SMART FOREST FIRE DETECTION

FILZA ATHIRAH BINTI NOR ASHIKIN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

SMART FOREST FIRE DETECTION

FILZA ATHIRAH BINTI NOR ASHIKIN

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

[SMART FOREST FIRE DETECTION]

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

STUDENT: FILZA ATHIRAH BINTI NOR ASHIKIN Date : 23/8/2024

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

SUPERVISOR : TS. ERMAN BIN HAMID Date :

DEDICATION

In the name of God, the Most Gracious, the Most Merciful. This project, *Smart Forest Fire Detection*, is dedicated to my family, friends and mentors for their unyielding encouragement which has been the basis of my achievements. Thank you to my mother and father who always pushed me to follow whatever interests I possess and thanks too, to all my buddies as well as colleagues who have always been a continuous source of inspiration. My heartfelt thanks go to supervisor/evaluator whose insightful guidance and constructive criticism were instrumental in taking this project from concept to reality. This work is also dedicated to those working tirelessly towards safeguarding our environment in the hope that this **Smart Forest Fire Detection** system would help in preserving and protecting nature's beautiful sceneries for generations to come .

ACKNOWLEDGEMENTS

Firstly, I would like to show my appreciation to En Erman Bin Hamid for having guided me and given me some priceless advice in the course of this project. His support always, coupled with his guidance, made it possible for me to deal with various challenges that come with the completion of this final year project 2021/2022 session. My evaluator too should not be left behind; Dr. Nazrulazhar Bahaman who gave his time to evaluate my work. Through his evaluation he was able to offer constructive feedback that helped me identify areas of improvement thus sharpening up my project.

Moreover, UTeM (Universiti Teknikal Malaysia Melaka) has provided a conducive environment and needed facilities where I could successfully carry out this project. Finally, i would like to extend my sincere gratitude towards my family and friends who have been so supportive and understanding. They paved way for me by believing in myself and together we were able to overcome the challenges that had surrounded us making us complete this assignment within the specified period of time.

ABSTRACT

The Smart Forest Fire Detection system is designed to provide real-time monitoring and alerts for forest fire incidents, addressing the urgent need for early detection and rapid response to prevent extensive damage. The project utilizes the ESP32 microcontroller, along with flame and smoke sensors, to continuously monitor environmental conditions and detect potential fires. Upon detection, the system sends immediate alerts to users via the Blynk mobile application, enabling prompt action to mitigate risks.

The project was motivated by the limitations of existing forest fire detection systems, which often lack real-time updates and accurate reporting of fire incidents. To overcome these challenges, the project was developed with three main objectives: to study and design methods for delivering real-time alerts and updates, to develop an application that accurately reports fire-related data such as smoke levels, and to thoroughly test the system's features to ensure reliability.

The hardware components, including the flame and smoke sensors integrated with the ESP32, are responsible for detecting fire conditions and transmitting data to the Blynk application. The system has undergone extensive testing to ensure that it functions effectively and meets the project's goals.

In conclusion, the Smart Forest Fire Detection system offers a significant improvement over traditional detection methods, providing a reliable solution for early detection and response to forest fires, ultimately helping to protect valuable natural resources and communities

TABLE OF CONTENTS

Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The alarming frequency of forest fires and their catastrophic consequences underscore the urgent need for an advanced fire defense system. Traditional fire safety measures often suffer from delayed notifications and lack a swift, automated response mechanism. In critical situations, every second counts, and the absence of a comprehensive solution puts lives and property at risk. This system will read the heat, flame, and smoke data using IoT and it will trigger the automatic water sprinkler.

Furthermore, most traditional fire detectors do not offer notifications through modern applications. Therefore, by using the innovation that is smart forest fire detection, users who have the application will receive an alert notification if there is a fire occurring in the nearby forest. This can help reduce fatal accidents among users because they can be alerted through the application. The identified shortcomings in current forest fire safety systems, including delayed responses and imprecise firefighting measures, emphasize the necessity for innovation. Smart Forest Fire Detection aims to address these challenges by implementing advanced sensor technologies for enhanced early detection and ensuring a swift response to emerging fire incidents.

1.2 Problem Statement (PS)

In areas prone to forest fires, campsite visitors, along with security personnel and the fire department on-site, face challenges in staying informed about potential fire incidents. Visitors camping near the forest are concerned about the proximity of their campsite to any nearby burning areas. However, they may not always have access to real-time information regarding fire incidents, particularly if they are out exploring or engaged in activities away from the campsite. Additionally, campsite visitors struggle to obtain current updates on forest smoke level, which are critical indicators of fire risk, especially during bad weather conditions. Without access to such information, campsite visitors may be unaware of emerging fire threats and find it difficult to take necessary precautions or report suspicious activities to authorities. This highlights the need for a robust fire monitoring and alert system tailored to campsite environments, providing visitors with timely updates and alerts about fire incidents and forest smoke level to ensure their safety and facilitate proactive measures to mitigate fire risks during their stay.

Table 1: Summary of Problem Statement

PS	Problem Statement
PS1	Users in forest fire-prone areas lack real-time alerts and updates about fire incidents.
PS2	Lack of awareness about ongoing forest fire leaves users uninformed and unprepared for potential threats
PS3	Users face challenges to get information about forest fires and smoke level because don’t have proper application.

1.3 Project Question (PQ)

The Project Research Question is used to identify the question of existing forest fire detection. Based on the research, we can conclude that there are a few weaknesses of the current forest fire detection. Table 2 shows the summary of the project research question.

Table 2: Summary of Project Question

PS	PQ	Project Question
PS1	PQ1	How can users receive real-time alerts and updates about forest fire incidents through an application?
PS2	PQ2	What methods can be employed to provide users with accurate information regarding forest fire and smoke level?
PS3	PQ3	How is the acceptance of Forest fire system implemented with IoT application

1.4 Project Objective (PO)

Project objective defines the improvement that want to achieve at the end of the project. The improvement must be considered based on the problem statement and the project question of this project. The objectives for this project are shown in below.

Table 3 : Summary of Project Objectives

PS	**PQ**	**PO**	**Project Objective**
PS1	PQ1	PO1	To study and design existing methods of delivering real-time alerts andupdates about forest fire incidents via an application.
		PO2	To develop an application of providing users withaccurate information such as forest fire smoke level.
		PO3	To test features of Smart forest fire detection applications.

