

**DEVELOPMENT OF AN IOT-BASED SMART FLOOD ALARM
SYSTEM BY USING NODEMCU ESP8266 ESP12-E**



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

**DEVELOPMENT OF AN IOT-BASED SMART FLOOD ALARM
SYSTEM BY USING NODEMCU ESP8266 ESP12-E**



MUHAMMAD FAIZ BIN MOHD NASIR

**This report is submitted in partial fulfillment of the requirements for the
degree of Bachelor of Electronics Engineering Technology (Industrial
Electronics) with Honours**

**Faculty of Technology Electronic and Computer Engineering University
Technical Malaysia Melaka**

2025

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : DEVELOPMENT OF AN IOT-BASED SMART FLOOD
ALARM SYSTEM BY USING NodeMCU ESP8266 ESP12-E

Sesi Pengajian : 2024/2025

Saya MUHAMMAD FAIZ BIN MOHD NASIR mengaku membenarkan laporan Projek
Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran
antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐

SULIT*

(Mengandungi maklumat yang berdarjah
keselamatan atau kepentingan Malaysia
seperti yang termaktub di dalam AKTA
RAHSIA RASMI 1972)

☐

TERHAD*

(Mengandungi maklumat terhad yang
telah ditentukan oleh organisasi/badan di
mana penyelidikan dijalankan.

☐

TIDAK TERHAD

Disahkan oleh:

MUHAMMAD FAIZ BIN MOHD NASIR

Alamat Tetap: _____

SITI HARYANTI BINTI HJ HAIROL ANUAR
Pensyarah
Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK)
Universiti Teknikal Malaysia Melaka (UTeM)

Tarikh : 06/01/2025

Tarikh : 06/01/2025

DECLARATION

I declare that this project report entitled development of an IoT-based smart flood alarm system by using NodeMCU ESP8266 ESP12-E 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 :

Author : MUHAMMAD FAIZ BIN MOHD NASIR

Date : 06/01/2025

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 (Industrial Electronics) with Honours. The member of the supervisory is as follows:

Signature :

Supervisor Name : SITI HARYANTI BINTI HJ HAIROL ANUAR

Date 21/1/2025

DEDICATION

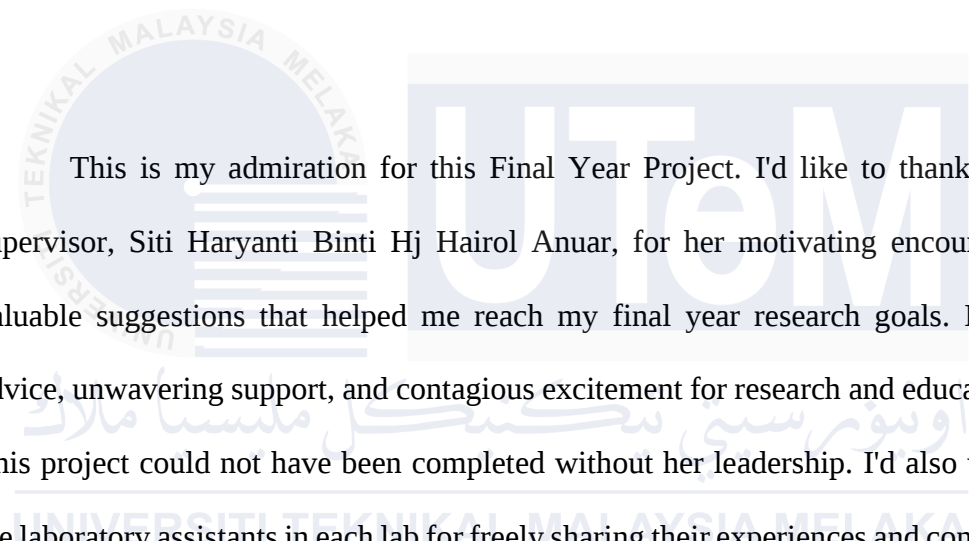
I dedicate this dedication to my beloved family, my father Mohd Nasir bin Mohd Nor, and my mother, Nik Maheran binti Nik Hassan, who never stopped praying and sending prayers for my success. Also, not forgotten my brothers and sisters who never stopped giving advice and being loyal supporters of this project in addition, I would like to thank my supervisor, Puan Siti Haryanti Binti Hj Hairol Anuar, who has provided guidance and advice to complete this project, Bachelor's Project2 (PSM 2).

Finally, to all my dear friends who have helped and are always with me in any situation. May all your good deeds be rewarded with kindness. Thank you so much for all your support and help.

ABSTRAK

Banjir merupakan salah satu bencana alam yang pernah berlaku di negara kita. Sebelum ini, kebanyakan sistem sedia ada yang dibangunkan hanya tertumpu kepada bidang tertentu. Oleh itu, sistem amaran banjir diperlukan untuk memberi amaran kepada penduduk tentang paras air terutama pada musim tengkujuh. Objektif kajian ini adalah untuk membangunkan sistem yang mengukur paras air banjir dan purata hujan harian menggunakan sensor ultrasonik dan tolok hujan tipping. Sensor ultrasonik akan mengukur tiga tahap, normal, amaran, dan bahaya. Tolok hujan tipping digunakan untuk mengira purata hujan harian. Penderia ini akan menghantar data kepada mikropengawal, yang disambungkan terus kepada ThingSpeak untuk menghantar maklumat ke telefon pengguna. Dengan sistem ini, pengguna akan menerima pesanan ringkas melalui aplikasi ThingSpeak, memberitahu mereka tentang paras air banjir semasa. Selain itu, ESP32-CAM dipasang untuk menyediakan rakaman video masa nyata paras air semasa, membolehkan pemantauan yang lebih cekap. Kesimpulannya, sistem ini membolehkan penduduk yang tinggal di kawasan rendah atau bersebelahan sungai mengambil langkah berjaga-jaga sekiranya berlaku banjir di kawasan mereka.

ACKNOWLEDGEMENTS



This is my admiration for this Final Year Project. I'd like to thank my fantastic supervisor, Siti Haryanti Binti Hj Hairol Anuar, for her motivating encouragement and valuable suggestions that helped me reach my final year research goals. Her continual advice, unwavering support, and contagious excitement for research and education are vital. This project could not have been completed without her leadership. I'd also want to thank the laboratory assistants in each lab for freely sharing their experiences and contributing vital insights to the research. Their unwavering support, readiness to answer inquiries, and great contributions were critical to the success of our project.

I would especially like to thank my parents for their constant support and spiritual encouragement. Their assistance has been beneficial in all aspects of my academic and personal development. I'm also grateful to my friends, whose encouragement and suggestions, either directly or indirectly, helped me finish this project. Their encouragement, sharing of ideas, and unwavering attitude kept me going till I succeeded. Finally, I'd want to convey my heartfelt appreciation to everyone who assisted me during this academic adventure and project, which was successfully finished.

TABLE OF CONTENTS

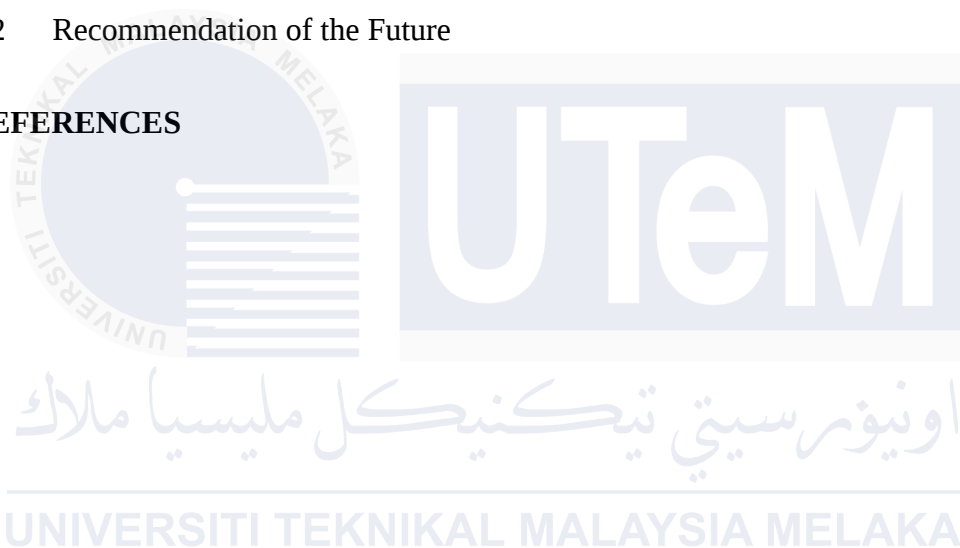
Declaration	
Approval	
Dedication	
Abstrak	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	ix
List of Tables	xii
List of Symbols and Abbreviations	xiii
Appendix	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Project Background	1
1.2 Problem Statement	4
1.3 Objective	4
1.4 Scope of Project	5

1.5	Research Questions	5
1.6	Significance of Study	6
CHAPTER 2 LITERATURE REVIEW		7
2.1	Introduction	7
2.2	Past-Related Research	7
2.2.1	Flood Monitoring and Warning System	8
2.2.2	Flood Level Control and Management Using Instrumentation Control	9
2.2.3	Early Flood Detection and Environment Monitoring System	10
2.2.4	Design Implementation of Flood Early Warning System for Residential Monitoring.	11
2.2.5	Enhancement of IoT-based Flood Detection and Prevention using Arduino UNO with Wi-Fi Module.	12
2.2.6	Smart Flood Monitoring System via IoT Platform for Early Warning of Road Closures Against Flood Events.	13
2.2.7	Smart IoT Flood Monitoring System	14
2.2.8	Flood Monitoring and Warning System with IoT	14
2.2.9	The Development of Smart Flood Monitoring System using Ultrasonic sensor with Blynk Application.	15
2.2.10	Internet of Thing-Based Smart Flood Forecasting and Early Warning System.	16
2.2.11	Smart Early Flood Monitoring System Using IoT.	17

2.2.12 Solar Fed Flood Alert System Using Arduino.	18
2.2.13 Water Level Monitoring and Flood Alerting by Applying IoT.	19
2.2.14 A Prototype for Flood Warning and Management System Using Mobile Networks.	20
2.2.15 A Water Level Detection IoT Platform Based on Wireless Sensor Network.	21
2.3 Research Gaps	21
2.3.1 Current Research and Technology	22
2.4 Comparisons of Existing Techniques of Smart Flood Detection	23
CHAPTER 3 METHODOLOGY	30
3.1 Introduction	30
3.2 Flowchart for Project Development	31
3.3 Block Diagram for Project Development	33
3.4 Project Hardware Selection	36
3.4.1 NodeMCU ESP8266 ESP12-E	37
3.4.2 Ultrasonic Sensor	39
3.4.3 LED (Traffic Light LED Module)	43
3.4.4 Buzzer	44
3.4.5 Hall Magnetic Sensor	46
3.4.6 LCD I2C	48
3.4.7 Tipping Rain Gauge	50

3.4.8	Servo Motor	52
3.4.9	Voltage Regulator	54
3.4.10	ESP32 CAM	55
3.4.11	Acrylic Perspex	55
3.4.12	Lithium Battery 3.7v	57
3.5	Project Software Selection	58
3.5.1	Arduino IDE	58
3.5.2	ThingSpeak	59
3.5.3	Web Server	61
3.5.4	Fritzing	62
CHAPTER 4 RESULTS AND DISCUSSION		63
4.1	Introduction	63
4.2	Software Part	64
4.2.1	Coding for ESP12e	64
4.2.2	Coding for ESP32-CAM	65
4.3	Hardware Part for ESP12e	76
4.4	Hardware Part for ESP32-CAM	76
4.5	Result for ESP12e	77
4.5.1	Water Level	78

4.5.2	Average of rain	81
4.6	Result for ESP32-CAM	83
4.7	Prototype	84
CHAPTER 5 CONCLUSION AND FUTURE WORKS		86
5.1	Conclusion	86
5.2	Recommendation of the Future	87
REFERENCES		88



LIST OF FIGURES

Figure 1.1 Statistics Of The Distribution Of Average Daily Rainfall [1]	2
Figure 1.2 Flood Level [4].	3
Figure 1.3 Rain Intensity [5].	3
Figure 2.1 Block Diagram of Flood Monitoring and Warning System [9].	9
Figure 2.2 Block Diagram of Flood Level Control and Management [10].	9
Figure 2.3 Block Diagram of Early Flood Detection [11].	10
Figure 2.4 Block Diagram of Flood Early Warning System [12].	11
Figure 2.5 Block Diagram of IoT-Based Flood Detection [13].	12
Figure 2.6 Block Diagram of Smart Flood Monitoring System Via IoT Platform [14].	13
Figure 2.7 Block Diagram of smart IoT flood monitoring system [15].	14
Figure 2.8 Block Diagram Of Flood Monitoring and Warning [16].	15
Figure 2.9 Block Diagram of Smart Flood Monitoring System [17].	16
Figure 2.10 Block Diagram of Flood Forecasting and Early Warning [18].	17
Figure 2.11 Block Diagram of The Smart Early Flood Monitoring System [19].	18
Figure 2.12 Block Diagram of Proposed Flood Alert System [20].	19
Figure 2.13 Block Diagram of Water Level Monitoring & Flood Alerting [21].	20
Figure 2.14 Block Diagram of Flood Warning and Management System [22].	20

Figure 2.15 Block Diagram of A Water Level Detection [23].	21
Figure 3.1 Flowchart for Smart Flood Alarm System	32
Figure 3.2 Block Diagram	34
Figure 3.3 Connection ESP32-CAM	35
Figure 3.4 NodeMCU ESP8266 ESP12-E	38
Figure 3.5 Ultrasonic Sensor HC-SR04 [32].	41
Figure 3.6 Ultrasonic Works [31].	42
Figure 3.7 LED Traffic Light [34].	44
Figure 3.8 Buzzer 5v	45
Figure 3.9 KY-024 Hall Magnetic Sensor [36].	47
Figure 3.10 LCD I2C [37].	49
Figure 3.11 Part of The Tipping Rain Gauge [8].	50
Figure 3.12 The Top of The Tipping Rain Gauge [8].	50
Figure 3.13 The Middle Part of The Tipping Rain Gauge [8].	51
Figure 3.14 The Bottom of The Tipping Rain Gauge [8].	51
Figure 3.15 Servo Motor SG90	53
Figure 3.16 Voltage Regulator LM2596	54
Figure 3.17 ESP32 CAM	55
Figure 3.18 Acrylic Perspex	56
Figure 3.19 Lithium Battery	57
Figure 3.20 Arduino IDE [38].	58
Figure 3.21 ThingSpeak	60
Figure 3.22 Function of ThingSpeak	60

Figure 3.23 Web Server	61
Figure 3.24 Fritzing	62
Figure 4.1 Circuit for Smart Flood Alarm System	76
Figure 4.2 Circuit for ESP32-CAM Monitoring	77
Figure 4.3 Result 1 for Water Level	78
Figure 4.4 Result 2 for Water Level	79
Figure 4.5 Result 3 for Water Level	80
Figure 4.6 Graph for Water Level	81
Figure 4.7 Result 1 for Rainfall	81
Figure 4.8 Result 2 for Rainfall	82
Figure 4.9 Result 3 for Rainfall	82
Figure 4.10 ESP32-CAM Monitoring	83
Figure 4.11 Prototype	84

LIST OF TABLES

Table 2.1 Comparison of journal	24
Table 3.1 List of Component	36
Table 3.2 Comparison of micro controller	37
Table 3.3 Comparison of ultrasonic sensor	39
Table 3.4 Comparison of LED	43
Table 3.5 Comparison of Buzzer	45
Table 3.6 Comparison of Hall Magnetic Sensor	46
Table 3.7 Comparison of LCD	48
Table 3.8 Comparison of Servo	53
Table 3.9 Comparison of Voltage Regulator	54
Table 3.10 Comparison of Material	55
Table 3.11 Comparison of Batteries	57

LIST OF SYMBOLS AND ABBREVIATIONS

For example:

μs : microseconds

s : seconds

m : *meter*

VCC : Voltage Common Collector

GND : Ground

LED : Light Emitting Diode

LCD : Liquid Crystal Display

PCB : Printed Circuit Board

DC : Direct Current

AC : Alternating Current

APPENDIX

Appendix A: Coding for ESP12e

66-71

Appendix B: Coding for ESP32-CAM

72-75



CHAPTER 1

INTRODUCTION

1.1 Project Background

Floods are among the most common and catastrophic disasters in the world, affecting millions of people every year. Floods and heavy rains are unavoidable occurrences that can result in loss of life and massive infrastructure damage. According to a report issued by (Jabatan Pengairan dan Saliran Malaysia, 2012) [1]. Figure 1.1 shows the average daily rainfall, the data explains that daily rainfall can increase the water level of the river. The increase in the river water level is measured using units of mm.



Figure 1.1 Statistics Of The Distribution Of Average Daily Rainfall [1]

Heavy rain in a few hours or days will cause flash floods. As shown in figure 1.1, the state of Terengganu received the most rainfall, 203mm. It may be due to the geography of the earth's surface which causes a lot of rain. Based on a report issued by the Department of Statistics, in 2020. Melaka is the smallest state in Malaysia with an area of 1,650 km² [2]. According to news issued by Astro Awani, 2023. Continuous heavy rain that continues up to 100mm per day is one of the causes of floods in Melaka, especially in Jasin [3]. Based on data released by the Department of Irrigation and Drainage Malaysia, 2024. The Durian Tunggal dam level is at the normal level which is 3.00m [4]. Figure 1.2 shows the flood level to determine the flood level is divided into 4 which are normal, alert, warning, and dangerous. The danger level is the river water starts to overflow and will cause flooding. The warning level means the river water is approaching the flood level. The alert level is the river water starting to rise. The normal level is the normal water level.



Figure 1.2 Flood Level [4].

There are several rain intensity categories of light rain 1-10mm, moderate rain 11-30mm, heavy rain 31-60mm and very heavy rain over 60mm. This reading is measured every hour. Figure 1.3 shows the rain intensity [5].

<u>Kategori Keamatan Hujan (dalam satu jam)</u>		
Renyai	1-10 (mm)	■
Sederhana	11-30 (mm)	■
Lebat	31-60 (mm)	■
Sangat Lebat	>60 (mm)	■

Figure 1.3 Rain Intensity [5].

The current chapter provides an example of the flood application along with its background or history, purpose, and some limitations. It explains what is contained in a range of services, features, and limitations. To increase the resilience level by IoT for early data collection for smart flood alert applications as the application was designed to popularize the concept of smart city and citizen engagement. Through such technology, one can detect and or forecast the occurrence of floods and prevent the event to reduce the effects.

1.2 Problem Statement

According to a report published by Adi et al, 2020. Flood disasters are among the most frequent disasters in Malaysia. A flood is a condition of water that submerges in a large area. Excessive rainfall is one of the causes. Also, the agencies involved may have a low level of preparation [6]. Lack of accurate and real-time water level detection. Current flood monitoring technologies are ineffective at correctly detecting increasing water levels and providing dependable early warning. Insufficient systems for measuring and analyzing rainfall. There is no effective mechanism for measuring rainfall quantities and calculating averages as part of flood assessments. Flood monitoring relies heavily on manual intervention. Current systems frequently rely on manual monitoring and data collecting, which limits efficiency and increases the likelihood of delayed reactions during emergencies. To prevent this from happening during the flood season, a "smart flood alarm system" device will be created to reduce unwanted accidents. The way to use the device is to place it in such a place as in low-lying areas that are often flooded or by rivers.

