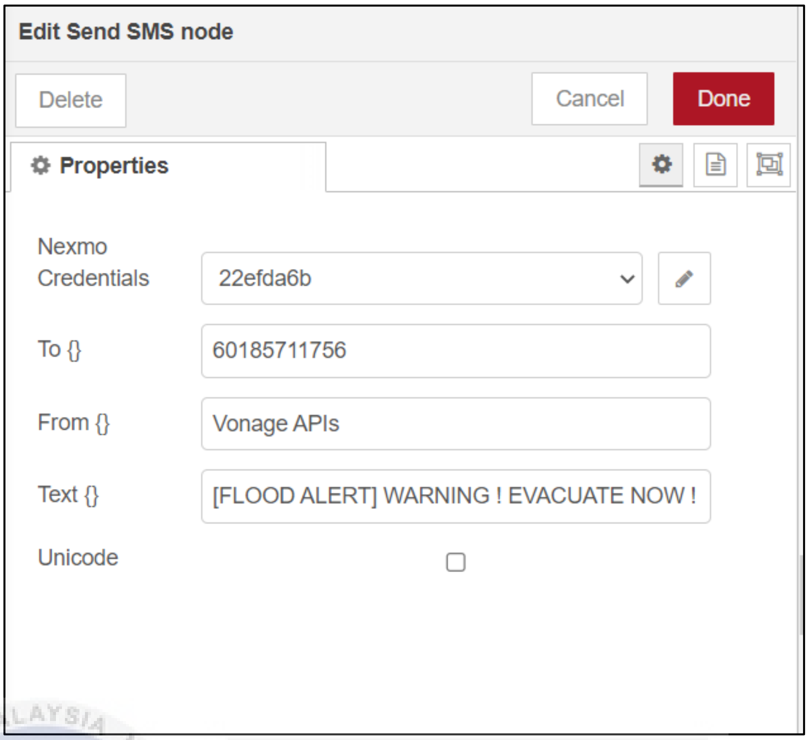


Figure 4.9: The configuration of “Send SMS” node.

4.3.2 Node-Red Dashboard

The dashboard's default tools for displaying data are a line chart and a level gauge as shown in Figure 4.10. The line chart can display the pattern of the water level, whilst the level gauge can display the value of the distance between the sensor and the water's surface.

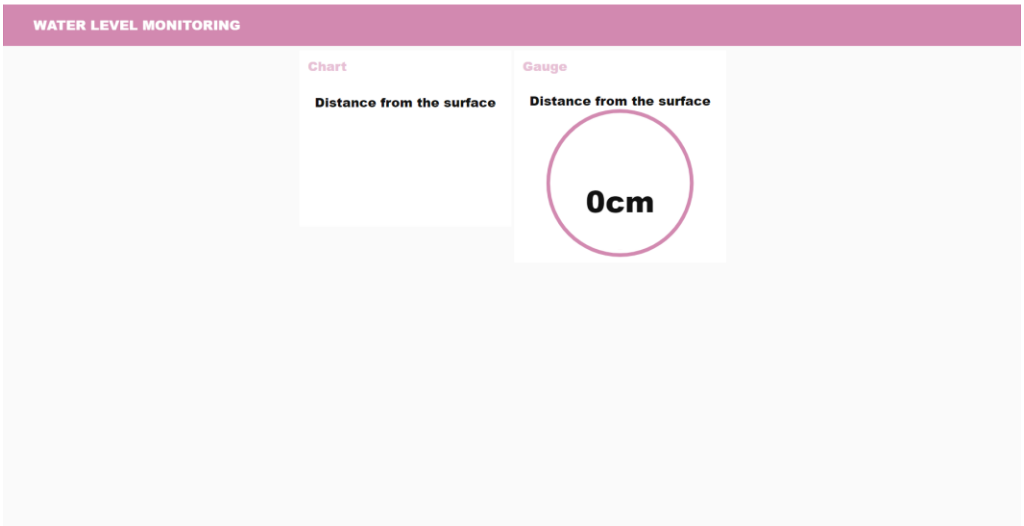


Figure 4.10: The Water Level Monitoring Dashboard

4.3.3 Setup SMS Provider using API Key

Both of the API Key and API Secret Key are provided by the website shown in Figure 10. The API Key is “22efda6b” while the API Secret Key is “EzPuKD8j623myeFt”. Vonage gave €2.00 free credit and each of the message cost €0.06 which this system roughly have 33 free tries.