1.5 Project Research Hypothesis

A research hypothesis is a statement created by researchers to improve the outcome of a research. Based on the research, the current Smart Forest Fire Detection has insufficient features and is not very satisfying. Some of the hypothesis have been suggested to improve the forest fire detection. Figure 1 shows the problem of the current Smart Forest Fire Detection and the hypothesis to improve.

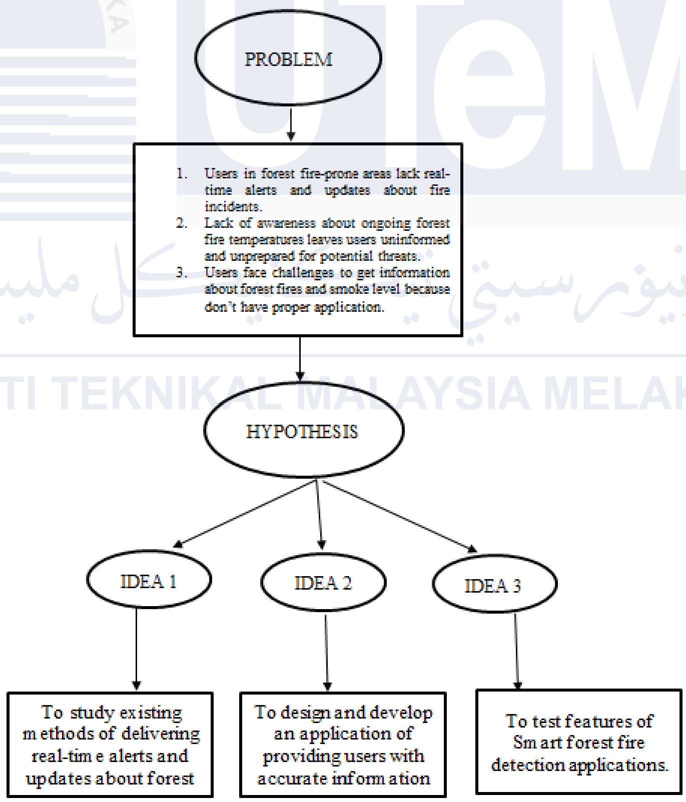

Figure 1: Project Research Hypothesis

1.6 Project Scope

The main purpose of Smart Forest Fire Detection is to create a smart and affordable forest fire detection system using advanced technology. We will focus on improving the current fire safety method which takes too long to respond to. The system will include special alarms for heat and smoke, and we will use the Internet of Things (IoT) to make it smart. IoT forest fire detection is one of the fire detectors that users can connect to for example users can get the fire alert instantly when the system detects the abnormal smoke of a potential forest fire. Then, user can see smoke level graph and fire level graph for every day. Finally, the user can see the history report of fire incidents that have happened before.

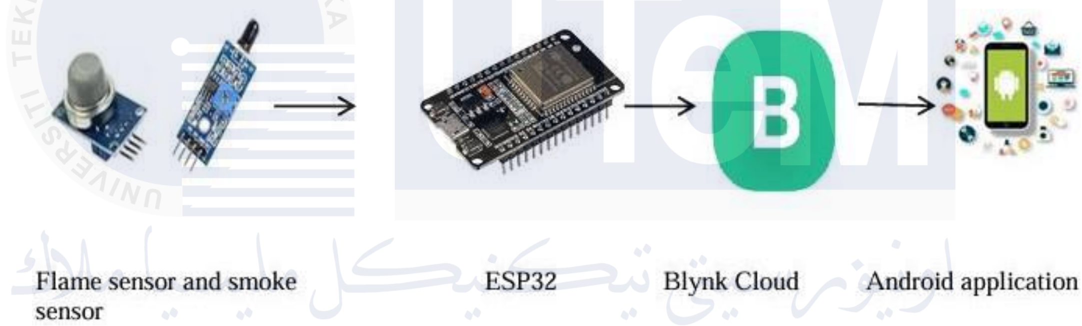

Figure 2: Flow of the system

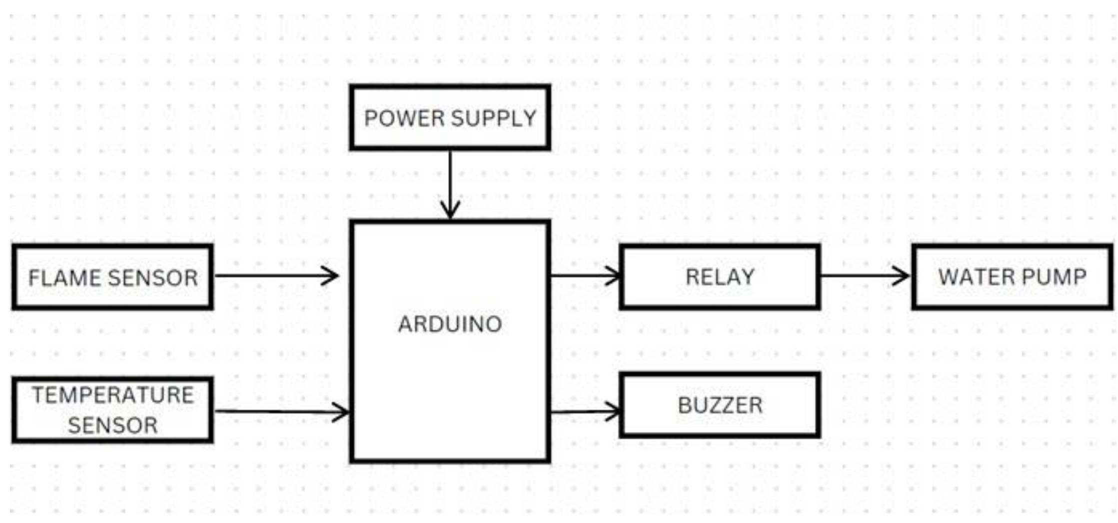

Figure 3: Block of the diagram

Figure 4: Suggested interface for Android Application

1.7 Project Contribution (PC)

Project contribution defines the expected output from this project. This part can be referred to the objectives of this project. The project contribution can be referred to the table below.