1.3 Objective

- a) To develop a flood detector that can detect the increased level of water and calculate the average amount of rain.
- b) To detect the current level of the flood where the data will be sent from NodeMCU ESP8266 ESP12-E to ThingSpeak Application.
- c) To analyze the performance of the development of an IoT-based smart flood alarm system.

1.4 Scope of Project

Research is underway on the benefits of using ultrasonic in smart flood alarm systems compared to other types of sensors. The HC-SR04 ultrasonic sensor has an accuracy limit range of 2cm to 400cm. This sensor is suitable for water level readings. The inability to detect certain materials such as soft surfaces or absorbents that might absorb or permeate ultrasonic waves, affects the sensor's ability to bounce back signals for measurement according to the journal issued by Raza and Monnet, 2019 [7]. The second sensor is the LM393 hall effect magnetic detector. Using a tipping rain gauge, where tipping rain gauge uses buckets to collect water. The amount of water accumulated in the bucket is calculated in ml units divided by the surface area of the rain catchment to get the average of rain. Magnetic sensors are placed in the bucket to calculate how many times the bucket pours water. One time the water is poured multiplied by the amount of water accumulated in the bucket [8]. This project uses only 2 sensors so that interruptions do not occur frequently, and the data is sent faster.

1.5 Research Questions

- a) What are the methods and technologies required to develop a flood detector that can detect an increased water level and calculate the average amount of water?
- b) How can the current level of a flood be accurately detected?
- c) How effective is an IoT-based smart flood alarm system detecting and alerting users about flood conditions, and what are the key performance metrics for such a system?

1.6 Significance of Study

This kind of research is highly valuable for refining the early experience and avoiding harm from floods. Thus, this developed smart flood warning system for IoT technology of NodeMCU ESP8266 ESP12-E hardware platform is capable of alarming the flood before entering the water more precisely and within less time than the other systems. In this system, a new palette of fresh sensors has been incorporated including ultrasonic sensors together with magnetic sensors, which allows us to pinpoint the changes in environmental parameters in real-time. This data is then passed to ThingSpeak where a much better forecast of the floods as per the diagram is made possible. Disaster information is well-timed; thus, it can be considered appropriate to help the inhabitants and the officials in terms of prevention. Such a kind of project can locate areas of interest and track them without establishing people there directly, which eradicates unnecessary loss of lives and property. It is coupled with mobile apps to allow an interchange of information in a way that assists in the cause of providing efficiency to a disaster response system.

CHAPTER 2



LITERATURE REVIEW

2.1 Introduction

This section focuses on research or selection of components, as well as system research to develop the project. Several related and comparable projects have been analyzed and distinguished in this chapter. The literature review section is an important component in showing the importance of this project and the expected results once it is completed.

2.2 Past-Related Research

The efficient implementation of the smart flood warning system project, a report on the reading materials and components about flood and flood control has been conducted.

These data and research materials are available in books, journals, newspapers, or websites that belong to a specific part. Of this piece of information, the project schedule becomes part of its specifications for successful completion on time. Therefore, the described projects can be used as reference solutions, as well as examples of optimal practices for enhancing various aspects of the smart systems for flood warnings, including the time for defining the levels, the frequency of the alert issuance, and the frequency of the data exchange in between the components of the smart system for flood warnings.

2.2.1 Flood Monitoring and Warning System

Referring to the report issued by Khan et al, 2020. The flood monitoring and warning system uses automatic sensors such as rain gauges, ram meters, and water level sensors to measure flood parameters [9]. This project uses an Arduino Uno-type ATMEGA238 microcontroller to collect data from sensors, calculate parameters like flow and water level, and trigger an alert via FM transmitter in flood situations. The Arduino ATMEGA238 is compatible with the system and easy to combine with other sensors, making it suitable for disaster applications. Figure 2.1 illustrates a flood monitoring system, powered by a battery, that collects, analyzes, and disseminates environmental data via wireless means, aiming for real-time environmental monitoring.

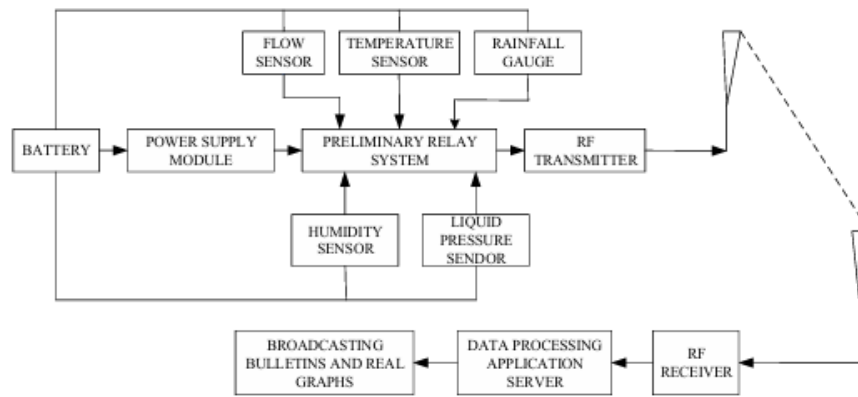


Figure 2.1 Block Diagram of Flood Monitoring and Warning System [9].

2.2.2 Flood Level Control and Management Using Instrumentation Control

According to the journal issued by Rajput et al, 2021. The flood control and management system uses instrumentation and control technology to monitor water levels in river catchment regions [10]. The project utilizes an Atmega328P-type microcontroller on an Arduino board for controlling water levels in model river systems, offering a cost-effective solution for integrating multiple sensors. Figure 2.2 shows the block diagram of flood level control and management shows the main components of the model. The PLC, ATmega328P, ultrasonic sensor, and LED screen work together to monitor water levels, provide early warnings, and control water flow for flood management.

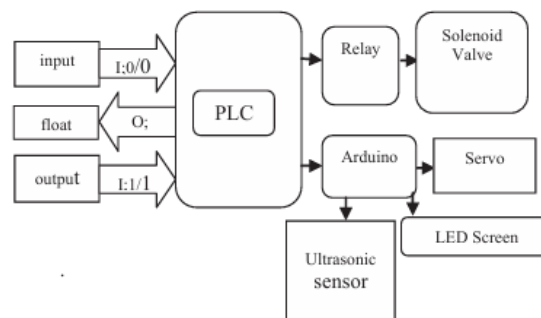


Figure 2.2 Block Diagram of Flood Level Control and Management [10].

2.2.3 Early Flood Detection and Environment Monitoring System

Referring to the report published by Bhuvana and Siva Shankar, 2022. This report used the ESP32 microcontroller to monitor a system that used IoT technologies for early flood detection and environmental monitoring [11]. The system uses sensors to gather data for flood prediction, monitor environmental elements, automate floodgate control, and reduce human interaction. The project uses an ESP32 Wi-Fi module for real-time data transmission from sensors to an IoT platform, enabling rapid monitoring of flood parameters and providing immediate alerts, ensuring reliable operation and extended battery life. Figure 2.3 shows an early flood detection system using ESP32 sensors. The system collects data, controls servo motors, and alerts LEDs and buzzers. It also sends data to the Blynk app for remote monitoring, ensuring environmental protection.

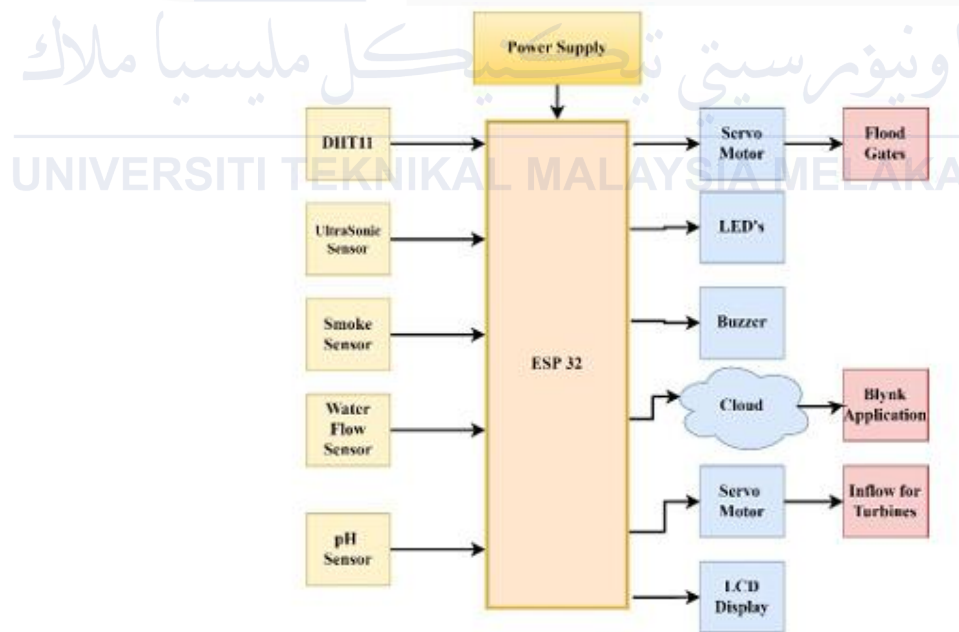


Figure 2.3 Block Diagram of Early Flood Detection [11].

2.2.4 Design Implementation of Flood Early Warning System for Residential Monitoring.

As per the report released by Simatupang and Arrazaq, 2023. The study article focuses on developing a Flood Early Warning System (FEWS) prototype for household monitoring to mitigate flood risk in Indonesia [12]. The system uses an Ultrasonic Sensor, Arduino Nano, NodeMCU, and HC-12 Wireless Serial Communication Module for accurate water level detection and flood early warning systems, displaying data online via the Blynk application. Figure 2.4 shows a block diagram of design implementation of flood early warning system showing a water level monitoring system that has two main units which are the transmitter unit and the receiver unit. A transmitter unit uses an ultrasonic sensor to measure water level, processed by a NodeMCU ESP8266 and sent to the Blynk cloud. The data is received by an HC-12 and sent to an I2C LCD, triggering alarms.

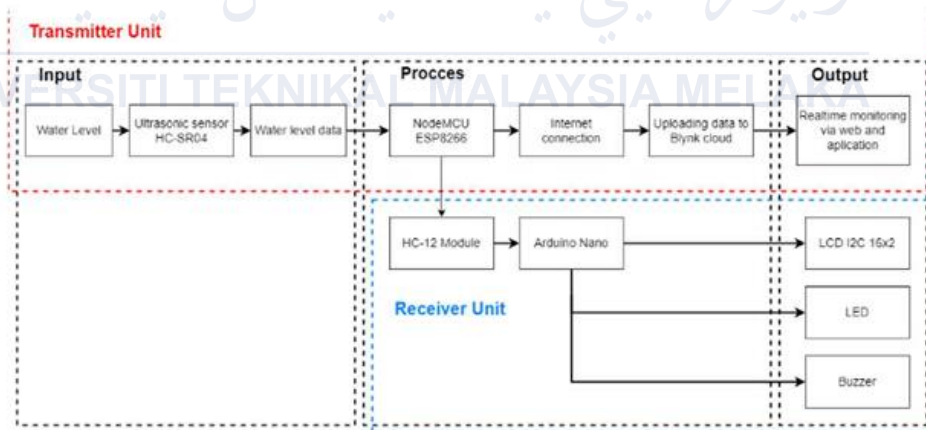


Figure 2.4 Block Diagram of Flood Early Warning System [12].

2.2.5 Enhancement of IoT-based Flood Detection and Prevention using Arduino UNO with Wi-Fi Module.

As indicated in the report published by Charaan et al, 2023. Enhancement of IoT-based flood detection has used the Wi-Fi Module to make this project an IoT project [13]. This project uses an Arduino Uno, humidity, flow, float, and ultrasonic sensors to predict and prevent floods. Data from these sensors is sent to the Bolt cloud for immediate alerts. The Arduino Uno is the main processing unit, facilitating real-time flood predictions and rapid prototyping for real-world applications. Figure 2.5 illustrates an IoT-based flood detection and prevention system using an Arduino-type UNO. The system receives water level and temperature data from sensors, displays it on an LCD screen, and controls a buzzer. The ESP8266 module sends data to a server for remote monitoring and control.

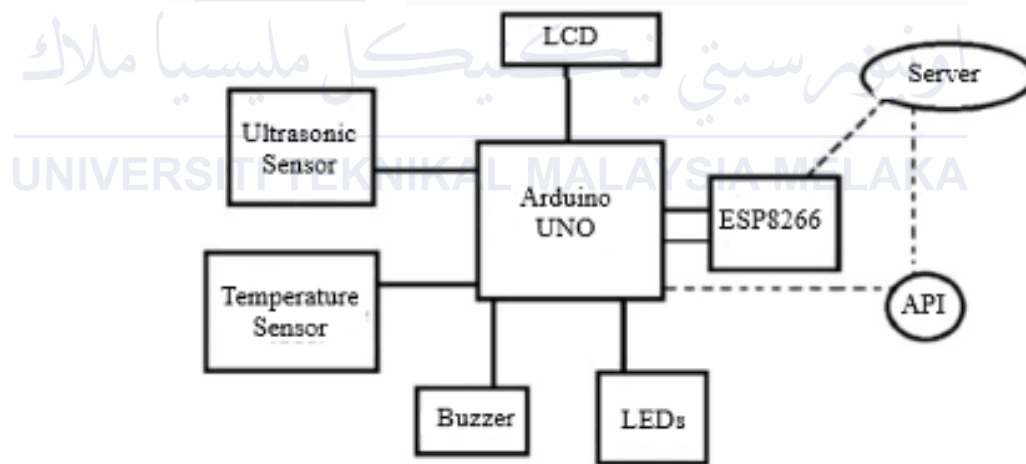


Figure 2.5 Block Diagram of IoT-Based Flood Detection [13].

2.2.6 Smart Flood Monitoring System via IoT Platform for Early Warning of Road Closures Against Flood Events.

As reported in the released document by Zainal and Ahmad, 2024. The Smart Flood Monitoring System utilizes IoT technology to provide early warnings of road closures during flood events [14]. The system alerts users about flooding-related road closures using an ultrasonic sensor. The Durian UNO controller sends water level data to the Blynk application, improving system efficiency and usability. The system integrates with ESP8266, offers wireless data transmission, and is user-friendly. Figure 2.6 shows the block diagram of smart flood monitoring. An ultrasonic sensor is used to calculate the water level and then the data will be sent to the microcontroller. The microcontroller is connected using the Blynk application. Data will be sent directly to the Blynk application and users can access Blynk just by entering their email to the Blynk application. Figure 2.6 shows the block diagram of smart flood monitoring.

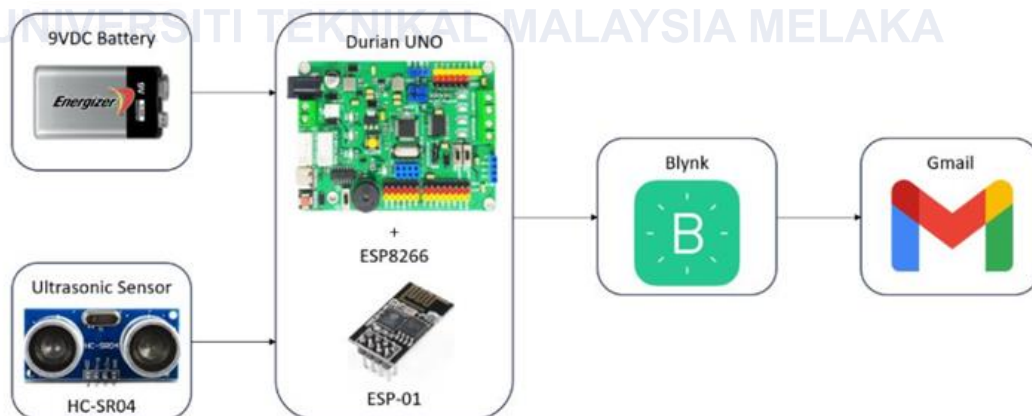


Figure 2.6 Block Diagram of Smart Flood Monitoring System Via IoT Platform [14].

2.2.7 Smart IoT Flood Monitoring System

Based on the journal published by Binti Zahir et al, 2019. The Smart IoT Flood Monitoring System operates by using an ARM Mbed NXP LPC1768 microcontroller to collect data from an ultrasonic sensor [15]. The ARM Mbed NXP LPC1768 is a suitable choice for high-performance, reliable, and speed-driven Smart IoT Flood Monitoring Systems. It receives input from an ultrasonic sensor and processes it to control various output devices, including a buzzer, LCD, LED, and stepper motor. The system also includes a Wi-Fi module for wireless communication and remote monitoring. Figure 2.7 illustrates the integration between sensors, processors, and various output devices for advanced control and monitoring applications.

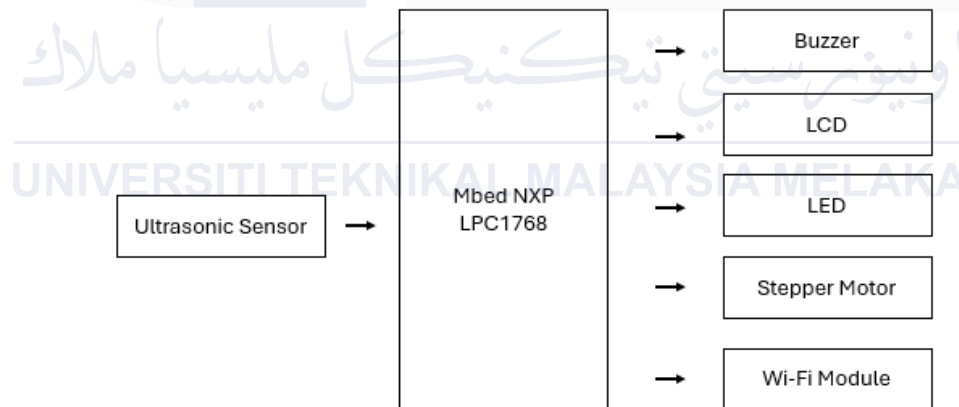


Figure 2.7 Block Diagram of smart IoT flood monitoring system [15].

2.2.8 Flood Monitoring and Warning System with IoT

As stated in the report published by Sufa, Yusof, and Sani, 2019. States that the Flood Monitoring and Warning System Project with IoT (FMWSI) uses technology (IoT) to improve the flood monitoring and warning system in Malaysia [16]. Figure 2.8 shows the block diagram of flood monitoring and warning shows the ultrasonic sensor connected to the

microcontroller to send the data. Microcontroller-type Arduino WeMosD1R1 will send the data to outputs such as LCD and LED. The LCD and LED will produce an output that shows the water level to inform the user. In addition, the data will be sent to the Blynk application to be displayed to users remotely. It is made as an IoT project.

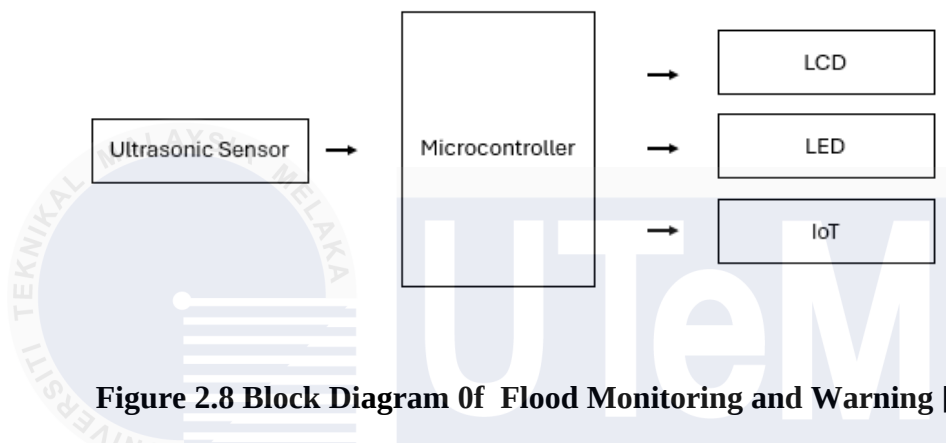


Figure 2.8 Block Diagram Of Flood Monitoring and Warning [16].