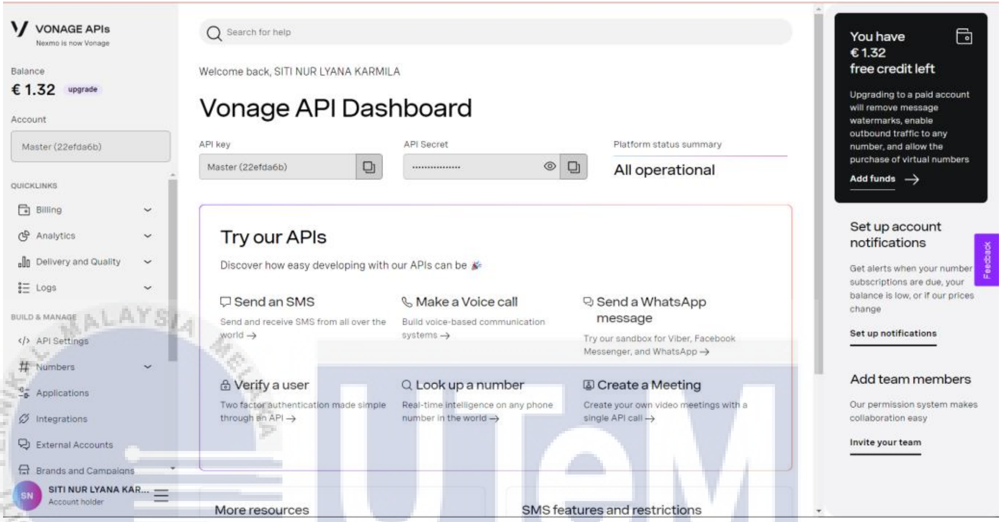


Figure 4.11: The website that provide API Key

4.4 Project Analysis

4.4.1 System Testing

The testing of the project begins by ensuring that Node-Red is started in the terminal by typing “node-red” in the command prompt of the computer after inject the power supply to the system. Then, the Node-RED Dashboard is available to access by pointing the default browser at http://localhost:1880/ui. As for the ultrasonic can detect the increase of the water in the river to make a measurement of the water level. When the water level rises, the ultrasound will sound an alert and send a signal. Typically, the ultrasonic can make a measurement which the available range of measurement for water is 10cm to 400cm. In this project, ultrasonic will communicate with the ESP32 to manage the river's water level.

However, for the time being, this project has only taken measurements ranging from 1cm to 30cm in order to produce a prototype. This research developed several algorithms to make ultrasonic work. The distance measured between the water and the ultrasonic sensor is shown in the table 4.2 below.

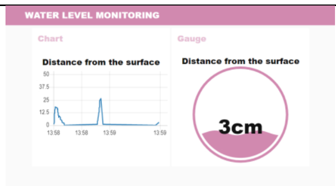
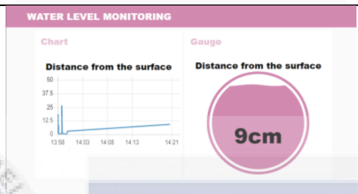
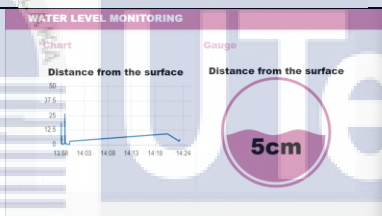
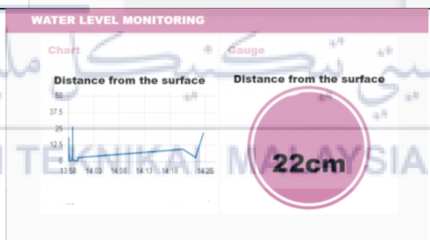
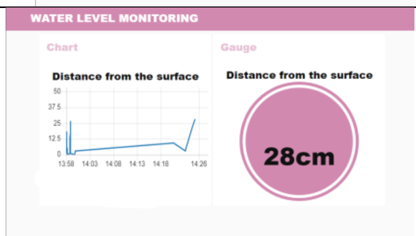
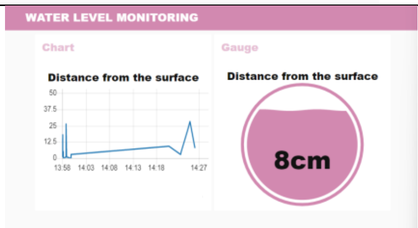
Level	**Distance**	**Alert**		**Action**
		SMS	**Buzzer**	
Safe	>15cm	No	No	No need for relocation
Mild Danger	Between 15cm to 6cm	Yes	No	Prepare for evacuation
Danger	<5cm	Yes	Yes	evacuate at an immediate rate