Table 3 : Table of Project Contribution

PS	PQ	PO	PC	Project Contribution
PS1	PQ1	PO1	PC1	Proposed an Android application using Blynk to view and connect to the Smart Forest Fire Detection by using a smartphone.
PS2	PQ2	PO2	PC2	Proposed a hardware system using ESP32 that can connect to Android applications and perform Smart Forest Fire Detection.

1.8 CONCLUSION

In conclusion, this chapter shows the project background, problem statement, project objectives, project research hypothesis, project scope, and project contribution for the Smart Forest Fire Detection project. The project background highlighted real-time alerts and updates regarding forest fire incidents and also the challenges faced by the users due to the lack of accessible and reliable information. The objectives of the project were outlined to develop an Android application that will be user-friendly, provide real-time alerts, forest fire smoke level updates, and facilitate reporting of suspicious activities. Improvements to this system have followed the rules suggested this project can satisfy the objectives and follow the phase-by-phase of the project. Other than that, by taking the strategy and scope of the project, this project will be more effective and ensuring the safety and well-being of communities threatened by forest fires.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter is discuss about the problem and solution of the existing forest fire detection, to have a better understanding about the concept and technique need to be implement in this project.

In this chapter, it contain the related publish information and material or article, previous project finding and research related to my objective to my project. Other than that, in this chapter it will compare the hardware and tools that can use in this project. This will show the combination of components that comprise an Internet of Things forest fire detection system. The forest fire detection system will provides accurate and timely information regarding potential fire detection.

The purpose of this project is to focus on the hardware and software development which are interest with ESP32 and an Android smartphone.

2.2 Research problem

i. Concept

The importance of developing IoT system that are secure, reliable and sustainable are importance in unlocking new opportunities for create the new innovation in various industries. According to (M.Kavitha, 2023), there are various type of object in the devices that include intelligence and communication capabilities such as sensors, cars and smartphone. Therefore, device are connected to humans as well as to each other and IoT was brought to light. Based on (M.Kavitha, 2023) the goals for smart forest fire is to include detecting and monitoring forest fires and controlling them before they break out. Other than that, to inform the forest department about fire breakouts as early as possible in order to minimize the effect of the breakout. We need IoT technology to meet the objective.

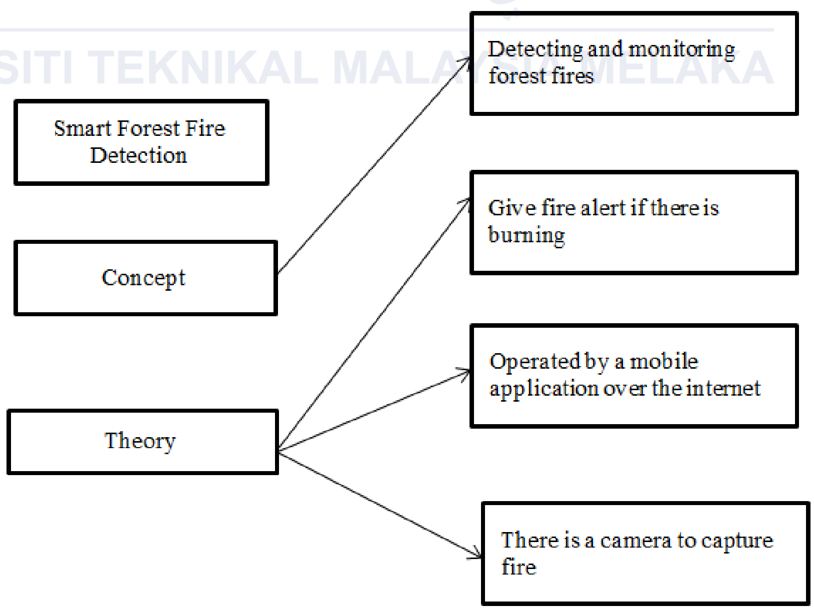

Figure 5: Summary of research problem

ii. Theory

The function of the smart forest fire detection is to help visitor/ user, guard and fire department to get the alert of fire if there is burning in the forest area. This system can be connected to a smart phone and the visitor or guard can easily watch the activity at the forest. Through the (Pamarthi Kanakaraja, 2019), IoT is a network that helps to connect to all physical devices to internet and to communicate with each other and share data.

iii. Application

There are many types of Smart Fire detection being introduced in the market. The most common and famous are :

1. IoT based forest fire detection and protection system(M.Kavitha, 2023).
2. Advances forest fire prediction and detection using arduino by IoT (Shreya soni, 2021)
3. Fire safety and alert system using arduino sensors IoT Integration (Fernandino S. Perilla,2021)
4. Fire monitoring and prevention system based on the severity of fire using nodeMCU and cloud(Madan Lan Saini, 2024)
5. Forest fireplace detection and caution system for disaster prevention(Ye Naing, 2023)
6. Forest fire using internet of things (Vidhi kumara,2022)
7. IoT based anti-poaching and fire alarm system for forest (Ishitha S.,2021)
8. IoT enabled forest fire detection and early warning system (A.Divya, 2019)
9. IoT based forest fire prediction framework (Banu priya prathaban, 2023)
10. Forest fire detetction using LoRa by extension of IoT (Mrs.s.sandhya,2020)

2.3 Related Work/ Previous Work

Following a through and in-depth search by the researchers, this section describes the related work, prior work and studies. In order to complete this project and completely comprehend and fundamental concept, it will explains and present the goal of each tool, concept, and theoretical framework. Then, it will identifies how the project will assist each research project.

M. Kavitha (2023),in their research IoT based forest fire detection and protection system has proposed architecture to detect forest fires, as depicted in the figure. This system uses temperature, humidity, PIR flame sensor, and ESP8266 sensors to continuously monitor the surrounding environment. A buzzer sounds to alert the authorities to the presence of a fire, and a Wi-Fi module notifies them of the incident. After that, an automatic preventive system steps in and sends a water-pumping motor to the impacted area. The architecture of the system, as shown in Fig. 1, incorporates Arduino nano for monitoring and control, ensuring effective data processing and gathering. The system detects something, sets off an alarm, and uses a Wi-Fi module to notify the police. In addition, a preventive measure is started that uses a water-pumping engine to reduce the chance of a fire spreading. The outcomes of this study reveal that the proposed Internet of Things -based fire detection and prevention system creates a reliable and economical means of safeguarding forest areas from fire dangers. The system's fire-related harm risks are reduced, and sensor technologies with Arduino mini's control allow for timely intervention. Furthermore, future research efforts to enhance the system's utility in forest fire control should focus on increasing scalability and adaptability in various environmental environments.