2.2.9 The Development of Smart Flood Monitoring System using Ultrasonic sensor with Blynk Application.

According to the documented report by Noar and Kamal,2017. The paper presents the development of a smart flood monitoring system using the Blynk platform for data transmission, integrating two NodeMCU [17]. The smart flood monitoring system utilizes two NodeMCU boards, one transmitting data wirelessly and the other triggering a buzzer and LED, enhancing system responsiveness during flood events. Figure 2.9 shows the block diagram development of a smart flood monitoring system showing where the first NodeMCU is connected from the Ultrasonic sensor. The first NodeMCU reads the data from the ultrasonic and the data will be sent to the Blynk application and display the data on the LCD. The second NodeMCU will read what is sent by the Blynk application and then activate the LED and Buzzer.

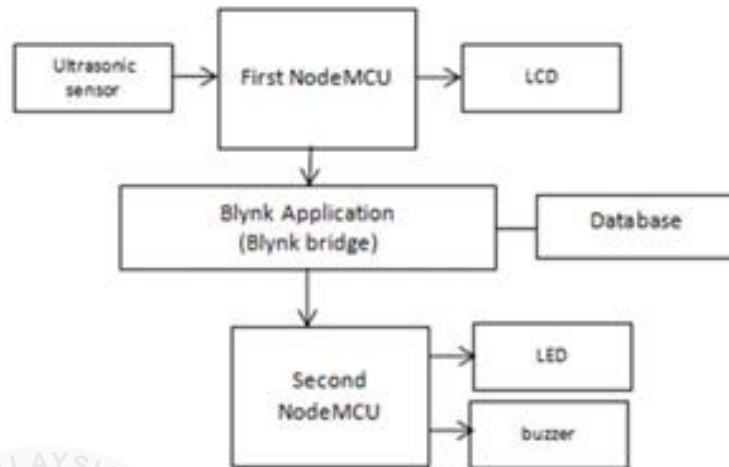


Figure 2.9 Block Diagram of Smart Flood Monitoring System [17].

2.2.10 Internet of Thing-Based Smart Flood Forecasting and Early Warning System.

Based on a report written by B. Shankar, T. John, S. Karthick et al. 2021. This project uses Internet of Things (IoT) architecture. The NodeMCU controller is the main component of the system, managing all activities. Sensors are responsible for detecting environmental characteristics such as temperature, humidity, snow melting, wind speed, rain, and air pressure [18]. The main goal of this initiative is to issue early flood warnings. This system aims to warn the community in the upstream and downstream areas about flash floods that will occur by continuously monitoring water levels, rainfall, and other related data. This early warning system is essential for immediate disaster response. Figure 2.10 shows the block diagram for the proposed method. In the modern world, there are extensive devices therefore using the BMP180 sensor that measures the temperature and humidity to differentiate between floods and let people know when there will be intense rain.

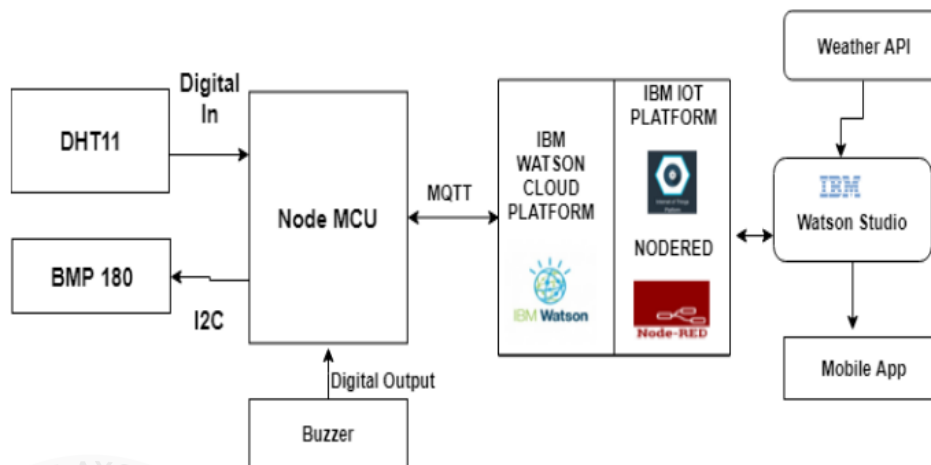


Figure 2.10 Block Diagram of Flood Forecasting and Early Warning [18].

2.2.11 Smart Early Flood Monitoring System Using IoT.

From the review conducted on MD. Ether Deowan, 2022. The project "Smart Early Flood Monitoring System Using IoT" uses IoT-enabled sensors, machine learning, and MATLAB to anticipate floods using time series forecasting [19]. They can observe these characteristics frequently, this may enable them to see flood characteristics before they appear thereby getting the communities out of large disasters. Figure 2.11 shows the block diagram of smart early flood monitoring system along with a detailed explanation of each part being discussed below. This system implements a NodeMCU ESP8266 microcontroller to handle the data transfer to ThingSpeak to indicate floods and predictions.

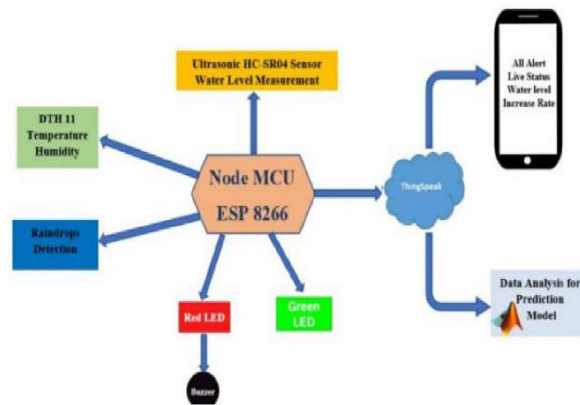


Figure 2.11 Block Diagram of The Smart Early Flood Monitoring System [19].

2.2.12 Solar Fed Flood Alert System Using Arduino.

The publications under study are the research article titled “Solar Fed Flood Alert System Using Arduino,” which was authored by M. Karthika; S. Paramasivam; D. Manivannan; and S. Karthik, in 2023. This article describes the design of a flood alarm system by integrating IoT technologies, Arduino, wireless communication, Bolt Wi-Fi, and solar panel. Sensors for floods and floods alerting systems belong to the system; they assist in offering timely notifications to prevent a disastrous situation [20]. Figure 2.12 shows the block diagram of the proposed flood alert system. The system aims to identify floods early, alert locals, and aid disaster management. It uses solar electricity, making it sustainable and reliable.

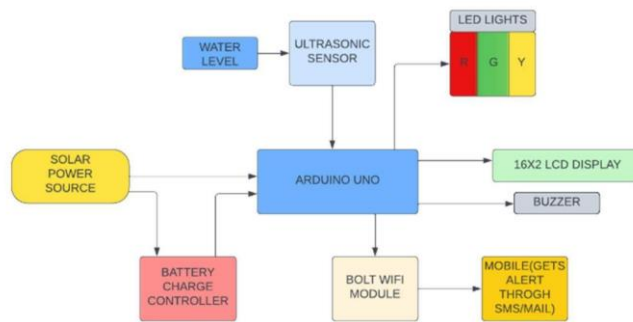


Figure 2.12 Block Diagram of Proposed Flood Alert System [20].

2.2.13 Water Level Monitoring and Flood Alerting by Applying IoT.

The report "Water Level Monitoring and Flood Alerting by Applying IoT" was authored by T. Opasworakun et al. 2024. Their work focuses on developing a robust and responsive flood warning system. The project created a flood warning system employing water level measurement devices outfitted with ultrasonic sensors and rain detection modules, all controlled by a microprocessor. Data was communicated using an IoT system that included ThingSpeak and Line apps, guaranteeing real-time data availability and effective communication [21]. Figure 2.13 shows the block diagram. The system uses microcontrollers and ultrasonic sensors to produce data, transmitted via ThingSpeak to a central control unit. Its primary function is to provide early flood alerts to locals, enabling swift response.

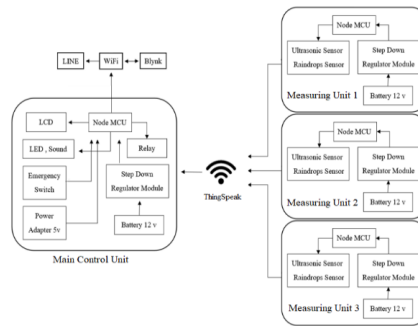


Figure 2.13 Block Diagram of Water Level Monitoring & Flood Alerting [21].

2.2.14 A Prototype for Flood Warning and Management System Using Mobile Networks.

The report titled "A Prototype for Flood Warning and Management System using Mobile Networks" is authored by S. Ali, F. Ashfaq, E. Nisar et al. 2019. Their work presents a sophisticated approach to flood warning and management leveraging mobile networks and IoT technology [22]. Figure 2.14 shows the block diagram of the system. The system uses an ESP32 board, sensors, and actuators to collect data for flood warnings and spillway management. MATLAB algorithms process the data, which is sent to an IoT platform, and an Android application enhances public response.

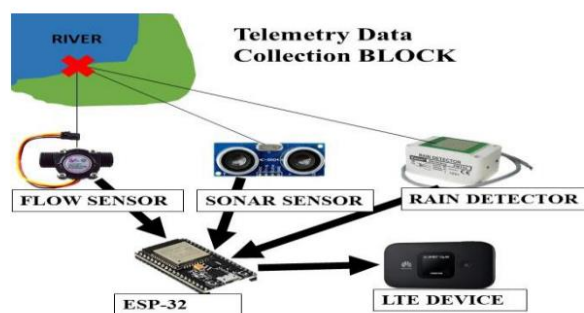


Figure 2.14 Block Diagram of Flood Warning and Management System [22].

2.2.15 A Water Level Detection IoT Platform Based on Wireless Sensor Network.

The research paper "A Water Level Detection: IoT Platform Based on Wireless Sensor Network" was authored by A. Prafanto and E. Budiman, in 2018. To monitor real-time water levels in flood-prone locations, the project uses a wireless sensor network using HC-SR04 proximity sensors and NodeMCU. This structure provides continual monitoring of water levels, which successfully reduces flood threats [23]. The technology uses ultrasonic sensors to detect water height and send results to a website for public monitoring, enabling real-time river level monitoring, and improving flood preparedness. Figure 2.15 shows the block diagram of a water level detection. The project uses HC-SR04 sensors and a NodeMCU for monitoring water levels in flood-prone areas.

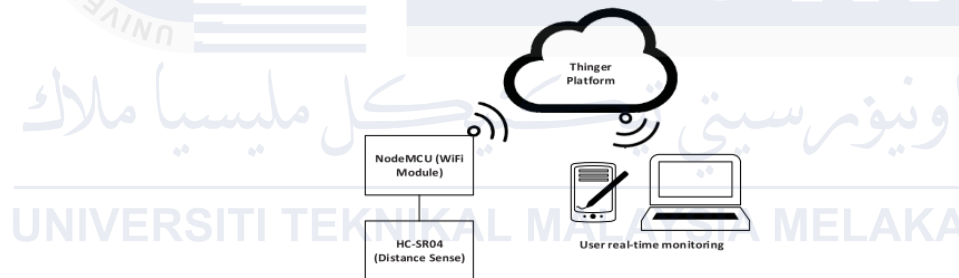


Figure 2.15 Block Diagram of A Water Level Detection [23].

2.3 Research Gaps

Research gaps are places where present information, expertise, or technology is insufficient or incomplete, showing the need for more inquiry to develop the subject. Identifying these gaps helps to guide future research efforts.

2.3.1 Current Research and Technology

According to Noar and Kamal, 2017, The study develops a smart flood monitoring system using ultrasonic sensors and Blynk apps, but suggests improvements in real-time monitoring, predictive analytics integration, scalability, energy efficiency optimization, and user interface refinement. According to the issued report by Rath et al. 2022, Current research lacks a comprehensive flood monitoring system, necessitating future improvements in algorithms, sensor upgrades, and communication protocols for improved flood alert systems.

According to Bhuvana and Siva Shankar, 2022, Current research lacks comprehensive environmental factors for flood detection, dam gate automation, and monitoring, necessitating advanced, accessible, and affordable early warning systems, particularly in underserved areas. Following the report issued by Charaan et al, 2023, Current flood detection research lacks multi-sensor integration, scalability, power efficiency, and reliable communication, necessitating further research for better sensors, larger coverage, sustainability, and real-time monitoring capabilities. As per the report released by Khan et al, 2020, The research focuses on developing cost-effective flood monitoring systems using sensor networks, with future opportunities to optimize sensors, consider solar power, and enhance data accuracy. The research aims to enhance flood warning systems (FEWS) mobility, accuracy, and cost-effectiveness through IoT technology. It proposes an IoT-based solution for real-time notifications, addressing shortcomings in manual flood monitoring. The Smart IoT Flood Monitoring System uses ARM Mbed IoT technology for data collection and distribution [16].

2.4 Comparisons of Existing Techniques of Smart Flood Detection

In the past journals have outlined several approaches for delivering high-quality projects, some of which include sound project management structures, incorporation of best technologies, integration of multi-disciplinary teams, and lastly effective quality checks. They help to be more in line with stakeholders' expectations, utilize new trends in software and hardware, focus on innovations, and look for efficient solutions. Table 2.1 shows the comparison of the comparison of the journals reveals that it is critical to approach project execution in some interrelated and complex ways and change management tactics should be tailored to the given project type, goals, and environment.



Table 2.1 Comparison of journal

REF	TITLE & AUTHOR	PROPOSED TECHNIQUE	ADVANTAGE	DISADVANTAGE
1. Ref [9]	Flood Monitoring and Warning System. (Khan et al, 2020)	The main goals of this project are to develop a system for measuring the level of water and the rate of water that will help people track the actual level of water.	• Enhance disaster preparedness through real-time monitoring and forecasting. • Cost-effective and easy to deploy, making it accessible for various regions.	• Rely on sensors that may require maintenance and calibration over time. • Limited by the reach of the sensor network, impacting coverage in remote areas.
2. Ref [10]	Flood Level Control and Management Using Instrumentation and Control. (Rajput et al, 2021)	Level monitoring and measurement systems, alarm systems, and the use of LEDs using which people can be warned and therefore help to reduce the effects of floods.	• Helps minimize the environmental impacts of flood hazards. • Reduces the risk of flood and channel migration hazards.	• Requires significant resources for implementation and maintenance.
3. Ref [11]	Early Flood Detection and Environment Monitoring System.	Multiple sensors such as temperature, humidity, water level, flow, and ultrasonics. An Android app with an intuitive GUI helps to display	• Early detection of floods helps in timely evacuation and reduces damage.	• System dependency on sensors and connectivity for accurate predictions. • False alarms triggered by sensor inaccuracies can lead to unnecessary panic.

	(Bhuvana and Siva Shankar, 2022)	all relevant data in a visual format.	Monitoring environmental parameters ensures the safety of aquatic life.	
4. Ref [12]	Design Implementation of Flood Early Warning System for Residential Monitoring. (Simatupang and Arrazaq, 2023)	Design and implement an affordable FEWS model for monitoring the performance of the designed prototype, based on the accuracy of sensor reading and prototype functionality.	- Accurate water level readings provided by the prototype system. - High consistency in sensor readings observed during testing.	- Lack of data logging features in the current prototype system. - Absence of renewable power source integration (like solar panels) for sustainability
5. Ref [13]	Enhancement of IoT-based Flood Detection and Prevention using Arduino Uno with Wi-Fi Module. (Charaan et al, 2023)	This system combines an Arduino Uno with a few sensors, including DHT-11, ESP8266 and others. This module produces a warning signal that is wirelessly transmitted to the appropriate authorities.	- Early flood detection through sensor integration. - Immediate alert system for nearby communities to evacuate promptly.	- Reliance on technology may lead to system failures during emergencies. - Limited effectiveness in areas with poor internet connectivity for data transmission

6. Ref [14]	Smart Flood Monitoring System via IoT Platform for Early Warning of Road Closures Against Flood Events. (Zainal and Ahmad, 2024)	uses IoT technologies to develop a smart flood monitoring system. This technology is intended to offer early notifications regarding road closures during flood disasters.	• Provides accurate water level measurements with 96.65% precision. • Enables authorities to promptly announce vehicle restrictions or road closures.	• Relies on electronic components that may have maintenance requirements. • Initial setup and implementation costs might be a factor.
7. Ref [15]	Smart IoT Flood Monitoring System. (Binti Zahir et al, 2019)	Smart IoT Flood Monitoring System warns the public before a flood occurs.	• Provides early warnings to residents to take preventive actions. • Helps in monitoring flood-prone areas efficiently.	• Initial setup and maintenance costs could be high. • Limited reach to areas without internet connectivity.
8. Ref [16]	Flood Monitoring and Warning System with IoT (Sufa, Yusof, and Sani,2019)	This system will help the residents to act before the floods get worse and prevent flood traps.	• Helps in monitoring flood-prone areas effectively. • Enables early warning notifications to residents before flooding	• Limited processing power and memory make possible limits for sophisticated activities.

9.Ref [17]	The Development of Smart Flood Monitoring System using Ultrasonic Sensor with Blynk Applications. (Noar and Kamal,2017)	Smart flood monitoring using ultrasonic sensor with Blynk application	• The Blynk application provides real-time data transmission for quick responses • Smart flood monitoring system enhances early detection of water levels	• Limited range of data transmission between NodeMCUs (less than 50 meters). • Relies on wireless connection which may be susceptible to interference.
10.Ref [18]	Internet of Thing-Based Smart Flood Forecasting and Early Warning System. (B. Shankar, T. John, S. Karthick et al. 2021)	The system monitors floods, and predicts events using IoT-based environmental monitoring	• Early detection of events like floods, fires, and gas/water leaks. • Provides alerts to prevent risks to human life and property.	• Reliance on technology may lead to system failures. • Initial setup costs for implementing the IoT infrastructure.
11. Ref [19]	Smart Early Flood Monitoring System Using IoT. (MD.	Water level and rainfall are monitored using ultrasonic and rain sensors, respectively. Data transferred to the	• Utilizes IoT technology for real-time data collection and monitoring.	• Dependency on technology may lead to system failures during disasters.

	Ether Deowan, 2022)	ThingSpeak server using IoT technology.	Enhances overall preparedness and response to flood situations.	
12. Ref [20]	Solar Fed Flood Alert System Using Arduino. (M. Karthika et al.2023)	Includes sensors, Arduino, wireless connectivity, IoT platforms, and Bolt Wi-Fi. Solar-powered for system efficiency and lower power use.	Solar-powered for system efficiency and lower power use	Limited errors during malfunctions
13. Ref [21]	Water Level Monitoring and Flood Alerting by Applying IoT. (T. Opasworakun et al. 2024)	Ultrasonic sensors are used for measuring water levels, and rain detection modules are integrated to monitor rainfall.	- Provides early flood warnings to enable prevention. - Allows residents to quickly assess situations and respond effectively.	Dependence on stable connectivity for IoT functionality
14. Ref [22]	A Prototype for Flood Warning and Management	The ESP32 microcontroller is utilized for data collecting and transmission.	- Efficient real-time monitoring of flood risk situations. - Utilizes ESP32 development board for data collection.	- Reliance on mobile networks may face connectivity issues. - Dependency on sensor units which can malfunction.