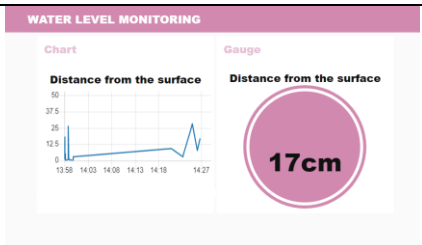
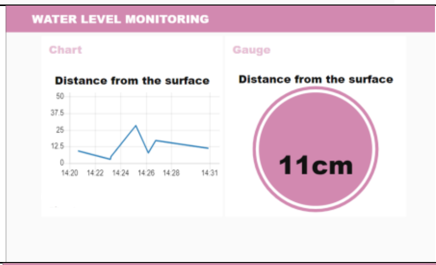
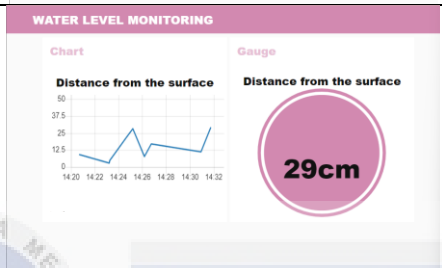
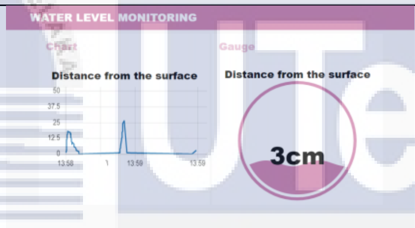
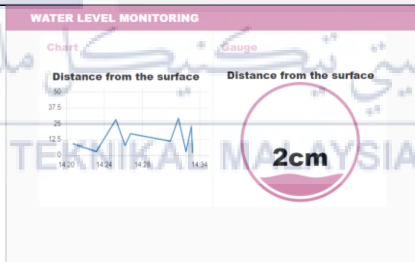
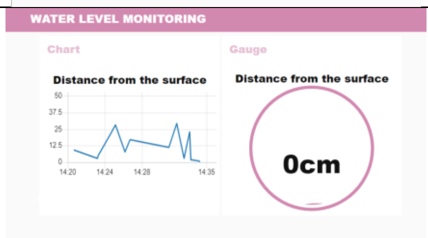
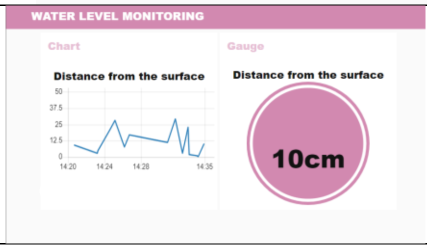
Table 4.2: The measurement of ultrasonic

From the Table 4.2 above, it can be able to concluded that when the water level is safe category, no action are to be taken. Meanwhile, when the water level in Mild Danger category, the SMS alert will be send to the recipient as in text of "[FLOOD WARNING] Water level is rising. Please be prepare for evacuation any time soon !". Lastly, when the water level is in Danger, buzzer will turn on and the SMS notification will be sent to the recipients as in text of "[FLOOD WARNING] EVACUATE NOW!"/

4.4.2 Result

Table 4.3 below is a result of direct observations of the prototype (manual testing) involving 20 trials. The obtained water level heights during the trials are range between of 26cm and 2cm. For the SMS alert system, the recipients phone received 8 messages alert.