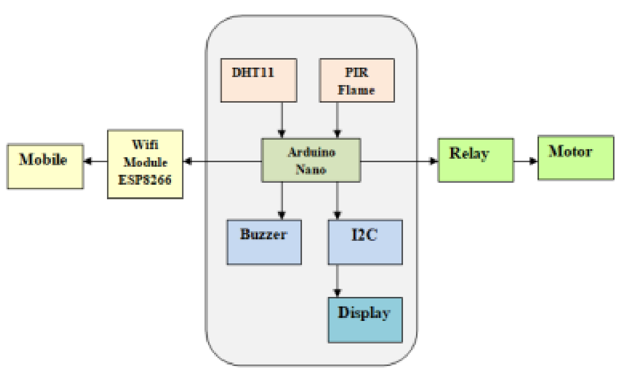

Figure 6: Proposed architecture to detect forest fire

Shreya soni (2021) , in their research Advanced forest fire prediction and detection using arduino UNO by IoT. The specific highlight of this research is an IoT-based solution intended to improve ways to treat forest fires. The technology presented in this work is based on ensuring optimal means for early detection and rapid response to fire outbreaks in forests. As such, by ensuring the minimal damage caused by fires and their efficient management, the workplace seeks to identify its importance in life. The proposed system includes a fire and motion sensor that will be operational in various forest locations. The sensor transmits data to the monitoring system, which helps analyze and raise the alert in case of a fire. A camera system fitted with the sensor helps capture the video, which would be important in forming the cloud and targeting fire causes. It will signal the users on the cloud-based application automatically using the local buzzer. Alerts are conveyed through a buzzer, ensuring timely response even when users are not actively monitoring the system.

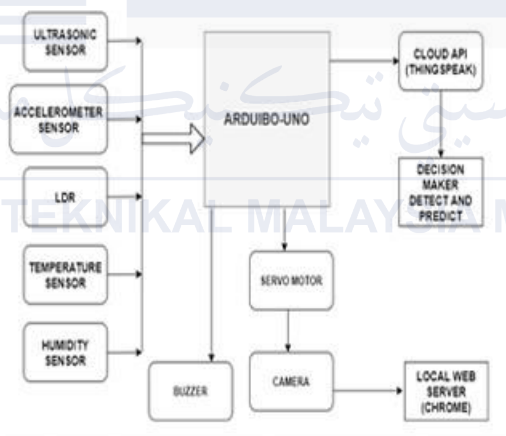

Figure 7:Block diagram of proposed system

Fernandino S. Perilla in their research of Fire safety and alert system using arduino sensors IoT Integration, (2021). The figure of fire safety and alert system using arduino sensors with IoT integration architecture design**.** It includes the following sensors flame detected, Keefe is a DH11 sensor which can record temperature and humidity, MQ-2 sensor are used to record gas and smoke, and a PIR motion sensor attached to the DH11 sensor for extra security. The system architecture is composed of a fire safety and alert system prototype that, once predetermined predetermined sensor values reach the desired threshold, alerts a resident and fire-fighting facility via SMS. It integrates smoke, flame, motion, and gas sensors, along with LED lights, a buzzer, an LCD panel, GPS, SMS modules, and relays, all programmed into an Arduino microcontroller for effective fire safety. Additional components include an Arduino GSM shield 2 module, a GPS module, a buzzer, LED lights, an LCD display panel, and relays. Threshold values for these components are programmed into the Arduino microcontroller, which processes sensor inputs and sends alerts to occupants and fire-fighting facilities. Then, Data collected by the sensors and components are transmitted to a server using the Arduino GSM Shield 2 module for future research and analysis. Relays, LED lights, an LCD display panel, a buzzer, an Arduino GSM shield 2 module, and a GPS module are additional parts. The Arduino microcontroller, which interprets sensor inputs and transmits alarms to residents and fire-fighting facilities, is programmed with threshold values for various components. Subsequently, the Arduino GSM Shield 2 module is used to transfer the data gathered by the sensors and parts to a server for future research and evaluation.

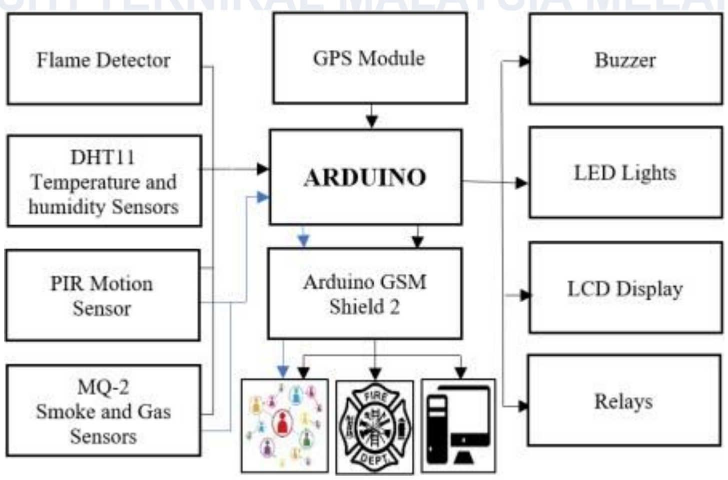

Figure 8:Fire Safety and Alert System Using Arduino Sensors with IoT Integration Architecture