	System using Mobile Networks. (S. Ali, F. Ashfaq, E. Nisar et al. 2019)			
15. <i>Ref</i> [23]	A Water Level Detection: IoT Platform Based on Wireless Sensor Network. (A. Prafanto and E. Budiman, 2018)	Water level sensing is carried out using an HC-SR04 proximity sensor and a NodeMCU.	- Real-time monitoring of river water level for public safety. - Low power consumption for cost-effective continuous operation.	Limited to specific areas with implemented sensor networks

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodology for developing this IoT-based smart flood alarm system is divided into two main components: hardware and software development. The process involves a systematic approach, beginning with the selection of suitable components based on the project's requirements. The selected hardware components include the NodeMCU ESP8266 and ESP32-cam microcontroller, HC-SR04 ultrasonic sensor, tipping rain gauge, LEDs, buzzers, and an I2C LCD display. These components are chosen for their compatibility, cost-effectiveness, and reliability in providing accurate real-time data. The integration process involves interfacing the sensors with the microcontroller, where the ultrasonic sensor measures the water level while the tipping rain gauge calculates the rainfall intensity. This integration is crucial to ensure the sensors operate harmoniously, with accurate data

collection and seamless communication with the microcontroller. The NodeMCU ESP8266 acts as the central processing unit, receiving data from the sensors and processing it for analysis and decision-making. The processed data is displayed locally on the I2C LCD and transmitted to the ThingSpeak platform for remote monitoring and alerting. On the software side, the Arduino IDE is used for programming the NodeMCU ESP8266, ensuring the microcontroller can accurately interpret sensor data and perform logical operations. The programming also includes setting thresholds for the ultrasonic sensor to categorize water levels into safe, warning, and danger zones. Simultaneously, the tipping rain gauge's software calculates the total rainfall by counting bucket tips and resets the data every 24 hours for accuracy. In addition, ESP32-CAM also works to broadcast live videos by using a web server to display the video. The system undergoes iterative testing in real-world conditions to ensure its reliability. Calibration is performed to fine-tune sensor readings, and troubleshooting addresses any discrepancies in data transmission or alert mechanisms. This comprehensive methodology ensures the smart flood alarm system is robust, efficient, and capable of delivering accurate early warnings

3.2 Flowchart for Project Development

The flowchart focuses on the sequential phases of project development, starting with the selection and interfacing of the water level sensor, magnetic sensor and the interfacing of the NodeMCU for data collection and programming for the acquisition of the data required.

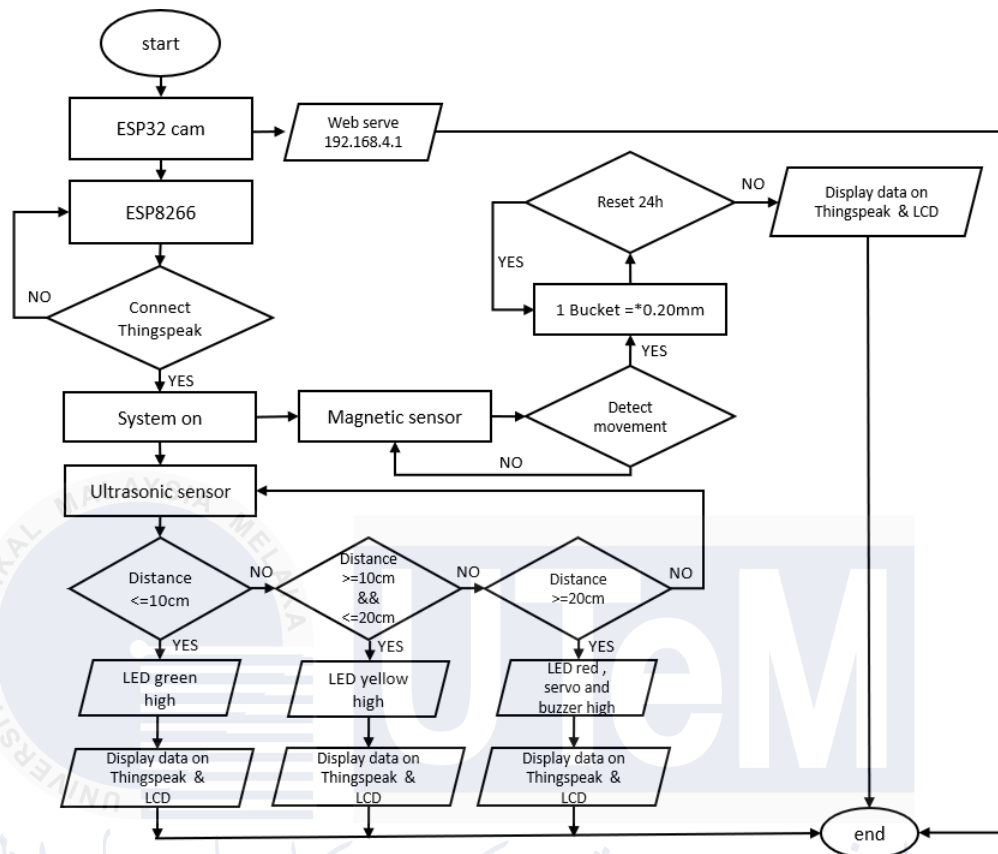


Figure 3.1 Flowchart for Smart Flood Alarm System

Based on figure 3.1 explains how the smart flood alarm system project works. ESP32 Cam is used to make real-time monitoring which is accessed via a web server with IP address 192.168.4.1. ESP8266 sends data to ThingSpeak. If ESP8266 cannot connect to ThingSpeak, the process will return to the beginning. If ESP8266 can connect to ThingSpeak, the system will start operating. There are two sensors used, the first is an ultrasonic sensor, measuring the distance and responding to the set distance. If the distance is less than 10cm, the green LED will light up and the data will be sent to ThingSpeak and displayed on the LCD. If the distance is more than 10cm and less than 20cm, the yellow LED will light up and the data will be sent to ThingSpeak and displayed on the LCD. If the distance is more than 20cm, the

red LED will light up, the buzzer will sound, in addition, the servo motor will move 180 degrees, and the data will be sent to ThingSpeak and displayed on the LCD.

The second sensor is a magnetic sensor, the magnetic sensor will detect when it detects the movement of the magnetic field found on the bucket tipping rain gauge. If it detects a magnetic field, it will be calculated by multiplying by 0.2mm. If it does not detect a magnetic field, no multiplication process will occur. Data will be recorded and read for up to 24 hours. If 24 hours are not enough, data will be taken continuously. If it has reached 24 hours, the data will be reset to 0mm and the data will be sent to ThingSpeak and displayed on the LCD.

3.3 Block Diagram for Project Development

Figure 3.2 shows the block diagram of the development of an IoT-based smart flood alarm system using NodeMCU ESP12-E. On the input side, two sensors are used which are Ultrasonic Sensor and Hall Effect Magnetic Detector Sensor. Where the function of ultrasonic is to measure the flood water level. Sensor and Hall Effect Magnetic Detector Sensor are for detecting magnetic fields, where the magnetic will be installed in the rain gauge to calculate how often the rain gauge moves.

The second part is the process. NodeMCU ESP12-E is used as a microcontroller. The data from the sensor will then be processed on the microcontroller. The microcontroller processes the sensor data to determine whether a flood condition exists and calculates the average rainfall. Controls device output (LED, Buzzer, LCD I2C) based on processed data. Send data to the ThingSpeak platform for remote monitoring and logging. On the output side, there are four outputs namely LED, Buzzer, LCD I2C, and ThingSpeak. Based on the processed data, the NodeMCU activates the LED and Buzzer for local alerts. The LCD I2C

displays the current water level and status. Simultaneously, NodeMCU sends the data to the ThingSpeak platform for remote monitoring

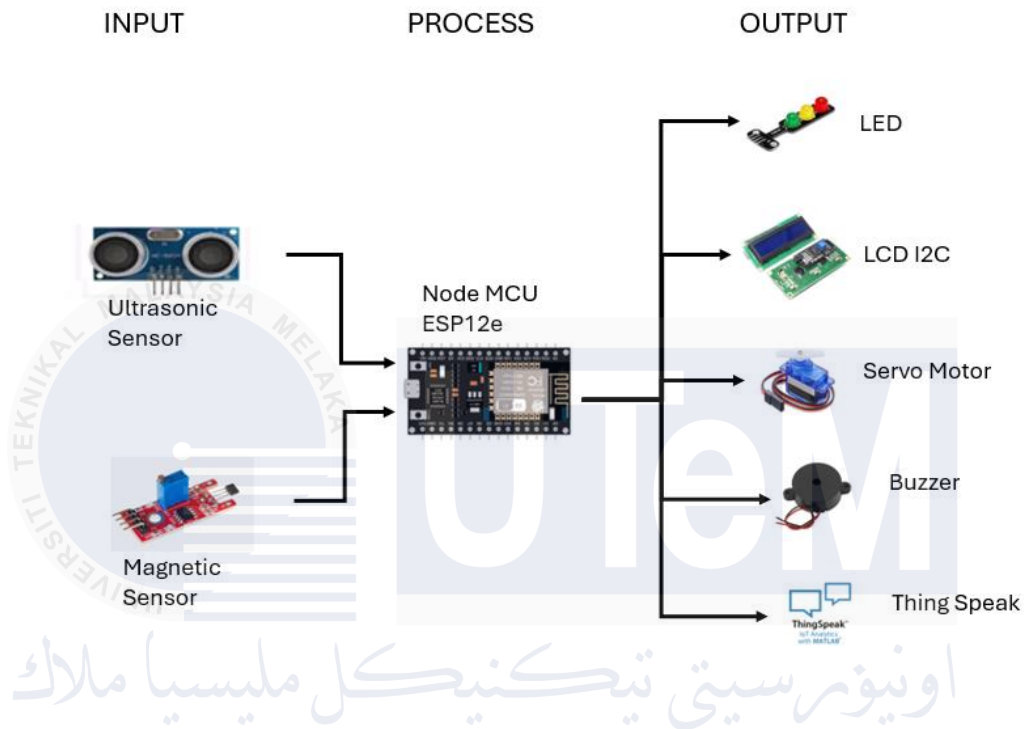


Figure 3.2 Block Diagram

Figure 3.2 shows the block diagram provides a clear overview of the components and their interactions in the development of an IoT-based smart flood alarm system project, showcasing how the system works from data acquisition to alerting and remote monitoring.

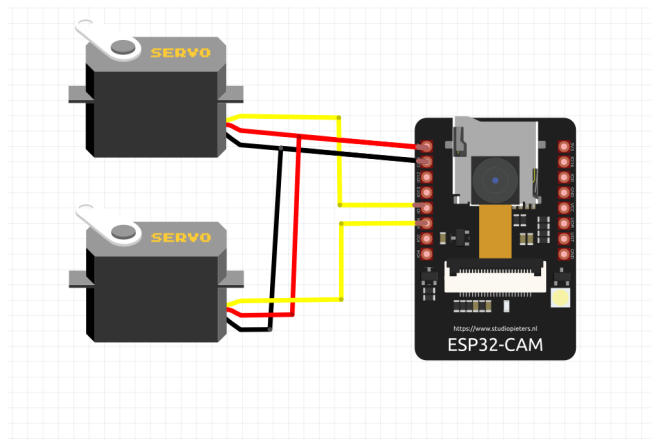


Figure 3.3 Connection ESP32-CAM

Figure 3.3 shows the ESP32-CAM connected to two servo motors configured for pan-tilt motion, enabling the camera module to move. Servo Motor 1 controls horizontal movement (pan), allowing the camera to rotate left and right, while Servo Motor 2 controls vertical movement (tilt), enabling the camera to move up and down. This setup provides full flexibility for the ESP32-CAM to monitor its surroundings. In a water level monitoring project, the ESP32-CAM streams video or captures images via a web server, allowing remote monitoring through a web browser. The servos are controlled using PWM signals programmed on the ESP32-CAM, enabling the camera's angle to be adjusted for specific areas based on user commands or automated settings.

3.4 Project Hardware Selection

The items used to complete this smart flood alarm system project are as shown in the following table.

Table 3.1 List of Component

No	Component	Price	Quantity	Total
1	NodeMCU ESP8266 ESP12-E	RM19.90	1	RM19.90
2	Ultrasonic Sensor	RM4.40	1	RM4.40
3	LED (Traffic Light LED Module)	RM2.50	1	RM2.50
4	Active Buzzer 5v	RM2.00	2	RM4.00
5	KY-024 hall magnetic sensor	RM4.50	1	RM4.50
6	LCD I2C 16x2	RM8.50	1	RM8.50
7	Cable Jumper (M-M, F-F, M-F)	RM5.00	3	RM15.00
8	Servo motor sg90	RM6.40	3	RM19.20
9	voltage regulator	RM14.50	2	RM29.00
10	Lithium battery 3.7v	RM10.00	3	RM30.00
11	Battery holder	RM3.00	1	RM3.00
12	ESP32 CAM	RM25.00	1	RM25.00
13	Acrylic Perspex	RM30.00	1	RM30.00
14	Tipping Rain Gauge	RM70.00	1	RM70.00
TOTAL				RM265.00

3.4.1 NodeMCU ESP8266 ESP12-E

In the project, the development of an IoT-based smart flood alarm system by using NodeMCU ESP8266 ESP12-E. NodeMCU ESP8266 ESP12-E is an open-source firmware and development kit based on the ESP-12E module, which is an ESP8266-based Wi-Fi module. NodeMCU V3 is closely related to the ESP8266 ESP-12E module, as it forms the basis for the wireless communication features of the NodeMCU development board. The ESP-12E module is a Wi-Fi SoC variant of the ESP8266, which is the core component of the NodeMCU board. It enables rapid prototyping of IoT products using the Arduino IDE. The board has multiple GPIO pins that support various communication protocols such as PWM, I2C, SPI, and UART. NodeMCU V3 is suitable for Wi-Fi applications and can be programmed using Arduino IDE software [30].

NodeMCU ESP8266 ESP-12E has advantages and disadvantages compared to various other platforms. To choose a suitable microcontroller for IoT projects, consider factors such as built-in Wi-Fi, processing speed (ESP32 has 240MHz), memory and storage options (ESP8266 and WeMos D1 R1 offer enough), and PIO and peripheral support (ARM mbed NXP LPC1768 provides broad continuous support).

Table 3.2 Comparison of micro controller

Type	Specifications		
	Wi-Fi Connectivity	Processing Power	Memory, Storage & Pin
ESP32	Integrated Wi-Fi and Bluetooth	Dual core, up to 240 MHz	Up to 16 MB Flash, 520 KB SRAM
Arduino Uno	No Wi-Fi	16 MHz, 8-bit	32 KB Flash, 2 KB SRAM, 1 KB EEPROM

ESP01	Integrated Wi-Fi	80 MHz, single core	1 MB Flash, 50 KB SRAM
Arduino Nano	No Wi-Fi	16 MHz, 8-bit	32 KB Flash, 2 KB SRAM, 1 KB EEPROM

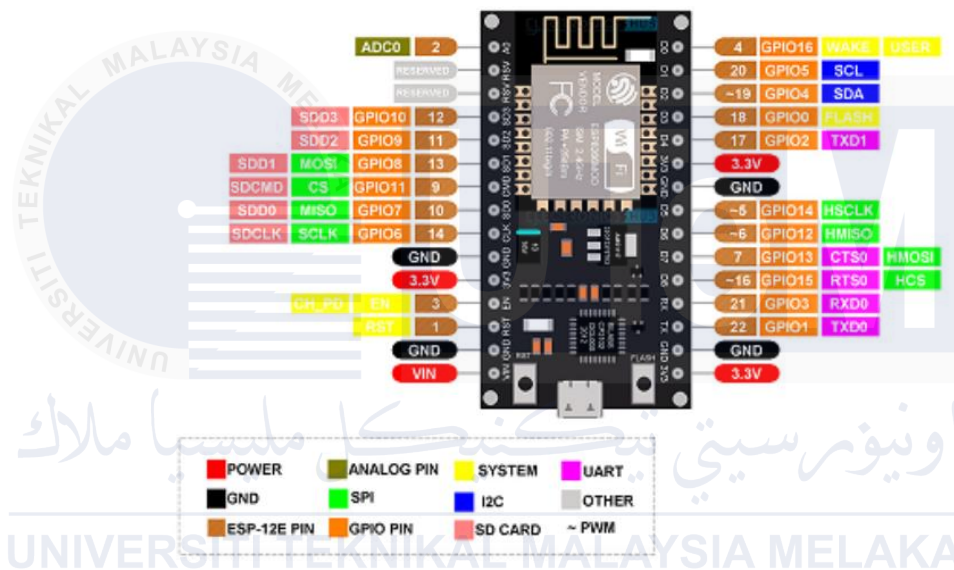


Figure 3.4 NodeMCU ESP8266 ESP12-E

Figure 3.4 shows NodeMCU ESP8266 ESP12. NodeMCU ESP8266 ESP-12E is version 3 NodeMCU based on the ESP-12E module, which is an ESP8266-based Wi-Fi module. The pinout of the NodeMCU board is displayed in the table below. If based on the original NodeMCU Devkit design, a typical NodeMCU board has 30 pins. Eight of the pins are needed for power, with two remaining. The remaining 20 pins link to the ESP-12E Module's pins.

The ESP8266 ESP12E type microcontroller was chosen for an IoT-based smart flood alarm system project due to its Wi-Fi connectivity and minimal pin usage of 10 I/O pins, making it the most suitable and affordable choice for this project.

3.4.2 Ultrasonic Sensor

An ultrasonic sensor is a device that uses ultrasonic sound waves to measure the distance between the sensor and the object. It emits sound waves that bounce back when it hits an object. The time taken by the wave to return is measured and converted to distance. This sensor is used in robotics, automation, and parking aids. There are several types of ultrasonic in the market and each ultrasonic has its advantages and disadvantages.

Table 3.3 Comparison of ultrasonic sensor

Type	Specifications		
	Detection Range	Accuracy	Price
HC-SR04	2cm-400cm	± 3 mm.	Cheap
MaxBotixLV- MaxSonar-EZ	0cm-645cm	± 1 cm.	Expensive
ParallaxPing	2cm-300cm	± 3 mm.	Normal

Ultrasonic sensor which is very used for measuring distance and HC-SR04 is utterly a sensor that recklessly uses an ultrasonic wave. For instance, people use it in robotics to control the motion of the robots as well as their avoidance system, tallying items and measuring the distance between objects. I will begin by outlining what is not, briefly defining it, and speaking of its main characteristics and applications. So, the HC-SR04 has a transmission and receiving element and it transmits an ultrasound wave and then measures the time duration taken to get the reflected wave back from the object. Adding the time taken by the echo and then multiplying it by the speed of sound and then the resultant distance of the object is obtained. [31].