No. of Trials	Distance (cm)	Alert		Dashboard	SMS
		Buzzer	SMS		
1	3	ON	YES		
2	9	OFF	YES		
3	5	ON	YES		
4	22	OFF	NO		-
5	28	OFF	NO		-
6	8	OFF	YES		

7	17	OFF	NO		-
8	11	OFF	YES		
9	29	OFF	OFF		-
10	3	ON	YES		
11	2	ON	YES		
12	0	ON	YES		
13	10	OFF	YES		

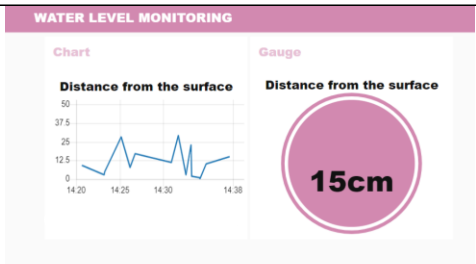
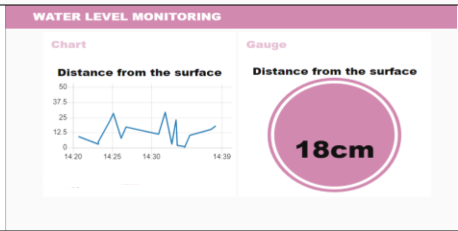
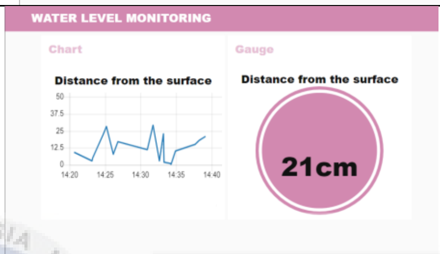
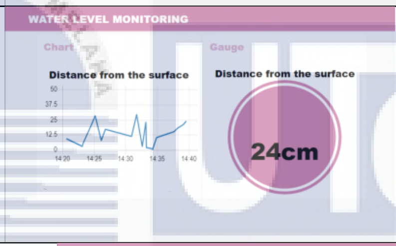
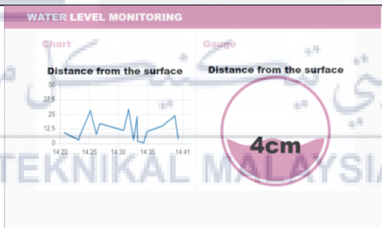
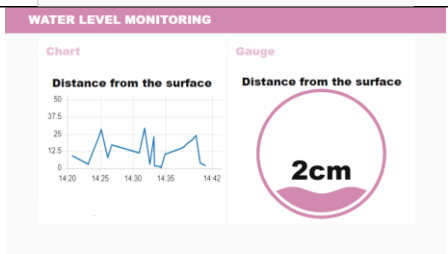
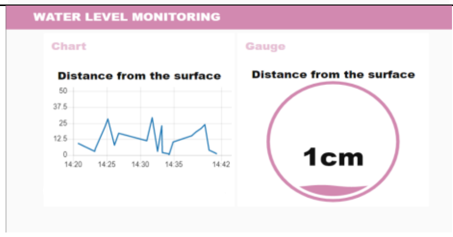
14	15	OFF	YES		
15	18	OFF	OFF		-
16	21	OFF	OFF		-
17	24	OFF	OFF		-
18	4	ON	YES		
19	2	ON	YES		
20	1	ON	YES		

Table 4.3 : The table of the trials

4.5 Discussion

The struggle during completing this project were most of the libraries downloaded in the Arduino IDE were not compatible with the MQTT protocol and Node-Red. After very thorough research, the library that compatible with both ESP32 and Node-Red is PubSubClient. Then, the connection of the ESP32 to the WiFi were not stable and keep reconnecting. And the solution for this issues was including a reconnecting void in the programming so then the ESP32 will automatically reconnecting to the WiFi connection without pressing any button on the board. Lastly, as the It was decided to discontinue the usage of GSM900A since the component requires an external 5v power supply because it appears to be impractical. Therefore, the send message method was replaced by subscribing to the API interface.