Madan Lan Saini(2024), the research is fire monitoring and prevention system based on the severity of fire using nodeMCU and cloud. The suggested system, shown in Fig. 4, is centred on sensor-actuator-based building fire detection and control. By fixing their flaws and boosting early fire detection while reducing false alarms, it seeks to improve upon current fire alarm systems.The system uses a variety of sensors, including smoke, flame, CO, LM35 (temperature), LDR (light), and CO2 sensors, to provide quick fire detection. These sensors aid in determining the strength of a fire; the thermal sensor, which detects differences in heat, is usually put on walls or ceilings. Every 100 milliseconds, the microcontroller NodeMCU gathers data from all sensors and uses that information to activate actuators in the event of a fire. To calculate the fee of fire outbreak, the gadget evaluates records from the LDR and warmth sensor. Only the water sprinkler turns on inside the case that a flame is detected and all other sensor values are beneath the required limits. The price at which fires get away affects how preventive actions must be taken. Measuring absolute flame depth isn't the only step in identifying possible fire risks. Accurate hearth detection depends on using sensors that degree adjustments in warmth rate, consisting of heat, LDR, CO, and CO2 sensors. As a result, the gadget places a sturdy emphasis on monitoring warmness alternate price as a important signal of a hearth beginning. Although this device makes use of the LM35 heat sensor, other sensors would possibly work nicely in similar situations.

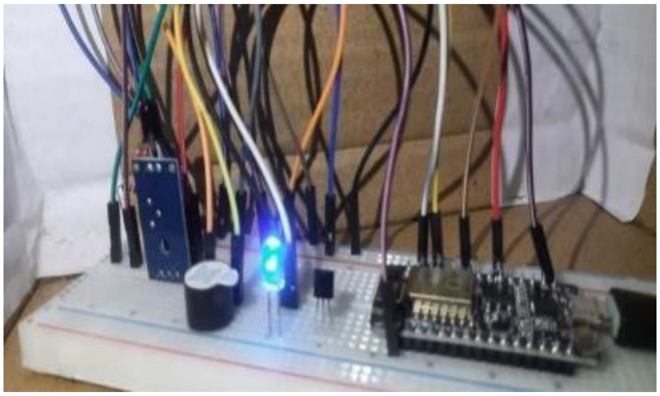

Figure 9:Fire detection and control system

Ye Naing(2023), based on the studies, Forest fireplace detection and caution system for disaster prevention. In this machine structure includes numerous additives working together to detect and reply to forest fires. Firstly, Smoke Sensor (MG-811), It will stumble on the excessive stage of Carbon Dioxide(c02) within the woodland place. Then it could be used for air purity measurement and indoor air monitoring. Then, the climate sensor (WS800), it will Measures numerous environmental parameters consisting of atmospheric temperature, rainfall, daylight availability, and wind pace. Then, Provides professional-grade measurements with a virtual interface and ready with air flow and radiation safety for accurate readings. Flame sensor(LM393) to locate the presence of flames, generally used in fire alarm systems and may locate flickering flames usually associated with fires. In this gadget additionally have temperature Sensor (DHT11) to Provides virtual output for moisture and temperature ranges and well matched with microcontrollers like Arduino and Raspberry Pi. Lastly, Arduino ATmega 2560 to Microcontroller board based at the ATmega2560 chipset, Features numerous digital and analog pins, UARTs, and a sixteen MHz crystal oscillator and Provides a platform for connecting and controlling the sensors, processing facts, and triggering responses.

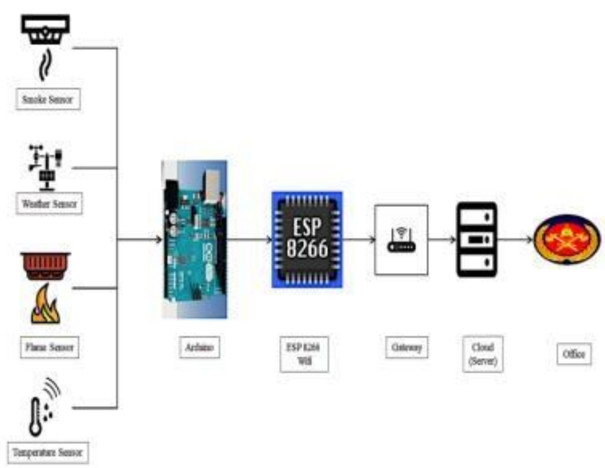

Figure 10: System Architecture

Vidhi kumari(2022), a research for early detection of forest fire using internet of things. Based on the research, the have hardware, communication tools and webpage. The project has a NodeMCU board. This includes an ESP8266 Wi-Fi module and micro USB ports that allow connections. A DHT11 sensor is there, which can measure temperature and humidity levels. Then there's an MQ2 smoke sensor capable of detecting gases like carbon dioxide by sensing changes in resistance. Furthermore the Flame Sensor (YG1006) uses LED signals to indicate flames within a wavelength range and a Resistive Soil Moisture Sensor measures soil moisture levels based on variations in resistance. Communication tools such as the Wi Fi module for connectivity and ThingSpeak, an open source API platform, for IoT data storage and retrieval are utilized. By using HTML, CSS and Bootstrap the systems webpage interacts with the ThingSpeak cloud database to offer real time updates on forest conditions to authorized users.

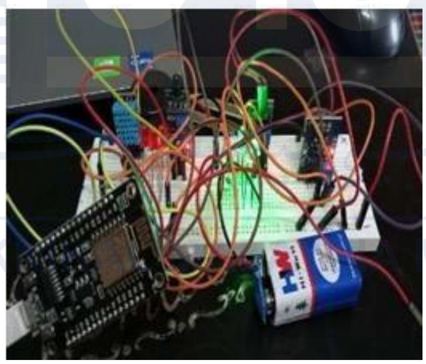

Figure 11: The hardware set-up of the forest fire monitoring system

Ishitha S.(2021), research for IoT based anti-poaching and fire alarm system for forest. There are using flowcharts, activity diagrams, and sequence diagrams, this part clarifies on the system design. The proposed IAFS (anti-poaching and fire alarm system) flowchart, which delineates the order of actions, is depicted in Figure 11. First, the process is started, and then each of the three sensors is turned on. After that, the system assesses the temperature, smoke and laser detection levels. In light of these circumstances, it either gathers and presents data in the cloud or, in the event of an alarm, notifies the relevant officer before ending the procedure. An activity diagram that dynamically illustrates the operation of the system is shown in Figure 12. The gadget starts reading input from the

LDR, temperature, and smoke sensors as soon as it is turned on. As long as the temperature stays below the threshold, it checks the status of smoke and LDR sensors; otherwise, it sends an alert notification.