Minimum range HC-SR04 sensor, which is also called distance measurement, means the minimum distance at which the sensor can minimize. This distance is normally of the order of 2cm. However, if the two objects are nearer each other than this distance, the sensor cannot discern between the two due to its extremely poor resolution in the short range. Thus, for the measurements to be accurate it is necessary to place the object at least 2 cm away from the Light Sensor.

On the other hand, the maximum range refers to the distance up to which the given sensor can detect an object with the maximal desirable distance. Based on the configuration of the HC-SR04 sensor the maximum range is only up to 400cm. Objects beyond this distance might go unnoticed because the reflected signal from the target can no longer be sensed due to the thresholds of the sensor. Therefore, this specific kind of sensor should be used where the range of measurements in centimeters is between 2 cm and 400 cm. To populate range, it is the time that elapses between the moment of sending a signal and the moment of receiving the echo signal. This formula is used near the determination of the distances through the ultrasonic waves, distances measured in centimeters CM are obtained from the microseconds (μs) travel time of the ultrasonic signal.

$$centimeter = \mu s / 58$$

The formula below is used to calculate the distance based on the travel time of the ultrasonic signal and the speed of sound. Where, distance is the measured distance, in meters (m) or centimeters (cm) depending on the unit used for the speed of sound. Duration is the travel time of the ultrasonic signal, usually measured in seconds (s) or microseconds (μs). The speed of sound is the speed of sound in air, about 340 meters per second (m/s) or 0.0343 centimeters per microsecond (cm/ μs).

$$Distance = \frac{(duration * speed\ of\ sound(340ms^{-1}))}{2}$$



Figure 3.5 Ultrasonic Sensor HC-SR04 [32].

Figure 3.5 shows an ultrasonic sensor HC-SR04 which has 4 pins. Vcc, Trig, Echo, and GND. Input the trigger to the module. The sensor requires a 10 μ s pulse on the Trig pin to start a measurement. This pulse informs the sensor to begin the distance-measuring procedure. The Echo pulse output to the user timing circuit. When ultrasonic waves strike an item and bounce back, the sensor's Echo pin turns high (emits a pulse). The duration of this strong signal is proportional to the time it takes for the waves to reach the item and return.

Figure 3.6 shows the ultrasonic sensor working. The Trig Pin is used as a signal transmitted to the object in front and the signal will be reflected and received by the Echo Pin. For this project, the Ultrasonic sensor measures the water level 3 distances. 400cm and above is a safe distance. 300cm to 400cm is a warning. 300cm below is a dangerous distance.

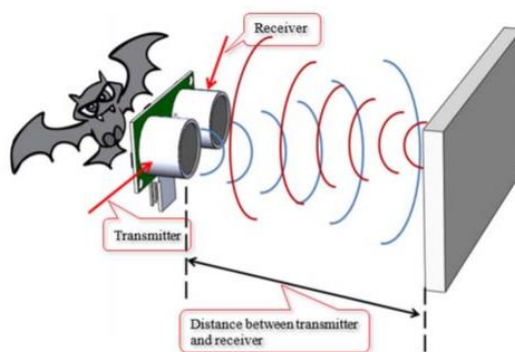


Figure 3.6 Ultrasonic Works [31].

The ultrasonic HC-SR04 was chosen for this project for several good reasons. First, it is the right choice because of its ability to measure distances, especially in the context of measuring water levels that require high accuracy. The project required accurate distance measurements for three categories of water levels: normal, precautionary, and dangerous. The HC-SR04 can measure distances with satisfactory accuracy, allowing accurate categorization for each water condition.

The HC-SR04 sensor is an economical choice for this project due to its reasonable price, offering high performance and value, ensuring efficient execution without compromising budget, making it an ideal choice.

3.4.3 LED (Traffic Light LED Module)

Light-emitting diodes (LEDs) are semiconductor devices that emit light when an electric current passes through them. Their primary function is to provide visual indicators, signaling the operational status of electronic devices [33]. There are many types of LEDs sold in the market, but not all LEDs can be used in a project. Table 3.4 shows the comparison of LED.

Table 3.4 Comparison of LED

Type	Specifications		
	Light Intensity	Shape	Price
Through-Hole LED	Relatively low, used for indication	consists of 1 LED and does not have a resistor	Cheap
Traffic Light LED Module	Very tall, designed for long-range visibility	consists of 3 LEDs and has a resistor	Expensive

The mini traffic light display module, with high brightness, is very suitable to produce traffic light system models. It stands out for its small size, simple wiring, and easy custom installation. Unlike Through-Hole LEDs, which are also small but used individually in basic electronics projects, mini traffic light modules combine several LEDs in one unit for more focused applications such as traffic light system simulations. Through-hole LEDs usually require more complicated wiring and installation, while these modules offer a more integrated and easy-to-use solution for specific purposes.



Figure 3.7 LED Traffic Light [34].

Figure 3.7 shows 4 pins. Pin R for LED Red, Pin Y for LED Yellow, Pin G for LED Green, and the last Pin is for GND. For this project, the traffic light module is used to give signals to users related to the increase in water level. When the water level is more than 400 cm, the green light will light up, while when the distance is 300 cm to 400 cm, the yellow LED light will light up, and when the distance is less than 300 cm red LED light will light up.

The IoT-based smart flood alarm system project utilizes an LED traffic light type due to its compact, integrated design, small size, easy wiring, and current-limiting resistor, making it a suitable choice for the project.

3.4.4 Buzzer

A buzzer is a simple but effective component for adding sound elements to a project/system. The tiny and compact 2-pin construction allows for easy usage on breadboards, Perf Boards, and PCBs. This makes it a commonly utilized component in most electrical applications. Buzzers are often classified into two sorts. A basic buzzer emits a continuous beep sound when energized, whereas a premade buzzer is larger and produces a beep. Beep. Beep [35]. There are many types of buzzers with different sizes and different sound levels. Table 3.5 shows the comparison of the buzzer.

Table 3.5 Comparison of Buzzer

Type	Specifications		
	Sound Frequency	Operating Voltage	Sound Level
Buzzer5v (Active/ Passive)	2kHz	5V	85dB 10cm
SFM-20B(Active)	2kHz ($\pm 500\text{Hz}$)	1.5V-3.6V	80dB 3V, 10cm
SFB-55(Active)	2.3kHz ($\pm 300\text{Hz}$)	3V-24V	95dB 12V, 30cm

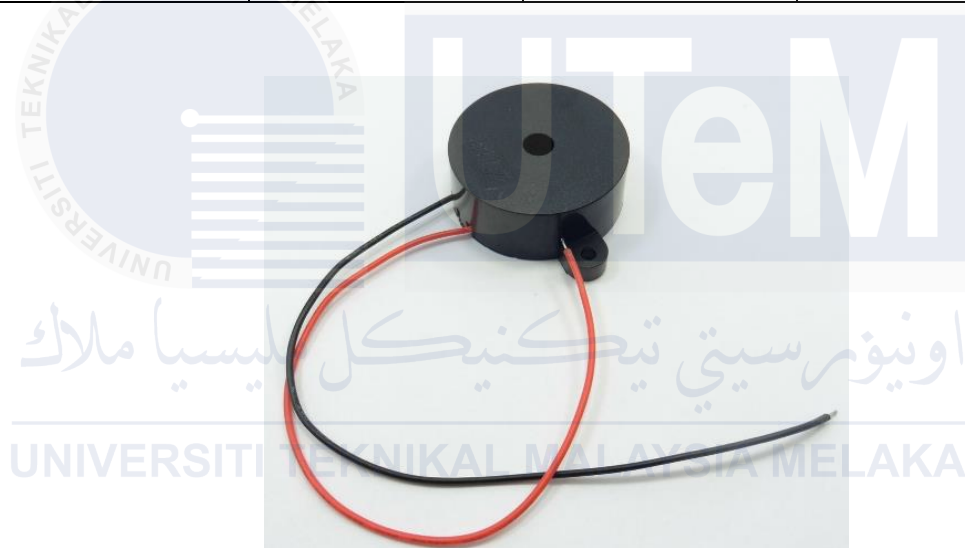


Figure 3.8 Buzzer 5v

The buzzer 5v was chosen for the IoT-based smart flood alarm system project compatibility with microcontroller logic voltages. Most microcontrollers (ESP8266, Arduino, etc.) operate on 5V or 3.3V logic voltages. 5V buzzers are easy to integrate as they are compatible with 5V-based systems without the need for additional circuitry such as voltage regulators.

3.4.5 Hall Magnetic Sensor

A Hall Magnetic Sensor is a device that detects magnetic fields. It consists of a Hall plate that releases electrons when exposed to a magnetic field. These electrons produce a voltage that is measured by the sensor. When the magnetic field changes, the direction of electron flow changes, causing a change in the detected voltage. This allows the sensor to detect changes in the magnetic field and produce an appropriate output signal. With that, Hall Magnetic Sensors are used in various applications such as position recognition, current measurement, and rotation detectors. There are various types of hall magnetic sensors that have different specifications. Table 3.6 shows the comparison of the hall magnetic sensor.

Table 3.6 Comparison of Hall Magnetic Sensor

Type	Specifications		
	Output signal	Operating Voltage	Sensitivity
KY-003	Signal digital	3.3V -5V	Low
KY-035	Signal analog	3.3V- 5V	Low
KY-024	Signal analog	3.3V-5V	High

KY-003 (Digital Hall Magnetic Sensor): Generates easy-to-interpret digital signals, ideal for fast detection in applications requiring a lightning response. Installation is simple and suitable for beginners or simple projects. KY-035 (Analog Hall Magnetic Sensor): Provides continuous magnetic field readings, suitable for applications that require continuous monitoring with high accuracy [36]. Flexible in use and commonly used in various projects. KY-024 (Analog Hall Magnetic Sensor): Very sensitive to current changes, suitable for high-precision AC or DC current measurements. Capable of producing stable

analog output, providing high reading accuracy in applications that require precise current monitoring.

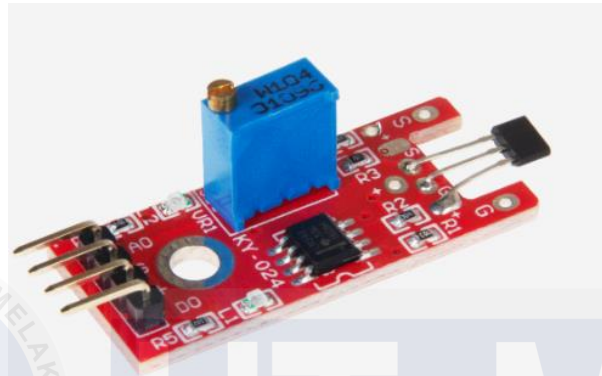


Figure 3.9 KY-024 Hall Magnetic Sensor [36].

Figure 3.9 shows the KY-024 hall magnetic sensor. This sensor is very important in the project development of an IoT-based smart flood alarm system because this sensor is used to calculate the amount of tipping rain gauge moves. This sensor will be installed on the outside of the tipping rain gauge and the magnetic field will be installed on the inside of the tipping rain gauge. The middle part moves when water passes through, and the sensor will detect the movement.

The Hall Magnetic Sensor KY-02 was chosen for an IoT-based smart flood alarm system project due to its stable analog output, easy integration with IoT platforms, low power consumption, high sensitivity to magnetic field changes, compact design, and quick response to flooding situations. This sensor enhances the system's reliability, performance, and effectiveness, supporting safety and risk mitigation in flood management.

3.4.6 LCD I2C

LCD 16x2 is a popular character display module that can display text and symbols in a format of 16 characters per line and two lines simultaneously. This module uses liquid crystal display (LCD) technology to display information and is usually used in electronic projects to display data such as text messages, temperature, time, and other information. 16x2 LCDs usually require a few pins to connect to a microcontroller or other operating system, and their setup and use require appropriate programming. There are several LCDs that are of different sizes and different installations. Table 3.7 shows the difference between LCDs that have I2C and those that do not have I2C.

Table 3.7 Comparison of LCD

Type	Specifications		
	Number of Cables	Operating Voltage	Price
LCD 16x2	16 cables (8 for data, 3 for control, 5V and GND).	5V	Cheap
LCD I2C 16x2	4 cables (VCC, GND, SDA, SCL) for I2C connection.	5V	Expensive

I2C (Inter-Integrated Circuit) LCD screens are liquid crystal displays that interface with microcontrollers or other processing units using the I2C protocol. A 16x2 LCD display with an I2C interface enables simple connection and communication with microcontrollers, decreasing the number of pins needed. This I2C Serial LCD module simplifies circuit

connections, saves I/O pins on the Arduino board, and facilitates firmware development with the widely available Arduino library [37].

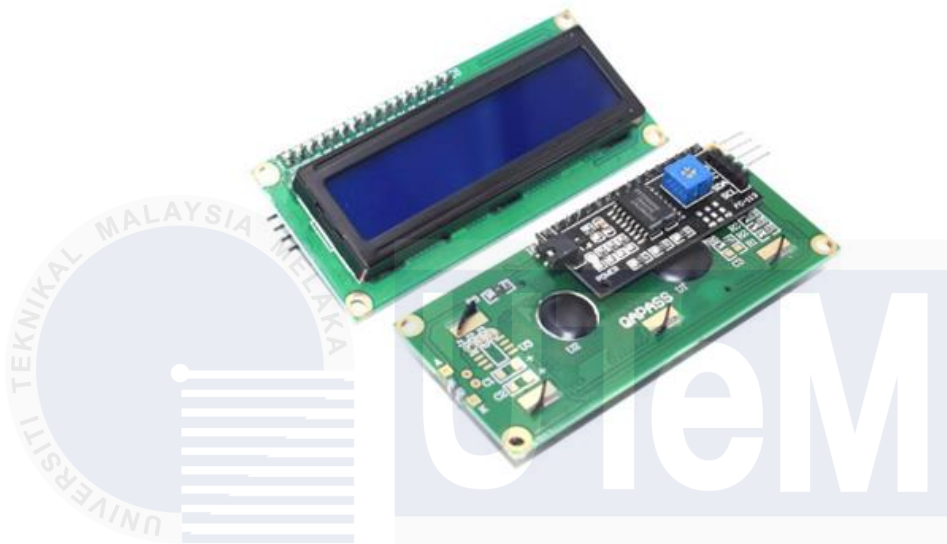


Figure 3.10 LCD I2C [37].

Figure 3.10 shows the LCD I2C. This type of LCD is very important in the development of an IoT-based smart flood alarm system project because this LCD is used to display the current water level and average rainfall. This LCD will display the water level distance read by the sensor. The water level distance will be displayed in cm units.

An LCD with an I2C interface has several advantages over a normal LCD that uses a parallel (parallel) interface. Here are some of the main advantages, among them is the reduction of Pins used. LCD I2C only needs 2 pins (SDA and SCL) for data and clock communication. The IoT-based smart flood alarm project utilized an I2C LCD for its ease of use, design cleanliness, and flexibility, compared to a normal LCD that requires 6-10 pins for control.

3.4.7 Tipping Rain Gauge

A tipping-type rain gauge is a device that measures the quantity of rain that falls. This contraption works by channeling rainwater via a funnel into a tiny seesaw-shaped container with two buckets on either side. When one of the buckets reaches a specific level, it will fill with water, releasing the stored water and repositioning the other bucket to collect additional rainfall. Each time the bucket is tilted, it is tallied, and the total number of tilts represents the total rainfall [8]. Figure 3.11 shows the parts of the tipping rain gauge.



Figure 3.11 Part of The Tipping Rain Gauge [8].

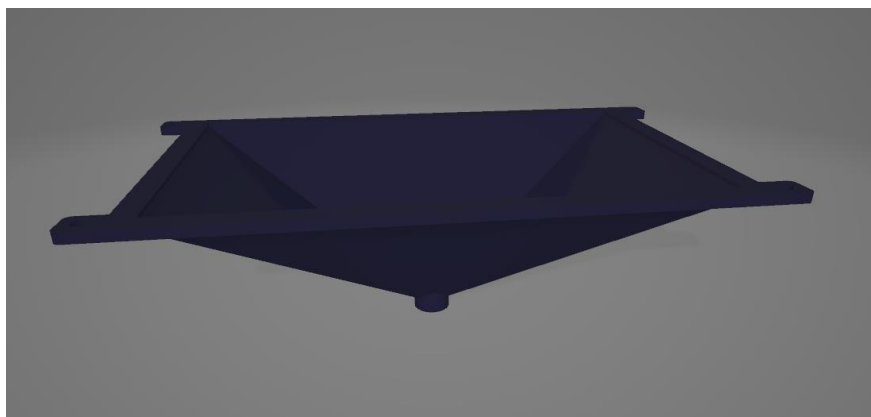


Figure 3.12 The Top of The Tipping Rain Gauge [8].

Figure 3.12 shows the top part of the tipping rain gauge. On the top part, there are two main functions, namely the funnel part. The function of the funnel is to direct rainwater to the tilted bucket mechanism. The second is the cover. The function of the cover is to protect the components from dirt.



Figure 3.13 The Middle Part of The Tipping Rain Gauge [8].

Figure 3.13 shows the center section for the tipping rain gauge which is divided into three. The first bucket is tipping. Once the bucket collects enough water (usually 0.2 mm to 0.5 mm of rainfall, it tips, dumping the water and positioning the other bucket under the funnel. The second one is counting. Each tip is counted by a mechanical counter or a magnetic reed switch and electronic counter. The third is a shaft mechanism, allowing the bucket to tilt and return to its initial position.

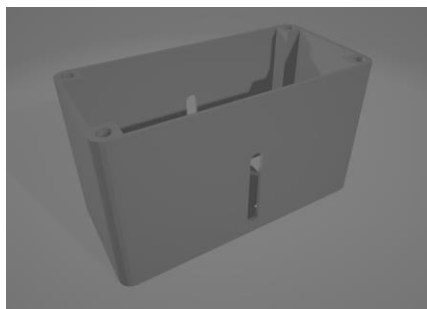


Figure 3.14 The Bottom of The Tipping Rain Gauge [8].

Figure 3.14 shows the bottom part of the tipping rain gauge, the main function of this part is as a part to combine the 2 parts earlier. To calculate the amount of rainwater, use the formula. To get the most accurate measurement possible, count the 100milliliters needed to move the centerpiece 22 times. By averaging, it got 4.5 milliliters per displacement. After that, it is necessary to calculate the surface area on the top of the tipping rain gauge by using the formula of a rectangle.

$$\text{Area} = \text{lenght} * \text{width}$$

After calculating the volume. By using the formula below the average rainfall will be able to be calculated.

$$\text{Average rainfall} = \frac{\text{total tipping} \times \text{sensitivity}}{\text{area}}$$

- Total Tipping is the number of times it rains on the gauge.
- Sensitivity (S) is the amount of rain (usually in mm) required to make the gauge "tipping" or moving.
- Catchment Area (A) is the surface area of the measuring device that catches rain (usually in cm² or m²).

3.4.8 Servo Motor

Servo motors are precise actuators that regulate position, speed, and torque in a range of applications. A servo's essential components are a DC or AC motor, a control circuit, and a feedback mechanism, such as a potentiometer. There are several types of servo motors on the market.

Table 3.8 Comparison of Servo

Type of servo	Voltage	Angle	Function	Price
SG90	4.8v-6v	0° - 180°	lifting light objects	Low
MG996R	4.8v-7.2v	0° - 180°	lifting medium- heavy objects	Medium expensive
DS3218	4.8v-6.8v	0° - 270°	lifting heavy objects	Expensive



Figure 3.15 Servo Motor SG90

In this project I chose a servo as in figure 3.15, which is a sg90 servo motor because for this project the things that need to be moved using this servo are a buzzer and an ESP32 cam, each weighing less than 10g. So, this servo is very suitable for this project.