4.6 Chapter Summary

The study's conclusion is that the prototype developed can detect water levels and communicate information to a smartphone via a notification message or SMS. For the simulation data of 20 trials, the tests with varying water levels demonstrate that the application indications work effectively. As a result, this prototype can deliver accurate water level information.

CHAPTER 5
CONCLUSION

5.1 Conclusion

This paper explored the development of disaster control technologies in Malaysia, mostly for flooding disasters, in order to reduce property and life damage. In comparison to existing solutions, this concept of project can assist authorities in controlling the flood situation throughout Malaysia by utilizing a strong sensor network infrastructure with a stable. The intended design prototype was successfully created and tested in-house. Using Node-Red technology, it was able to record the water level and relay the data to the base station. In addition to recording data, Node-Red can deliver SMS alert notifications using the provided API platform.

5.2 Recommendation of future work

The number of sensor nodes can be increased in future investigations to construct a more comprehensive system and to obtain more exact data. To have additional flood measuring parameters, several types of sensors should be implemented. Aside from that, this project might employ a Google-based GPS system to determine the location of flood-prone areas. Furthermore, for long-term source consumption, a solar panel might be included to ensure a continuous power supply. Last but not least, because of this work is presently built only for a in-house testing, it is advised that on-site testing be performed to obtain more exact data.

APPENDIX

```
//Libraries
#include <PubSubClient.h>
#include <WiFi.h>

//WiFi Credentials
#define wifi_ssid "milaxx"
#define wifi_password "milahaha"

//Define MQTT server and topics
#define mqtt_server "iot.eclipse.org" //protocol server
#define Distancecm_topic "distanceCM" //topic

//define sound speed in cm/Us
#define SOUND_SPEED 0.034

//declare ultrasonic pinout
const int trigPin = 5;
const int echoPin = 18;

//port for buzzer
int buzzer = 19;
int frequnecy = 0;

//Rename WiFiClient to espClient and PubSubClient to client
WiFiClient espClient;
PubSubClient client;

volatile float distancecm;
long duration;

void setup ()
{
  Serial.begin(115200);//serial communication

  //set pinout for sensors
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin,INPUT);

  //define the buzzer pin as output
  pinMode(buzzer,OUTPUT);

  //for connection between MQTT and ESP32
  setup_wifi();
  client.setServer(mqtt_server,1883);
  client.setClient(espClient);
```

```
}

//WiFi Setup
void setup_wifi()
{
  delay(10);
  //Start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to");
  Serial.println(wifi_ssid);

  WiFi.begin(wifi_ssid, wifi_password);

  while(WiFi.status()!=WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

//Reconnecting
void reconnect()
{
  //loop until they reconnected
  while(!client.connected())
  {
    Serial.print("Attempting MQTT connection...");
    if(client.connect("ESP32Client"))
    {
      Serial.println("connected");
    }
    else{
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println("try again in 5 seconds");
      delay(500); //wait 5 seconds before reconnecting
    }
  }
}

//coding for Ultrasonic Sensor operation
void sensorUS ()
{
```

```
  //Clears the trigPin
  digitalWrite(trigPin,LOW);
  delay(200);

  //Sets the trigPin on HIGH state for 10Us
  digitalWrite(trigPin,HIGH);
  delay(100);
  digitalWrite(trigPin,LOW);

  //Calculation
  duration = pulseIn(echoPin,HIGH);
  distancecm = duration*SOUND_SPEED/2;

  //Display on Serial Monitor
  Serial.print("Distance (cm): ");
  Serial.println(distancecm);
  delay(1000);
}

//main coding
void loop()
{
  sensorUS();
  delay(100);

  if(!client.connected())
  {
    reconnect();
  }

  //Mentioned below directly executed in string url for node-red
  Serial.print("Distance : ");
  Serial.println(String(distancecm).c_str());
  client.publish(Distancecm_topic, String(distancecm).c_str(),true);
  client.loop();

  if(distancecm < 6)
  {
    digitalWrite(buzzer,HIGH);
    delay(100);
  }
  else
  {
    digitalWrite(buzzer,LOW);
  }

}
```

REFERENCES

Ab Halim, M. R. (2015). *Flood Monitoring and Warning System*.