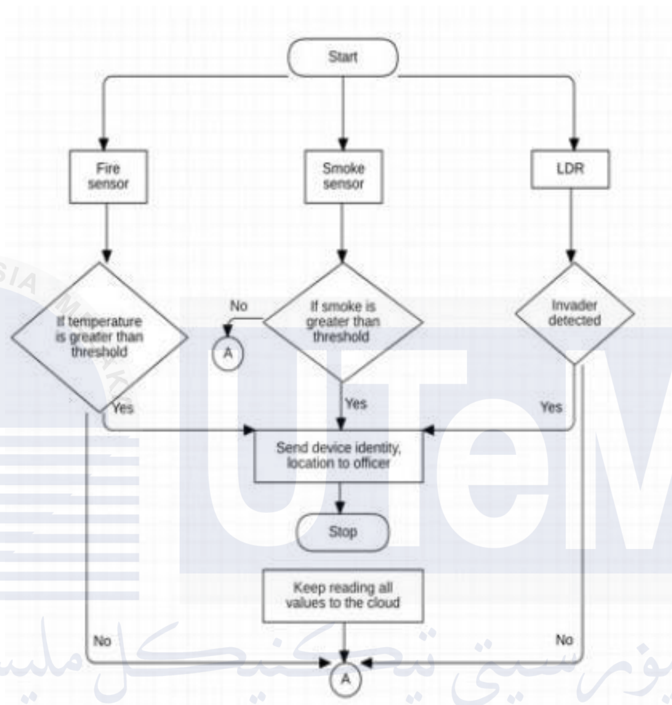

Figure 12: Flowchart of the proposed IAFS

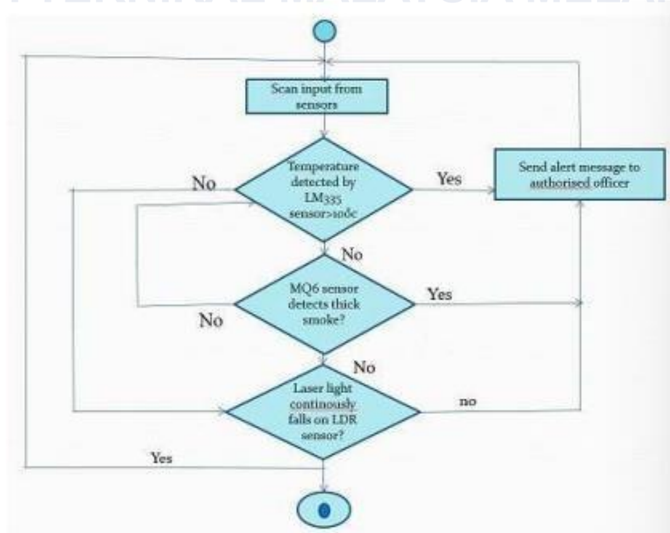

Figure 13: Activity diagram of the proposed IAFS

A.Divya(2019), based on research IoT enabled forest fire detection and early warning system. In this part of the research, it is recommended to use WSN (Wireless Sensor Network) for an improved fire detection system in the early stage by planting wireless sensors at strategic positions. The equipment consists of different sensors that are joined by wireless means with a small satellite to transmit data and then delivered to a ground station where it is analyzed. With the use of WSN, the system is expected to help detect fire threats early. Temperature and smoke sensors have been placed strategically across the forest so that they can monitor CO2 and temperature levels. A group of sensors directs the signals to a microcontroller and analyzes and responds to the changes in the environment; this is particularly helpful if there is a case that requires immediate action. The advantages provided by the system include prompt response, single installation, and workers' environment monitoring on a permanent basis. The sensors used in the device employ electrodes and electrolyte, which are put into a plastic case in order to prevent any gas leaks and electrical contact. The principle of the operation is based on an electrochemical reaction taking place between gas entering the sensor and generating measurable output. Then, the Internet of Things (IoT) feature can get the data transmission through GPRS modem SIM900, enabling an internet connection for users to remotely monitor data.

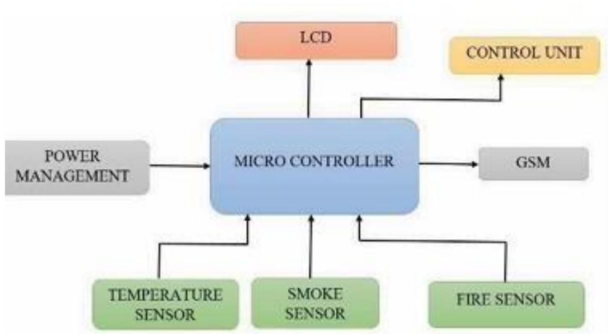

Figure 14:Proposed scheme

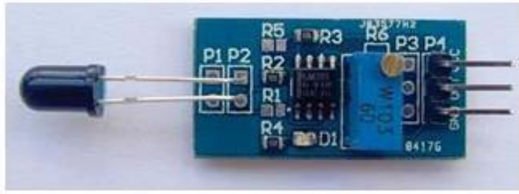

Figure 15 : Fire sensor

Figure 16: IoT board

Banu priya prathaban(2023), Based on research IoT based forest fire prediction framework. In this research, there are the result and discussions. The country experiences a wide range of temperatures, with daily averages fluctuating betwixt 30°C 22°F and 30°C 86°F , and an annual mean of 10°C 50°F . The area was characterized by hot summers and cold winters. The proposed commercial utilizes the Arduino IDE platform. If the demodulator outputs from the three designated sensors proceed below the predetermined threshold and indicates the absence of an afforest fire. These demodulator readings was transmitted in real time to an exchange base station, enabling successive monitoring of the afforest environment. Any deflection from the convening door eruptive triggers a warning about effectiveness fire hazards using Boa technology. The ESP8266 Wifi allows the data to be stored in the cloud for approach when needed. Figure illustrates the transcription is alive stipulation prior to fire detection. Then, in case of a forest fire, the LoRa system will transmit a signal to the receiver if the sensor readings surpass the usual or predefined thresholds at a particular moment. Additionally, the GSM module will dispatch a text message to the designated mobile number, providing details regarding the location of the forest fire.