3.4.9 Voltage Regulator

A voltage regulator is an electronic component that keeps the electrical voltage constant despite changes in the load or power source. It guarantees that electrical equipment receives the necessary and safe voltage to perform correctly, whether fixed (linear) or variable (switching).

Table 3.9 Comparison of Voltage Regulator

Type	Voltage	Efficiency	Lcd
LM2596	4v-40v	High	Yes
LM7805	7v-35v	Low	No
AMS1117	6v-15v	Low	No

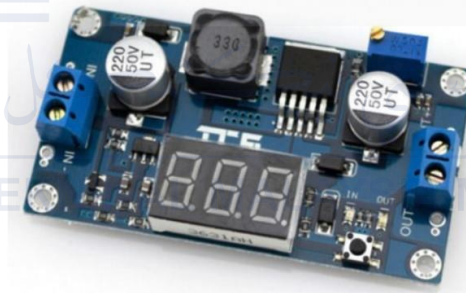


Figure 3.16 Voltage Regulator LM2596

Table 3.9 shows the difference in types of voltage regulators. For this smart flood alarm system project, a voltage regulator like Figure 3.16 uses an LM2596 voltage regulator because this voltage regulator has many advantages compared to other voltage regulators, including the LM2596 has an LCD to display the input and output voltage coming out of the battery. In addition, the LM2596 can lower the voltage that is compatible with the ESP8266, which is 5v.

3.4.10 ESP32 CAM

ESP32-CAM is a microcontroller module that includes an ESP32 microprocessor and an OV2640 camera. It is compatible with Wi-Fi and Bluetooth, making it ideal for surveillance, facial recognition, and video streaming applications.

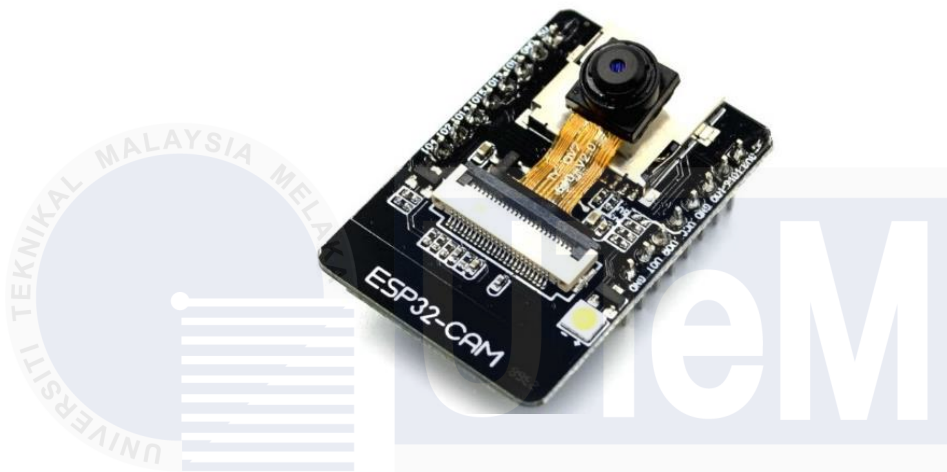


Figure 3.17 ESP32 CAM

This ESP32 CAM was chosen for this project because this ESP32CAM can monitor via the cam directly which is connected to the web server.

3.4.11 Acrylic Perspex

Acrylic's strength, adaptability, and beautiful look make it suitable for a variety of applications, including signs, displays, windows, and protective coverings. It is readily cut, mounded, and shaped, making it a popular choice for both commercial and DIY applications.

Table 3.10 Comparison of Material

Type	Material	Formability	Durability
Acrylic (Perspex)	Transparent, strong plastic material.	Easy to shape, cut, bend, and engrave.	Very durable

Wooden Board	Made from wood.	It can be cut, but harder to shape precisely.	Durable but can be affected by moisture and weather.
Cardboard	Laminated paper or lightweight board.	It is easy to cut and fold but lacks durability.	Less durable,

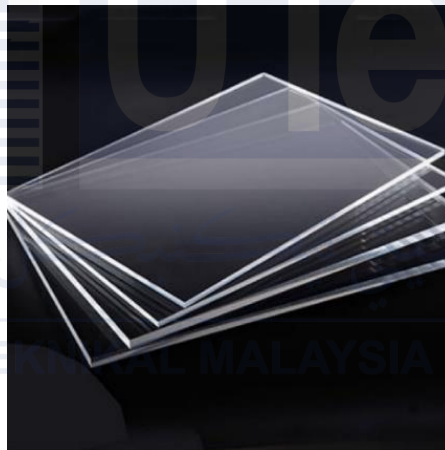


Figure 3.18Acrylic Perspex

There are 3 materials chosen to make the prototype for this smart flood alarm system project. The differences in terms of material, formability and durability found in table 3.10 have shown that acrylic Perspex material as in figure 3.18 is the most suitable material to use for making prototypes because it is easy to shape and durable.

3.4.12 Lithium Battery 3.7v

Batteries are the most important energy source in this project because with the right energy source, data can be sent accurately and correctly. There are several types of batteries available on the market, including lithium batteries, AA batteries, AAA batteries and others. Table 3.11 below shows several types of batteries with their actual differences.

Table 3.11 Comparison of Batteries

Type	Voltage	Capacity	Chemistry
Lithium Battery	3.7V	3500mAh	Lithium-ion
AA Battery	1.5V	1800mAh	Alkaline
AAA Battery	1.5V	1000mAh	Alkaline



Figure 3.19 Lithium Battery

Based on the differences in table 3.11, Lithium Battery is the most suitable battery for this project because of its large voltage and capacity, suitable for controlling microcontrollers and other inputs and outputs. When the voltage is high, it can be reduced using a voltage regulator. In addition, the battery can also be recharged using charging.

3.5 Project Software Selection

In the development of an IoT-based smart flood alarm system using the NodeMCU ESP8266 ESP-12E, selecting the appropriate software tools is crucial for efficient project development and implementation. This section will outline the software tools and platforms chosen for this project, along with the rationale behind their selection.

3.5.1 Arduino IDE

Arduino IDE is one of the most crucial tools to write and send code to Arduino microcontroller boards and other connected peripherals or modules such as NodeMCU ESP8266 ESP 12E. This is where programming software for microcontrollers, which will foster the coupling of circuit design and embedded codes come into play; programming, compiling, and uploading sets of instructions to these microcontrollers is made easy to make [38]. The Arduino IDE is software for creating a basic platform for users where basic tools are also provided for them along with other additional tools feasible for the experts. It can accommodate a large and different sized group of microcontrollers with a variety of Arduino boards and similar devices like ESP8266 used.



Figure 3.20 Arduino IDE [38].

Figure 3.20 shows the Application of Arduino IDE. On the nature and benefits side of things, the first benefit of Arduino IDE is that it is friendly and the fact that Arduino IDE is friendly is good for beginners. A large Community Base means that there are extensive communities that support it, this can come in the form of many tutorials, forums, and resources for the large community of users of various software. Scalability and Swiss army knife: designed to accommodate almost any microcontroller while being supported by a wide library. Arduino's integrated development environment is a crucial factor in the usage of an IoT-based smart Flood Alarm System by NodeMCU ESP8266 ESP-12E and other microcontrollers. Furthermore, the availability of many users in the community and different guides and tutorials make it suitable for both beginners and advanced users within IoT and embedded system programming domains because of its simplicity and functionality.

3.5.2 ThingSpeak

ThingSpeak is an IoT platform that enables the user in a certain way to view real-time data concerning ThingSpeak or call an app based on the rules as such that ThingSpeak attains. This is because LoRa improves IoT applications from the instruments, which, earlier, receive data from the integrated sensors or, which received data also, and to apologize for the assembly data in addition to recognizing the received data. [39]. This should also be about explaining and further extend this ThingSpeak yet proceed to demonstrate how this is done also in this too as well as show how this ThingSpeak can integrate with NodeMCU ESP8266 Smart Flood Alarm System. This is well suitable in IoT activities particularly for the schools, those who are somewhat interested in IoT, and the small-minded petty businesses in IoT since it is incredibly easy to run as seen above coupled with power as pointed above. Another product which was created under MathWorks was considered in the

same way as the announcement to develop and release this product and MATLAB to support the compatibility of Analytical and Visualization toolbox.

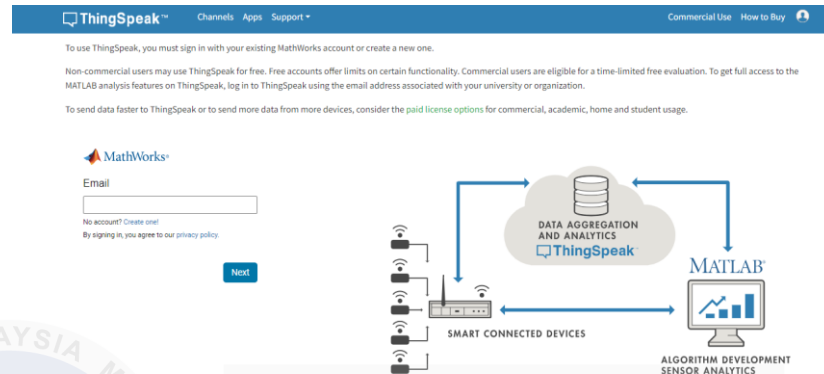


Figure 3.21 ThingSpeak

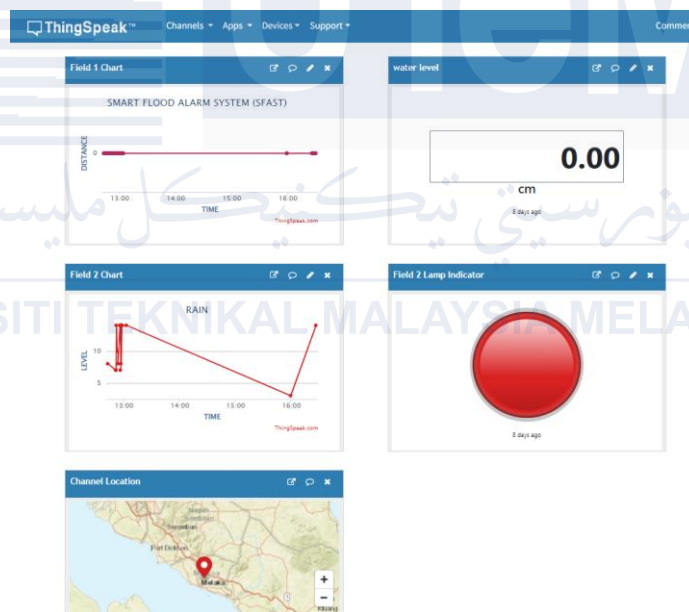


Figure 3.22 Function of ThingSpeak

Figure 3.22 shows the display for the project development of an IoT-based smart flood alarm system by using NodeMCU Esp8266 Esp12-E. Where the display will produce a graph of the water level and real time. Then the water level will be updated every 15 seconds. The second display is always showing the water level. The third display is the average rainfall that is calculated and converted into a graph and the data will be saved. The

fourth display is a lamp indicator intended to show a warning when the water level is at a dangerous level. The fifth display is the location. The location where this project is installed, with this location it will be easier to monitor.

3.5.3 Web Server

The web server is used for the ESP32 CAM to access the camera through the web server. Live view can be seen through the web server and then the servo motor to move the camera can also be moved using the web server. Figure 3.23 shows the web server that is used in smart flood alarm system.

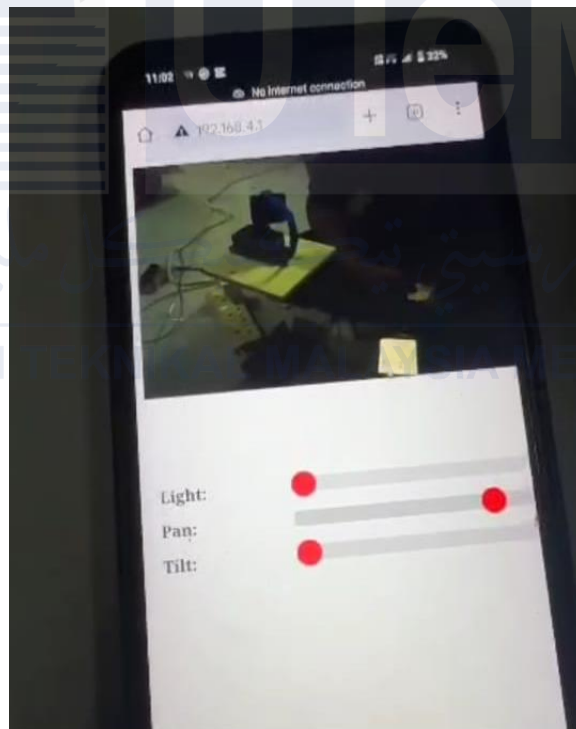


Figure 3.23 Web Server

3.5.4 Fritzing

Fritzing is a valuable tool for designing a Smart Flood Alarm System that utilizes components like the ESP12E, ultrasonic sensor, magnetic hall sensor, servo motor, buzzer, LED, and LCD. It allows developers to create a clear and efficient prototype of the circuit. Fritzing provides an intuitive platform for visualizing, designing, and documenting the circuit, making the development of this flood alarm system more streamlined and error-free.

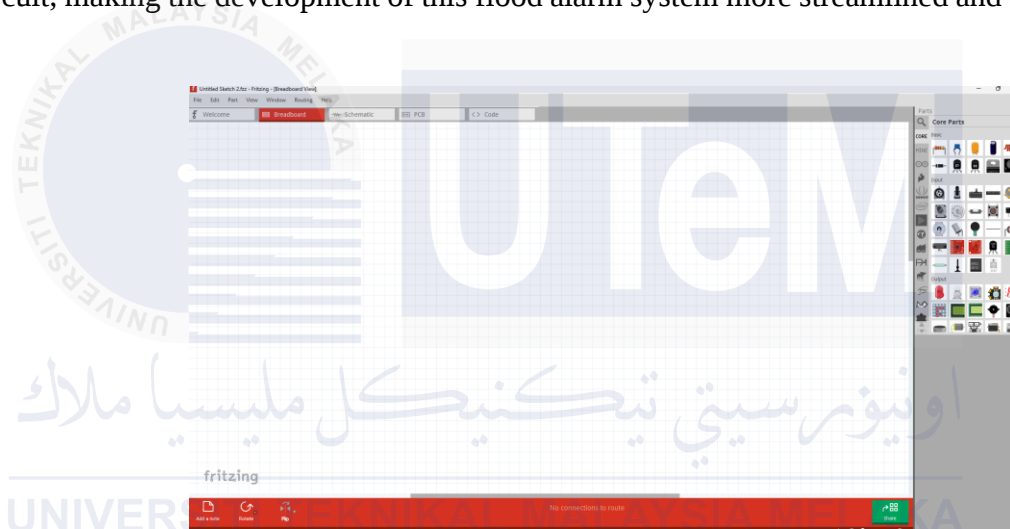


Figure 3.24 Fritzing

CHAPTER 4

RESULTS AND DISCUSSION



4.1 Introduction

In this chapter, we will present the results of the development and testing of an IoT-based smart flood alarm system using NodeMCU ESP8266 ESP12-E. The results are analyzed to evaluate system performance, reliability and effectiveness in detecting flood conditions. Upon the conclusion of the project, it underwent testing. Successful achievement was noted in the wireless communication and transmission of data between NodeMCU ESP12e, Ultrasonic sensor, tipping rain gauge. The sensed data were displayed through the ThingSpeak application and concurrently recorded in the database developed within

ThingSpeak application. This prototype underwent testing in distinct locations, and the ensuing results are deliberated upon in the subsequent sections.

4.2 Software Part

The software component of the IoT-based smart flood alarm system is critical to its operation, as it integrates sensor data collecting, processing, and communication with IoT platforms. This section covers the software architecture, development environment, coding, and integration with Wi-Fi and IoT systems.

4.2.1 Coding for ESP12e

The ultrasonic sensor is used to measure the water level distance. For this prototype, the distance used is in cm units. Estimate the depth of the river to be 35cm. If the distance between the river level is 0-10cm, the green LED light will light up to indicate the normal level. If the river level distance is 10-20 cm, the yellow LED will light up to indicate the alert level. If the distance between the river level exceeds 20cm, the red LED light will light up, the buzzer will sound, and the servo motor starts to move 180 degrees. The buzzer and red LED are connected once using pin D8. The river water level will be displayed on the i2C LCD, and the data will be uploaded to ThingSpeak.

To calculate the actual average rainfall, based on a report that has been studied with an area of Malacca measuring 1650 square km. In the Jasin dam area with an average rainfall of 100mm per day, floods can occur. So, for this prototype, the estimated value is as follows which will be explained on the coding, the magnetic sensor will detect the movement of the magnetic field where every time the movement contains rainwater as much as 0.20 mm of water is needed to move the rain gauge. So, every time the movement will be multiplied by 0.20mm. The data will be sent to the ThingSpeak application. The data will be updated every

24 hours, every 24 hours the data will start with 0mm. For this coding every 2 minutes the data will be reset to take new data every 2 minutes. Appendix A coding for first part for measure the water level and average of the rain.

4.2.2 Coding for ESP32-CAM

This code is designed to control an ESP32-CAM module with a Pan-Tilt system using two servo motors for horizontal (pan) and vertical (tilt) movement. It connects the ESP32-CAM to a WiFi network and starts a camera server accessible through a web browser using its IP address. The servo motors are controlled via input from the Serial Monitor using commands like PAN or TILT followed by an angle (0° to 180°). The ESP32-CAM's camera is initialized with specific settings to capture images or stream video. This code is ideal for projects requiring surveillance with dynamic camera angle adjustments. Appendix B coding for second part for esp32-cam.