Afnan Uda, M., Jambek, A., & Hashim, U. (n.d.). Smart IoT Flood Monitoring System You may also like Design 5.0 µm Gap Aluminium Interdigitated Electrode for Sensitive pH Detection. *Journal of Physics*. https://doi.org/10.1088/1742-6596/1339/1/012043

Jamali, A., & Giman, J. P. (2021). Performance Analysis of IOT based Flood Monitoring Framework in Sub-urban. *2021 IEEE 7th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*. https://doi.org/10.1109/icsima50015.2021.9526313

Satria, D., Yana, S., Munadi, R., & Syahreza, S. (2017). Prototype of Google Maps-Based Flood Monitoring System Using Arduino and GSM Module. *International Research Journal of Engineering and Technology*, *4*(10).

Sheeba Rani, D., Jayalakshmi, G. N., & Vishwanath, P. B. (2020). *Low Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks Flood Alerting and Rainfall prediction*. Proceedings of the Second International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2020).

Siek, M., & Larry, L. (2021). Design and Implementation of Internet of Things and Cloud Technology in Flood Risk Mitigation. *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*. https://doi.org/10.1109/icoris52787.2021.9649589

Syed Ahmed, A., Fasih, A., Ehsan, N., Usama, A., & Jehan, Z. (2018). *A Prototype for Flood Warning and Management System using Mobile Networks*. Proceedings of 2020 17th International Bhurban Conference on Applied Science & Technology (IBCAST).

Syed, A., Ali, F., Ashfaq, E., Nisar, U., & Azmat, J. (n.d.). *A Prototype for Flood Warning and Management System using Mobile Networks*.

Yassin, M. I. H. M., Wan, A. T., Newaz, S. H. S., & Omar, S. (2021). LoRa Based Real-time Flood Detection and Monitoring System: A Brunei Darussalam Based Study. *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*. https://doi.org/10.1109/icecit54077.2021.9641097

Shivashankar, Jijesh, J. J., Bolla, D. R., Penna, M., Sruthi, P. V., & Gowthami, A. (2020). Early Detection of Flood Monitoring and Alerting System to Save Human Lives. *2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*. https://doi.org/10.1109/rteict49044.2020.9315556

Yuliarman, S., Jandoni Horas, P. S., Hasna Aliya, R., & Agatha, E. S. (2020). Design of Automatic Water Flood Control and Monitoring Systems in Reservoirs Based on Internet of Things (IoT). *2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT)*.

Dario, J. (2016). Programming Visually with Node-RED: Wiring Up the Internet of Things with Ease. Retrieved June 9, 2022, from Toptal Engineering Blog website: https://www.toptal.com/nodejs/programming-visually-with-node-red

Jost, D. (2019, October 7). What is an Ultrasonic Sensor? Retrieved June 9, 2022, from Fierce Electronics website: https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor#:~:text=An%20ultrasonic%20sensor%20is%20an,sound%20that%20humans%20can%20hear).

What is the working principle of the buzzer? - Quisure. (2020). Retrieved June 9, 2022, from Quisure.com website: https://www.quisure.com/blog/faq/what-is-the-working-principle-of-the-buzzer

Buzzer : Working, Types, Circuit, Advantages & Disadvantages. (2021, July 26). Retrieved June 9, 2022, from ElProCus - Electronic Projects for Engineering Students website: https://www.elprocus.com/buzzer-working-applications/

ESP32 with HC-SR04 Ultrasonic Sensor with Arduino IDE | Random Nerd Tutorials. (2021, July 20). Retrieved June 9, 2022, from Random Nerd Tutorials website: https://randomnerdtutorials.com/esp32-hc-sr04-ultrasonic-arduino/

Overview of ESP32 features. What do they practically mean? (2022). Retrieved June 9, 2022, from Tutorials website: https://www.exploreembedded.com/wiki/Overview_of_ESP32_features._What_do_they_practically_mean%3F