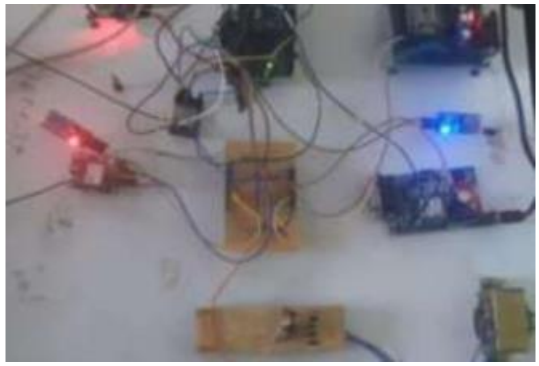

Figure 17:Active system before detection of fire

Figure 18:Active system after detection of fire

Pankaj Garg(2023), based on research An integrated IoT solution for forest fire detection. Installing a fire alarm system is an important safety precaution to prevent the spread of fire and save lives. The three critical parts of the system—communication, computation, and prediction, and sensing and navigation—work together to effectively detect and respond to fire information, collect data and then transfer it to the system for further analysis. For example, smoke detectors use light sensors to detect smoke particles and trigger an alarm as soon as they are detected. Once sensor data is collected, the calculation and prediction phase comes into play, where sophisticated algorithms analyze collected data to determine the presence of fire Using parameters such as humidity, temperature, CO_2 concentration , the system indicates that these values exceed presset thresholds or Facilitates dissemination of alerts to authorities or concerned individuals by means such as email, text message, or telephone Fire alarm systems can be used if it has spacious rooms attached to a central checkpoint, to allow emergency personnel to be dispatched to a fire. By monitoring temperature, humidity, and CO_2 levels in the atmosphere, the system enables rapid intervention, reduces damage, and protects the environment, resources, and wildlife from forests fire mitigation Sensor technology, forecasting systems, communication systems, and forest fire detection. And by conducting scholarly research on techniques, researchers help increase the effectiveness and reliability of fire alarm systems in urban and natural settings.

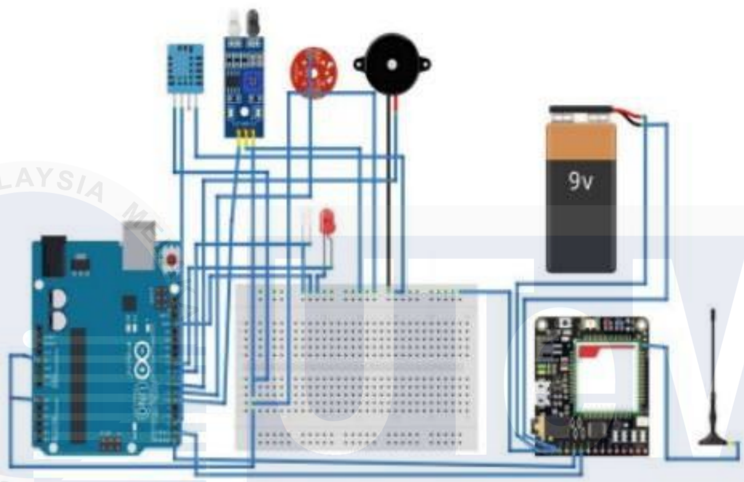

Figure 19:Architecture design (circuit diagram)

2.4 Critical review of current problem and justification

Inspired by past research, forest fire detection systems work just like the tools we use every day. They automatically monitor forests for signs of fire, using sensors and smart algorithms. Authorities are alerted when unusual conditions are detected that could indicate a fire. These systems operate without human supervision at all times, helping to detect fires quickly and minimize damage. They are like automatic guards in the forests, using technology to keep them safe.

From the related work that being study, there are many types of development and methods being used. As a conclusion, a summary has been done as shown in table 2.1.

Table 4: Summary of critical review

Research Title	Purpose	Description	Advantage
IoT based forest fire	The purpose of this	In this system, there	The advantage of this

detection and protection Author: M.Kavitha	project is to provide timely alerts to authorities and initiates preventive measures to mitigate the spread of fires, ultimately safeguarding forest areas from fire dangers.	are sensor to send notification of the fire incident. Other than that, there are an automatic preventive system is activated, deploying a water-pumping motor to the impacted area to reduce the chance of the fire spreading.	system is automatic preventive system which is deploys a water-pumping motor to the affected area, reducing the spread of the fire. This comprehensive approach not only minimizes fire risks but also provides a cost-effective solution for safeguarding forests.
Advanced forest fire prediction and detection using arduino UNO by IoT Author: Shreya soni	This project purpose is to tackle the problem of wildfires and use IoT technology to create a system that detect fires quickly.	In this system, it includes a camera that record video footage during fire incidents and also enable authorities to receive immediate alerts when a fire is detected.	This system have a camera that can control by a servo motor for 360 degree rotation to detect object that may have caused the fire.
The figure of fire safety and alert system using arduino sensor with IoT integration architecture design Author:	This system purpose is to create a fire safety and alert system that can notify residents and fire department via SMS when sensor detect fire burning.	All of the sensor will connected to an Arduino microcontroller. If sensor indicate abnormal situations, the microcontroller will triggers alert. It	This system have data collection to allow for improve fire safety procedures and system.