APPENDIX A

```
//IOT

#include <Wire.h>
#include <ESP8266WiFi.h>
#include <ThingSpeak.h>

//WiFi credentials
const char *ssid = "Abe_domak21";
const char *password = "faiz2222";
// ThingSpeak credentials
unsigned long channel = 2383083;
const char *apiKey = "GRVULX5TJQKTSEVF";
const char *server = "api.thingspeak.com";

//WiFi client
WiFiClient client;

//lcd

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2); // 0x27 adalah alamat I2C standar

//servo
#include <Servo.h> // Library Servo
#define SERVO_PIN D3
Servo myServo;

// ULTRA
#define TRIG_PIN D5 // D5
#define ECHO_PIN D0 // D6
```

```
// MG

#define RAIN_PIN D4

volatile int rainCount = 0;

unsigned long lastResetTime = 0; // Variabel untuk menyimpan waktu terakhir reset

const unsigned long resetInterval = 120000; // Interval reset dalam milidetik (2 menit)
```

```
// LED
#define LED_GREEN D6
#define LED_YELLOW D7
#define LED_RED D8

void IRAM_ATTR rainInterrupt()
{
    rainCount++;
}
```

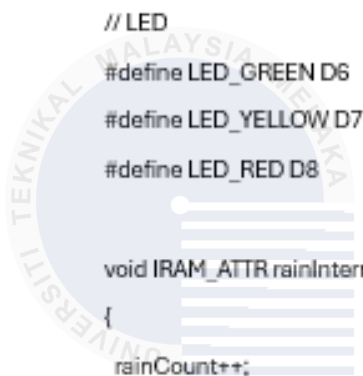
```
void setup()
{
    // Inisialisasi koneksi WiFi
    WiFi.begin(ssid, password);

    Serial.print("Connecting to WiFi ");
    while (WiFi.status() != WL_CONNECTED) {
        delay(10);
        Serial.print(".");
    }

    Serial.println();

    Serial.println("WiFi connected");

    // Inisialisasi ThingSpeak
    ThingSpeak.begin(client);
```



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

```

// ULTRA
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);

// MG
pinMode(RAIN_PIN, INPUT);
attachInterrupt(digitalPinToInterrupt(RAIN_PIN), rainInterrupt, FALLING);

// LED
pinMode(LED_GREEN, OUTPUT);
pinMode(LED_YELLOW, OUTPUT);
pinMode(LED_RED, OUTPUT);

// Servo
myServo.attach(SERVO_PIN);
myServo.write(0); // Posisi awal servo di 0 derajat

```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

```

// Inisialisasi Serial
Serial.begin(115200);
}

void loop()
{
    // Inisialisasi LCD
    lcd.init();
    lcd.backlight();
    lcd.clear();

    // MG
    static int lastRainCount = 0;
    if (millis() - lastResetTime >= resetInterval)

```

```

{
  // Reset rainfall setiap 2 menit
  lastResetTime = millis();
  rainCount = 0;
  Serial.println("Rainfall reset to 0.");
}

```

```

if (rainCount != lastRainCount)
{
  lastRainCount = rainCount;
}
float rainfall = lastRainCount * 0.7;
Serial.print("Rainfall: ");
Serial.print(rainfall);

```

```

Serial.println(" mm/h");
// ULTRA

```

```

long duration;
float distance;

```

```

digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
duration = pulseIn(ECHO_PIN, HIGH);

```

```

// Hitung jarak (kecepatan suara adalah 34300 cm/s)
distance = (duration * 0.0343) / 2;
// Balikkan jarak untuk ditampilkan
float invertedDistance = 35.0 - distance; // Misalkan batas maksimum adalah 20 cm
if (invertedDistance < 0)

```

```

{
    invertedDistance = 0; // Jangan biarkan nilai negatif
}

// Tampilkan jarak terbalik di Serial Monitor
Serial.print("Inverted Distance: ");
Serial.print(invertedDistance);
Serial.println(" cm");

// Menampilkan water level di baris atas LCD
lcd.setCursor(0, 0); // Baris pertama, kolom pertama
lcd.print("level: ");
lcd.print(invertedDistance);
lcd.setCursor(12, 0); // Baris pertama, kolom pertama
lcd.print("cm");

lcd.setCursor(0, 1); // Baris kedua, kolom pertama
lcd.print("rain: ");
lcd.print(rainfall);

lcd.setCursor(12, 1); // Baris pertama, kolom pertama
lcd.print("mm/h");

// Kontrol LED berdasarkan invertedDistance
if (invertedDistance >= 0 && invertedDistance <= 10)
{
    digitalWrite(LED_GREEN, HIGH);
    digitalWrite(LED_YELLOW, LOW);
    digitalWrite(LED_RED, LOW);
}
else if (invertedDistance > 10 && invertedDistance <= 20)
{

```

```

digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_YELLOW, HIGH);
digitalWrite(LED_RED, LOW);
}
else if (invertedDistance > 20 && invertedDistance <= 35)
{
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_YELLOW, LOW);
digitalWrite(LED_RED, HIGH);

// Servo bergerak ke sudut 180 derajat
for (int angle = 0; angle <= 180; angle += 1) {
myServo.write(angle); // Servo bergerak dari 0 hingga 180 derajat
delay(15); // Waktu jeda untuk pergerakan servo
}
for (int angle = 180; angle >= 0; angle -= 1)
{
myServo.write(angle); // Servo bergerak kembali dari 180 hingga 0 derajat
delay(15);
}
}
else
{
// Jika tidak ada dalam rentang, matikan semua LED
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_YELLOW, LOW);
digitalWrite(LED_RED, LOW);
}

int Z = ThingSpeak.writeField(channel, 1, invertedDistance, apiKey); // Mengirim jarak ke field
1

int Y = ThingSpeak.writeField(channel, 2, rainfall, apiKey); // Mengirim hitungan magnet ke
field 2
}

```

APPENDIX B

```
#include <WiFi.h>

#include <esp_camera.h>

#include <ESP32Servo.h>

#define SERVO_PAN_PIN 14

#define SERVO_TILT_PIN 15

Servo servoPan;

Servo servoTilt;

const char* ssid = "Your_SSID";

const char* password = "Your_PASSWORD";

#define PWDN_GPIO_NUM -1

#define RESET_GPIO_NUM -1

#define XCLK_GPIO_NUM 0

#define SIOD_GPIO_NUM 26

#define SIOC_GPIO_NUM 27

#define Y9_GPIO_NUM 35

#define Y8_GPIO_NUM 34

#define Y7_GPIO_NUM 39

#define Y6_GPIO_NUM 36

#define Y5_GPIO_NUM 21

#define Y4_GPIO_NUM 19

#define Y3_GPIO_NUM 18

#define Y2_GPIO_NUM 5

#define VSYNC_GPIO_NUM 25

#define HREF_GPIO_NUM 23

#define PCLK_GPIO_NUM 22

void startCameraServer();

void setup(){

  Serial.begin(115200);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

    delay(500);
```

```

Serial.print("");

}

Serial.println("WiFi connected");
Serial.println(WiFi.localIP());

// Setup servo motor
servoPan.attach(SERVO_PAN_PIN);
servoTilt.attach(SERVO_TILT_PIN);

// Tetapkan sudut permulaan
servoPan.write(90);
servoTilt.write(90);

// Setup kamera
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;

config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;

config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;

config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;

config.pin_d7 = Y9_GPIO_NUM;

config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;

```

```

config.pixel_format = PIXFORMAT_JPEG;

if(psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.println("Camera init failed with error 0x%x", err);
    return;
}

startCameraServer();

```

```

Serial.println("Camera Ready! Use 'http://" + WiFi.localIP().toString() + "' to connect");
}

void loop() {
    // Kawalan servo untuk pan-tilt dengan serial input

    if (Serial.available() > 0) {
        String command = Serial.readStringUntil("\n");

        if (command.startsWith("PAN")) {
            int angle = command.substring(3).toInt();

            angle = constrain(angle, 0, 180);

            servoPan.write(angle);

            Serial.println("Pan: " + String(angle));
        } else if (command.startsWith("TILT")) {
            int angle = command.substring(4).toInt();

            angle = constrain(angle, 0, 180);

```

```
servoTilt.write(angle);  
  
Serial.println("Tilt: " + String(angle));  
}  
}  
}  
  
void startCameraServer() {  
  // Kod pelayan kamera di sini (boleh rujuk library ESP32-CAM)  
  }
```



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

4.3 Hardware Part for ESP12e

The hardware component of the IoT-based smart flood alarm system is critical to its functioning since it contains all the physical components that detect water levels and sound alerts. This section describes the hardware architecture, component selection, circuit alerts. This section describes the hardware architecture, component selection, circuit design, and assembly instructions. Figure 4.1 shows the circuit of the project.

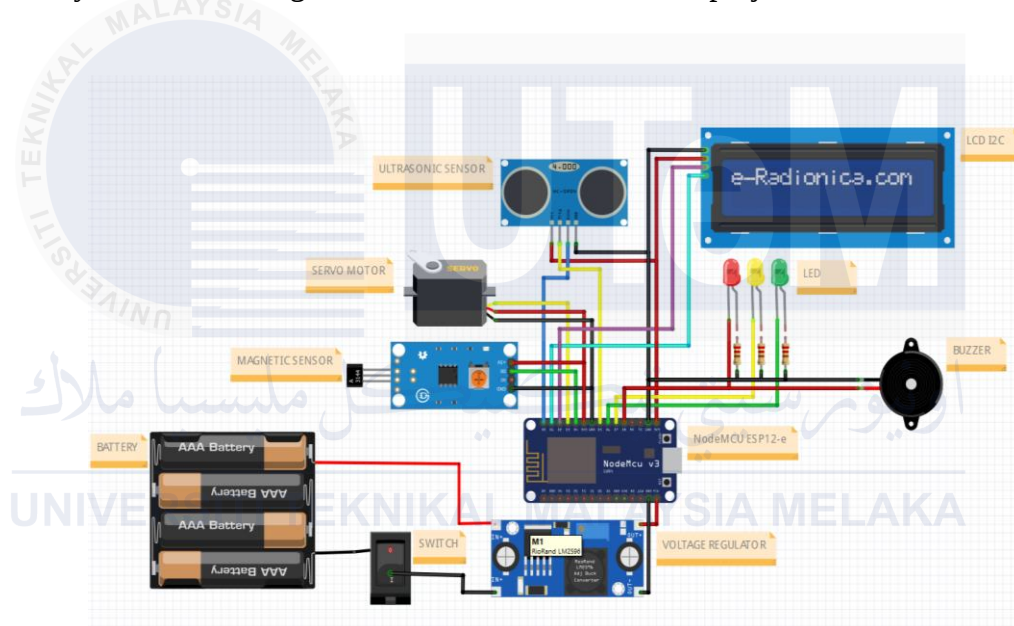


Figure 4.1 Circuit for Smart Flood Alarm System

4.4 Hardware Part for ESP32-CAM

The figure 4.2 shows the ESP32-CAM connected to two servo motors configured for pan-tilt motion, enabling the camera module to move. Servo Motor 1 controls horizontal movement (pan), allowing the camera to rotate left and right, while Servo Motor 2 controls vertical movement (tilt), enabling the camera to move up and down. This setup provides full flexibility for the ESP32-CAM to monitor its surroundings. In a water level monitoring project, the ESP32-CAM streams video or captures images via a web server, allowing remote

monitoring through a web browser. The servos are controlled using PWM signals programmed on the ESP32-CAM, enabling the camera's angle to be adjusted for specific areas based on user commands or automated settings

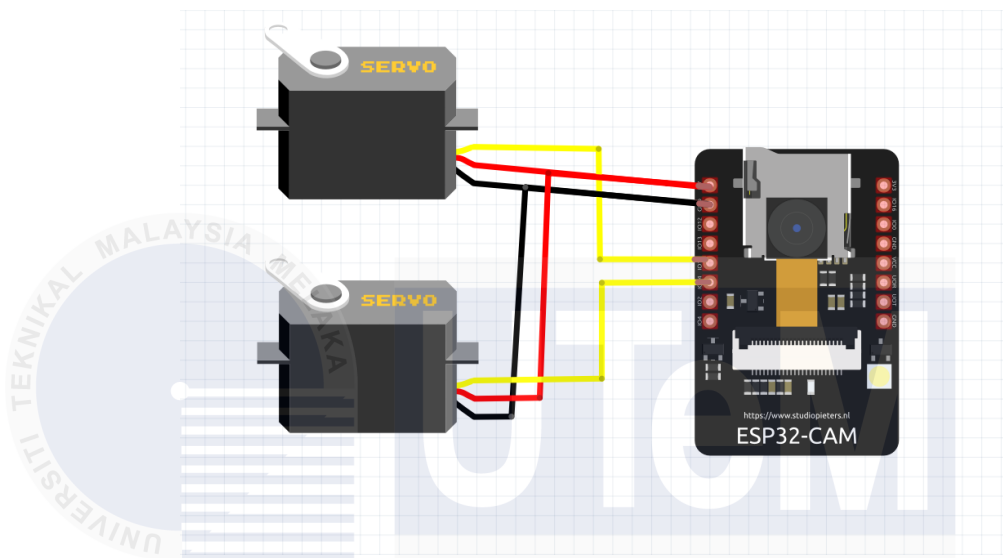


Figure 4.2 Circuit for ESP32-CAM Monitoring

4.5 Result for ESP12e

This section describes the results of designing and testing an IoT-based smart flood alarm system with the NodeMCU ESP8266 ESP12-E. The system's functionality, dependability, and efficacy in detecting flood conditions and sending out timely notifications are used to assess the outcomes. The data obtained from the water level sensor, the alarm system's responsiveness, and the interaction with the IoT platform are all covered in depth.

4.5.1 Water Level

When the rainwater level exceeds 30cm, the red LED lights up and will be displayed on the I2C LCD and the same data will be sent to Figure 4.3 below, the estimated depth of the river water level is 30cm. The closer the water level is to the sensor, the greater the displayed value. ThingSpeak takes 15 seconds to read data. All data taken will be 15 seconds late.

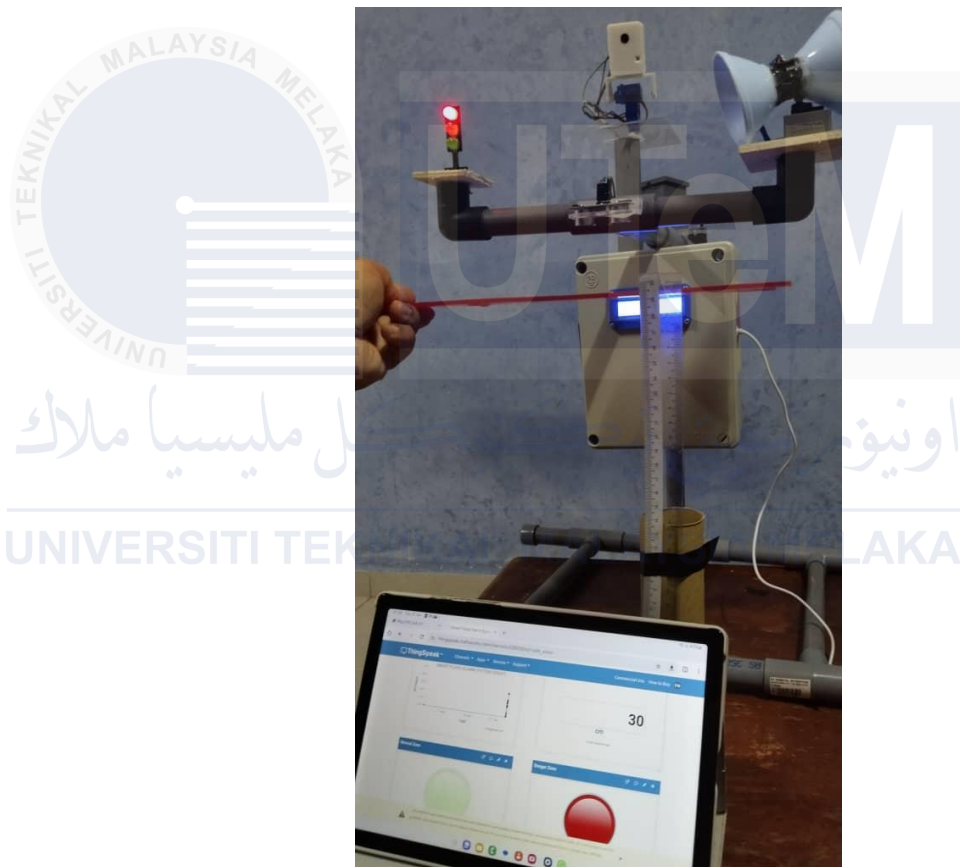


Figure 4.3 Result 1 for Water Level

When the rainwater level exceeds 10cm to 20cm, the yellow LED will light up and will be displayed on the I2C LCD and the same data will be sent to Figure 4.4 below, the estimated depth of the river water level is 14cm. The closer the water level is to the sensor, the greater the displayed value. ThingSpeak takes 15 seconds to read the data.

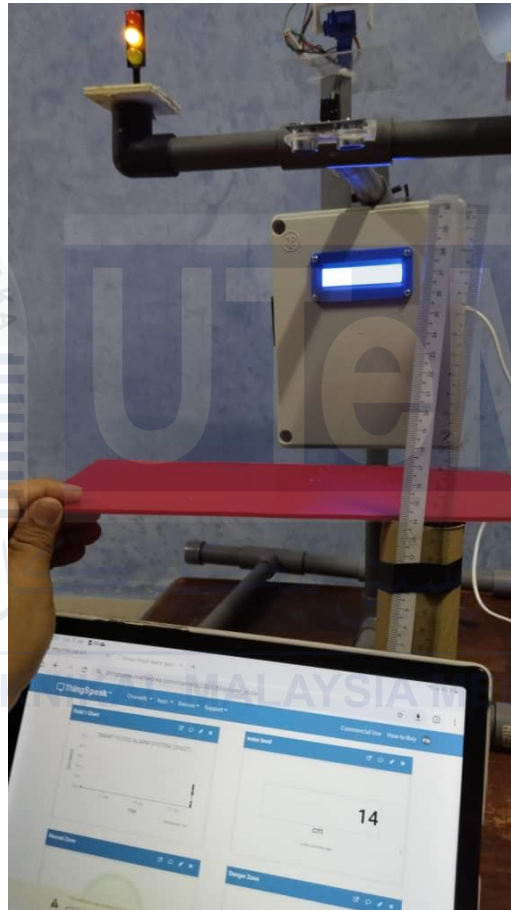


Figure 4.4 Result 2 for Water Level

When the rainwater level is less than 10cm, the green LED will light up and will be displayed on the I2C LCD and the same data will be sent to Figure 4.5 below, the estimated depth of the river water level is 0cm. The closer the water level is to the sensor, the greater the displayed value. ThingSpeak takes 15 seconds to read the data.



Figure 4.5 Result 3 for Water Level

The graph below shows the height of the water level read by the sensor and then sent directly to ThingSpeak. Each point on the graph takes 15 seconds for ThingSpeak to update the data.

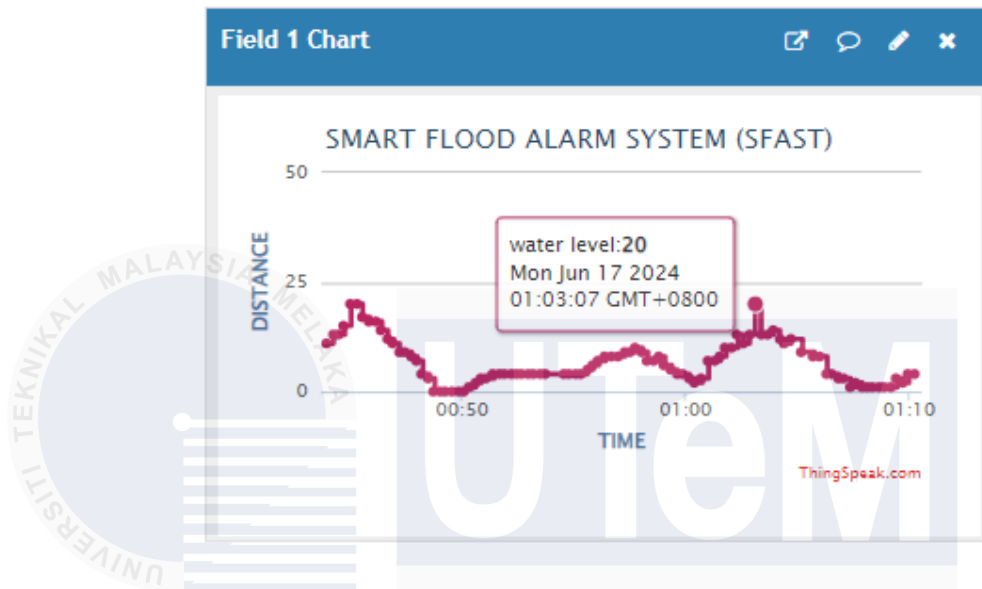


Figure 4.6 Graph for Water Level

4.5.2 Average of rain

Figure 4.7 shows the average rainfall data taken. For this prototype, the LED indicator used in this part is not coded with ESP, but in ThingSpeak it is possible to set a value for the LED to light up. As in the picture below. When the average rainfall reading is 0mm to 1mm, the green LED will light up to indicate a drizzle is taking place.



Figure 4.7 Result 1 for Rainfall

Figure 4.8 shows the average rainfall data taken. When the average rainfall reading is 1mm to 2mm, the yellow LED will light up to indicate moderate rainfall is occurring.

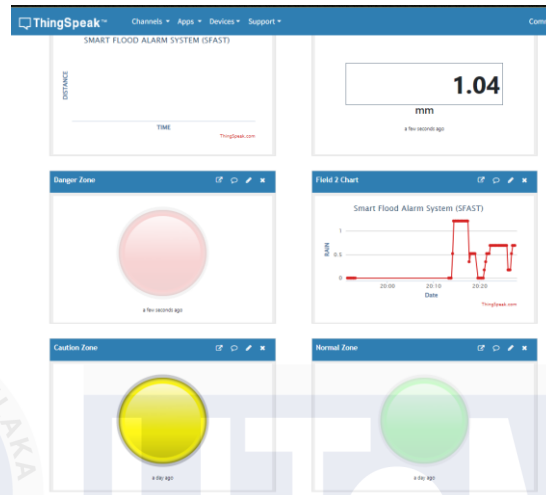


Figure 4.8 Result 2 for Rainfall

Figure 4.9 shows the average rainfall data taken. When the average rainfall reading exceeds 2mm, the red LED will light up to indicate heavy rain is occurring.

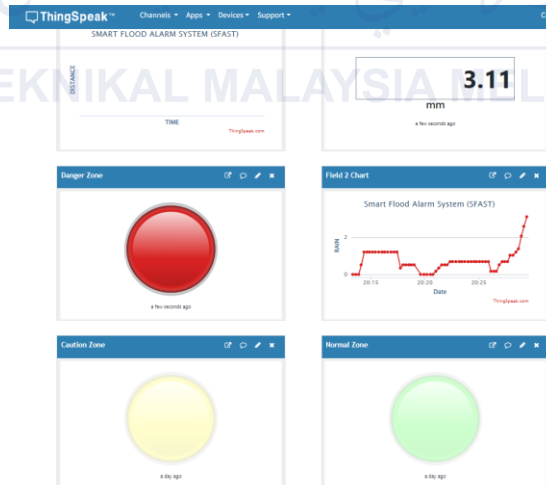


Figure 4.9 Result 3 for Rainfall

4.6 Result for ESP32-CAM

The figure 4.10 shows a user interface on a mobile browser connected to an ESP32-CAM module, displaying a live video feed streamed from the camera via a local WiFi network. Below the video, there are slider controls labeled Light, Pan, and Tilt. The "Light" slider likely adjusts the brightness of an LED connected to the ESP32-CAM, while the "Pan" and "Tilt" sliders control the horizontal and vertical movement of servo motors attached to a pan-tilt mechanism, allowing dynamic adjustment of the camera's viewing angle.

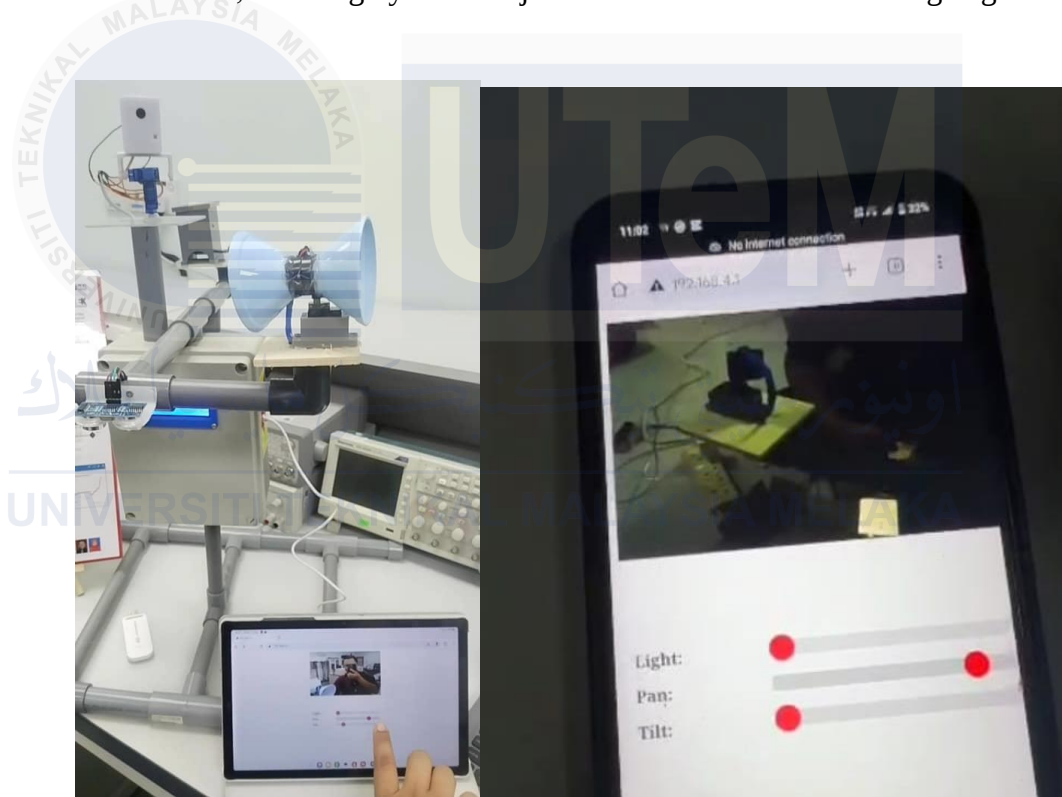


Figure 4.10 ESP32-CAM Monitoring

4.7 Prototype

A prototype is an initial model or trial version of a product, system, or idea developed for testing and evaluation purposes. It is used to identify weaknesses, test functionality, and gather feedback before creating the final, complete version. Prototypes help speed up development and ensure the product meets user requirements. Figure 4.11 shows the prototype for this smart flood alarm system project that uses a PVC box container to place electronic components such as batteries, microcontrollers, voltage regulators and other connections. This PVC box serves to protect the components from being exposed to water that will cause a short circuit.

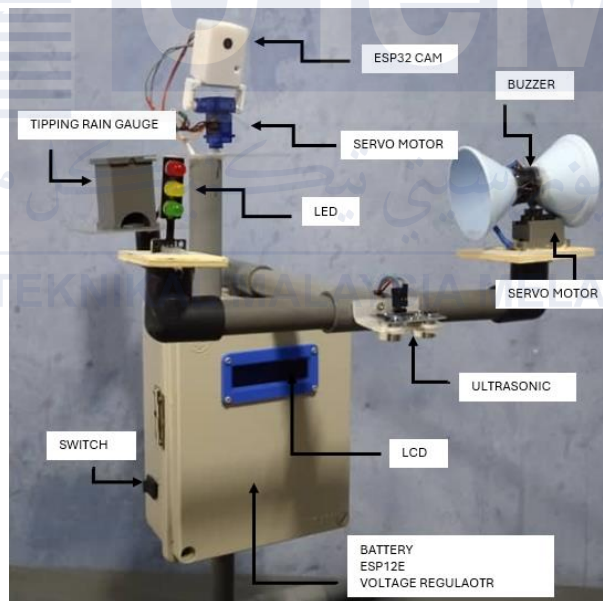


Figure 4.11 Prototype

Based on figure 4.11. ESP32-CAM, a camera module that can capture videos and send data to other devices via Wi-Fi. It's commonly used for surveillance or visual analysis applications. In this project the function is to create a surrounding monitor to see the water level. Tipping rain gauge, a device to measure rainfall. It works by collecting rainwater in a funnel that tips when a certain amount of water is collected, sending a signal to record the

rain measurement. Servo motors are used to move the ESP32 cam and buzzer. The purpose of moving the buzzer using a servo motor is to send a 360-degree sound wave. In addition, the purpose of moving the ESP32 cam is to record live videos more widely. LED lights used as indicators. For example, green for normal status, yellow for warnings, and red for emergencies. Buzzer component that produces sound, typically used for alerts or emergency notifications. Ultrasonic is to measure the distance of the water level that has been set and is divided into three parts, namely normal, warning and danger. The LCD is to display the amount of rain and the amount of water level detected by the sensor. The switch connected from the battery to the voltage regulator is used as an on/off microcontroller.



CHAPTER 5

CONCLUSION AND FUTURE WORKS



5.1 Conclusion

The goal of creating an IoT-based smart flood alarm system using NodeMCU ESP8266 ESP-12E is to improve flood monitoring capabilities, deliver timely warnings, and allow for better data visualization and analysis. The system focuses on real-time flood monitoring by continuously measuring water levels and accurately measuring distances using the HC-SR04 ultrasonic sensor. The buzzer provides timely alerts for growing water levels. ThingSpeak IoT data visualization and analysis in the cloud improves the system when integrated with it by combining all data into one location.

5.2 Recommendation of the Future

To improve the future IoT based smart flood alarm system. Given the right implementation of the AI and machine learning for analysis, chances of obtaining an actual improved flood warning system based on IoT in the future are possible. With the help of the meteorological data received from other stations, the occurrence of possible flood events can be estimated due to previous records made for the previous floods. Furthermore, it would be beneficial to include more connection points which can possibly employ 5G for better performance regarding sensors and cloud services. Creation of an application for monitoring and control on the portable device will assist users get prompt notification and real time data. Last but not the least, the incorporation of solar-powered sensors, therefore, are as sustainable relevant as they are important when there is a power blackout.

REFERENCES

- [1] 2022 Jabatan Pengairan dan Saliran Malaysia and Laporan, “Laporan Banjir Tahunan 2011/2012,” pp. 1–104, 2012.
- [2] StatsMalaysia, MyCensus(2020), and DOSM, “Dosm : Melaka Negeri Bersejarah,” pp. 1–10, 2020.
- [3] B. Terkini, “Hujan cecah 100mm antara punca banjir di Melaka,” pp. 1–7, 2023.
- [4] U. Aras and A. Data, “Data Aras Air Sungai,” pp. 6–7, 2024.
- [5] D. I. Jawa and D. I. Jawa, “Data Hujan Tahunan,” pp. 15–16.
- [6] J. Adi, S. Nordin, M. Mohammad Tahir, S. Azali, H. Diana, and G. Fionna, “Pengaruh monsun terhadap bahaya banjir: kajian kes dataran banjir beaufort, sabah,” *J. Kinabalu*, vol. 26, no. 2, pp. 165–182, 2020.
- [7] K. A. Raza and W. Monnet, “Moving objects detection and direction-finding with HC-SR04 ultrasonic linear array,” *Proc. 5th Int. Eng. Conf. IEC 2019*, pp. 153–158, 2019, doi: 10.1109/IEC47844.2019.8950639.
- [8] B. G. Aggiustatutto, “DIY Arduino Rain Gauge Introduction : DIY Arduino Rain Gauge Step 1 : How It Works,” pp. 1–10.
- [9] A. Khan *et al.*, “Flood monitoring and warning system: Het-sens a proposed model,”

2020 2nd Int. Conf. Comput. Inf. Sci. ICCIS 2020, no. 4, 2020, doi: 10.1109/ICCIS49240.2020.9257693.

[10] A. R. Rajput, D. B. Kumbhar, H. J. Joshi, N. M. Dhawale, R. H. Joshi, and S. S. Patil, "Flood Level Control and Management Using Rajput, Aditya R. Kumbhar, Dinesh B. Joshi, Hrushikesh J. Dhawale, Nandkishor M. Joshi, Rewa H. Patil, Shweta S. Instrumentation and Control," 2021 Int. Conf. Comput. Commun. Green Eng. CCGE 2021, pp. 1–5, 2021, doi: 10.1109/CCGE50943.2021.9776471.

[11] J. Bhuvana and T. P. Siva Shankar, "Early Flood Detection and Environment Monitoring System," Int. Interdiscip. Humanit. Conf. Sustain. IIHC 2022 - Proc., pp. 1418–1422, 2022, doi: 10.1109/IIHC55949.2022.10060568.

[12] J. W. Simatupang and T. A. Arrazaq, "Design Implementation of Flood Early Warning System for Residential Monitoring," Proceeding - 2023 Int. Conf. Radar, Antenna, Microwave, Electron. Telecommun. Empower. Glob. Prog. Innov. Electron. Telecommun. Solut. a Sustain. Futur. ICRAMET 2023, pp. 158–163, 2023, doi: 10.1109/ICRAMET60171.2023.10366589.

[13] R. M. D. Charaan, J. Shobana, P. Krishnamoorthy, B. A. Princy, R. J. Abinaya, and K. Murugesan, "Enhancement of IoT based Flood Detection and Prevention using Arduino UNO with WiFi Module," 2023 9th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2023, pp. 2290–2294, 2023, doi: 10.1109/ICACCS57279.2023.10112673.

[14] Z. A. Zainal and F. Ahmad, "Smart Flood Monitoring System via IoT Platform for Early Warning of Road Closures Against Flood Events," vol. 5, no. 1, pp. 389–396, 2024.

[15] S. Binti Zahir *et al.*, "Smart IoT Flood Monitoring System," J. Phys. Conf. Ser., vol. 1339, no. 1, p. 012043, Dec. 2019, doi: 10.1088/1742-6596/1339/1/012043.

[16] H. F. A. Sufa, M. I. Yusof, and M. A. A. Sani, "Flood Monitoring and Warning System With Iot," Mjit, vol. 3, no. 2, 2019.

[17] N. A. Z. M. Noar and M. M. Kamal, "The development of smart flood monitoring system using ultrasonic sensor with blynk applications," 2017 IEEE Int. Conf. Smart

Instrumentation, Meas. Appl. ICSIMA 2017, vol. 2017-Novem, no. November, pp. 1–6, 2017, doi: 10.1109/ICSIMA.2017.8312009.

[18] B. M. Shankar, T. J. John, S. Karthick, B. Pattanaik, M. Pattnaik, and S. Karthikeyan, “Internet of Things based Smart Flood forecasting and Early Warning System,” *Proc. - 5th Int. Conf. Comput. Methodol. Commun. ICCMC 2021*, no. Iccmc, pp. 443–447, 2021, doi: 10.1109/ICCMC51019.2021.9418331.

[19] M. E. Deowan, “Smart Early Flood Monitoring System Using IoT”.

[20] M. Karthika, R. Mohan Das, E. J. D’Cunha, A. A. Dar, G. P. Kumar, and A. Kumar, “Solar Fed Flood Alert System Using Arduino,” *3rd IEEE Int. Conf. Technol. Eng. Manag. Soc. Impact using Mark. Entrep. Talent. TEMSMET 2023*, 2023, doi: 10.1109/TEMSMET56707.2023.10150080.

[21] T. Opasworakun, “Water Level Monitoring and Flood Alerting by Applying IoT,” pp. 11–14, 2024.

[22] S. A. Ali, F. Ashfaq, E. Nisar, and U. Azmat, “A Prototype for Flood Warning and Management System using Mobile Networks,” 2019.

[23] A. R. Rajput, D. B. Kumbhar, H. J. Joshi, N. M. Dhawale, R. H. Joshi, and S. S. Patil, “Flood Level Control and Management Using Instrumentation and Control,” *2021 Int. Conf. Comput. Commun. Green Eng. CCGE 2021*, pp. 1–5, 2021, doi: 10.1109/CCGE50943.2021.9776471.

[24] M. Ragnoli, V. Stornelli, D. Del Tosto, G. Barile, A. Leoni, and G. Ferri, “Flood monitoring: a LoRa based case-study in the city of L’Aquila,” *PRIME 2022 - 17th Int. Conf. Ph.D Res. Microelectron. Electron. Proc.*, pp. 57–60, 2022, doi: 10.1109/PRIME55000.2022.9816747.

[25] G. Nitish Satya Sai, K. S. Manikanta, S. G. Arjula, R. Sudheer, B. N. Rao, and S. N. Gamidi, “Flood Prediction System using IoT and LoRa Technologies,” *2021 IEEE Bombay Sect. Signat. Conf. IBSSC 2021*, 2021, doi: 10.1109/IBSSC53889.2021.9673453.

- [26] R. Ismail, W. S. Suhaili, and R. K. Patchimuthu, "IoT Based Water Quality Monitoring in Relation to Flood and Drought in Brunei Darussalam," *2023 13th Int. Conf. Inf. Technol. Asia, CITA 2023*, pp. 54–59, 2023, doi: 10.1109/CITA58204.2023.10262743.
- [27] A. Prafanto and E. Budiman, "A Water Level Detection: IoT Platform Based on Wireless Sensor Network," *Proc. - 2nd East Indones. Conf. Comput. Inf. Technol. Internet Things Ind. EIconCIT 2018*, pp. 46–49, 2018, doi: 10.1109/EIconCIT.2018.8878559.
- [28] T. M. Thekkil and N. Prabakaran, "Real-time WSN based early flood detection and control monitoring system," *2017 Int. Conf. Intell. Comput. Instrum. Control Technol. ICICICT 2017*, vol. 2018-Janua, pp. 1709–1713, 2017, doi: 10.1109/ICICICT1.2017.8342828.
- [29] N. A. Z. M. Noar and M. M. Kamal, "The development of smart flood monitoring system using ultrasonic sensor with blynk applications," *2017 IEEE Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2017*, vol. 2017-Novem, no. August, pp. 1–6, 2017, doi: 10.1109/ICSIMA.2017.8312009.
- [30] A. Al Dahoud and M. Fezari, "NodeMCU V3 For Fast IoT Application Development," *Notes*, no. October, p. 5, 2018.
- [31] Handson Technology, "Handson Technology User Guide HC-SR04 Ultrasonic Sensor Module User Guide User Guide: Ultrasonic Sensor V2.0," pp. 1–10, 2024.
- [32] Cytron Technologies Sdn. Bhd., "HCSR04 User ' s Manual," no. 408, pp. 1–38, 2013.
- [33] Zaini Miftach, "Light Emitting Diode," pp. 53–54, 2018.
- [34] Damascus, "412 ARDUINO LED TRAFFIC LIGHTS MODULE Description : Features :," p. 412.
- [35] "buzzer," p. 2023, 2023.
- [36] F. Sensorkit and X. Wiki, "KY-024 Linear magnetic Hall Sensor Pinout Code example Arduino Code example Raspberry Pi," p. 1397.

- [37] Handson Technology, “I2C Serial Interface 1602 LCD Module,” *User Guid.*, pp. 1–8, 2021.
- [38] M. Fezari and A. Al Dahoud, “Integrated Development Environment ‘ IDE ’ For Arduino,” no. October, 2018.
- [39] Hans-Petter Halvorsen, “ThingSpeaks,” pp. 1–26, 2017.



APPENDICES

week Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14
BDP Briefing														
Learn how to make a report														
Choose a title and submit a proposal														
Draf and submit a literature review														
Email week logbook at Epsm (TASK1)														
Draf and submit Chapter 1 and Chapter 2														
Draf and submit Chapter 3														
Draf Chapter 4 and Chapter 5														
Email week logbook at Epsm (TASK2)														
Presentation														

week Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Claim form														
Calibration Sensor														
Prototype Project														
Email week logbook at Epsm (TASK1)														
Test Project														
Draf and submit Poster														
Update Report														
Email week logbook at Epsm (TASK2)														
Presentation														