Fernandino S. Perilla		may activate alarms, lights, sprinkler systems, and send alerts via SMS and system logs all activities then sends data for further analysis.	
Fire monitoring and prevention system based on the severity of fire using nodeMCU and cloud. Author : Madan Lan Saini	This system will provide timely warnings and observations to reduce the risk of fire-related dangers.	All of the sensor in this system will connected to a microcontroller board called NodeMcu. When these sensor detect fire, microcontroller will triggers responses including alert notifications sent via Blynk cloud service and SMS to fire department station.	The advantage of this system is quickly detect fires and the system will activate the fire suppression system, turns off main power supply, and activates water pumps and fire extinguishers in case of fire detection. The system's automation reduces the risk of human error, ensuring efficient and effective fire detection and control.
Forest fireplace detection and caution system for disaster prevention. Author:	The purpose of the forest fire detection and warning system is to prevent disasters by detecting and responding to forest	In this system, there are environmental parameters such as atmospheric temperature, rainfall, sunlight	The system is compatible with popular microcontroller platforms like Arduino and

Ye Naing	fires in a timely manner. The system incorporates various components that work together to achieve this goal.	availability, and wind speed. It provides professional-grade measurements with a digital interface and is equipped with ventilation and radiation protection for accurate readings	Raspberry Pi, offering flexibility in implementation and customization. Then, the system provides a cost-effective solution for forest fire detection and warning, making it accessible for deployment in various settings.
Early detection of forest fire using internet of things. Author: Vidhi kumari	The purpose of this system to develop low-cost IoT-based device to safeguard wildlife, herbal plants, trees, and tribal communities by preventing forest fires through early detection. The project aims to enhance the prediction, detection, and control of forest fires.	A NodeMCU embedded system transmits real-time data to a dedicated Forest Official Webpage every 15 seconds. If any abnormal readings indicative of environmental conditions conducive to a forest fire are detected by the sensor nodes, immediate alerts are triggered on the webpage. As example, the	This system's ability to provide accurate fire location information allows for more efficient allocation of firefighting resources, ultimately improving overall response effectiveness and minimizing the impact of forest fires

		detection of flames activates a red LED signal, while the presence of smoke activates a yellow LED signal. Forest officials can analyze the information displayed on the webpage.	
IoT based anti-poaching and fire alarm system for forest. Author: Ishitha S.	The purpose of this system is to combat illegal activities such as tree smuggling and animal poaching while also preventing forest fires. This system aims to protect valuable trees like teak and sandalwood, as well as endangered animals such as tigers and rhinos, by utilizing remote sensor modules integrated into an Arduino board.	If sensor readings exceed predefined thresholds, indicating potential threats like fire or intrusion, alert messages are sent to authorized officers via SMS gateway. The system's ability to detect changes in temperature, smoke levels, and light enables it to identify and respond to forest fires, illegal logging, and animal poaching incidents.	The advantage of this project is cost-effectiveness and efficiency in detecting intruders and forest fires. By employing sensors such as temperature, smoke, and LDR, along with a Wi-Fi module and cloud-based platform, the system can continuously monitor forest conditions in real-time.
IoT enabled forest fire detection and early warning	The purpose of this system is using the wireless sensor network	This system use data wirelessly to a satellite that can	This system require minimal upkeep and it can monitored

system Author: A.divya	which is WSNs to detect forest fire early and can spot changes in temperature and carbon dioxide levels to prevent fire.	relays the information to a ground station for analysis. If there have fire, this system will give the reaction immediately.	remotely, improved safety for workers and reducing costs.
IoT based forest fire prediction framework Author : Banu priya prathaban	The purpose of this system architecture is to design a comprehensive solution for early detection and management of forest fires using a combination of sensors, microcontrollers, and communication modules.	In this system, they uses two 12v adapters for power. They also have sensor to collect data on temperature, humidity and gas levels. If there have a fire, the system send alerts to nearby stations via GSM and LoRa modules.	This system offers early fire detection, long-range communication, and remote monitoring capabilities. It ensures reliable performance throughout the year and accounts for sensor battery life
An integrated IoT solution for forest fire detection Author: Pankaj Garg	This project aims to leverage IoT technology to detect forest fires in their early stages, enabling timely response and effective mitigation strategies.	The components in this system such as humidity, smoke, and flame sensors, along with microcontrollers like Arduino UNO and communication modules like SIM 808 GSM/GPRS module will gather	In this system, it provides real-time monitoring of environmental parameters, enhancing the effectiveness of fire detection and response efforts. Additionally, the system's integration with cloud platforms

		data from the forest environment and analyze it to detect signs of fire. When fire detected, the system send alerts and the data will uploaded to the cloud for further analysis.	enables data visualization and analysis, facilitating informed decision-making by authorities.

2.5 Proposed solution/further project

From the related project that being study, there are many types of development and methods being used. As a conclusion for the different type of the smart forest fire detection system, there are three types of system that have been chosen to build my own system.

There are three type of different system which is Fire monitoring and prevention system based on the severity of fire using nodeMCU and cloud (Madan Lan Saini , 2024). Other than that, the second one is early detection of forest fire using IoT (vidhi kumari , 2022) . The third one is IoT based anti-poaching and fire alarm system for forest (Ishitha S, 2021).

Based on financial standpoint, there are all inexpensive and easy to find the items. The suggested approach makes use of cutting-edge sensor technology, data analysis methods, and communication networks to allow for real-time monitoring and quick reaction in the event of a forest fire. Then, this system use WiFi access enables the security or fire department to link to the internet and utilise a smartphone app for communication with the guest. By employing sensors such as temperature, smoke, and LDR, along with a Wi-Fi module and cloud-based platform, the system can continuously monitor forest conditions in real-time (Ishitha.s, 2021).

Table 5:Comparison between the system

Title	**Fire monitoring and prevention system based on the severity of fire using nodeMCU and cloud**	**Early detection of forest fire using IoT**	**IoT based anti-poaching and fire alarm system for forest**
Cost	Cheap	Cheap	Reasonable
Wifi connectivity	Yes	Yes	Yes
Power source type	Plugged to a socket	Plugged to a battery	Plugged to a socket
Real time data input	Yes	Yes	Yes
Advantage	Anywhere with internet access can monitor and control the system remotely. This allows for status updates, real-time notifications of fire occurrences, and the flexibility to change system parameters or response procedures as necessary.	This system's ability to provide accurate fire location information allows for more efficient allocation of firefighting resources, ultimately improving overall response effectiveness and minimizing the impact of forest fires	Data stored in the cloud makes it simple to access, retrieve, and analyse previous data about fire events or poaching.

To solve the issues with the current system, it is advised that sophisticated forest fire detection be put into place. In this project, we design the software system to monitor the forest fire detection system. The user will be able to check the daily smoke level data in the forest campsite by connecting wirelessly to the programme. In this project, the NodeMCU is utilised to establish an internet connection, and Blynk is used to develop the application. Smoke sensors, flame sensors, power supplies, and buzzers are some of the components that will be used. Data will be transmitted via Firebase (Cloud) so that the user may access it.

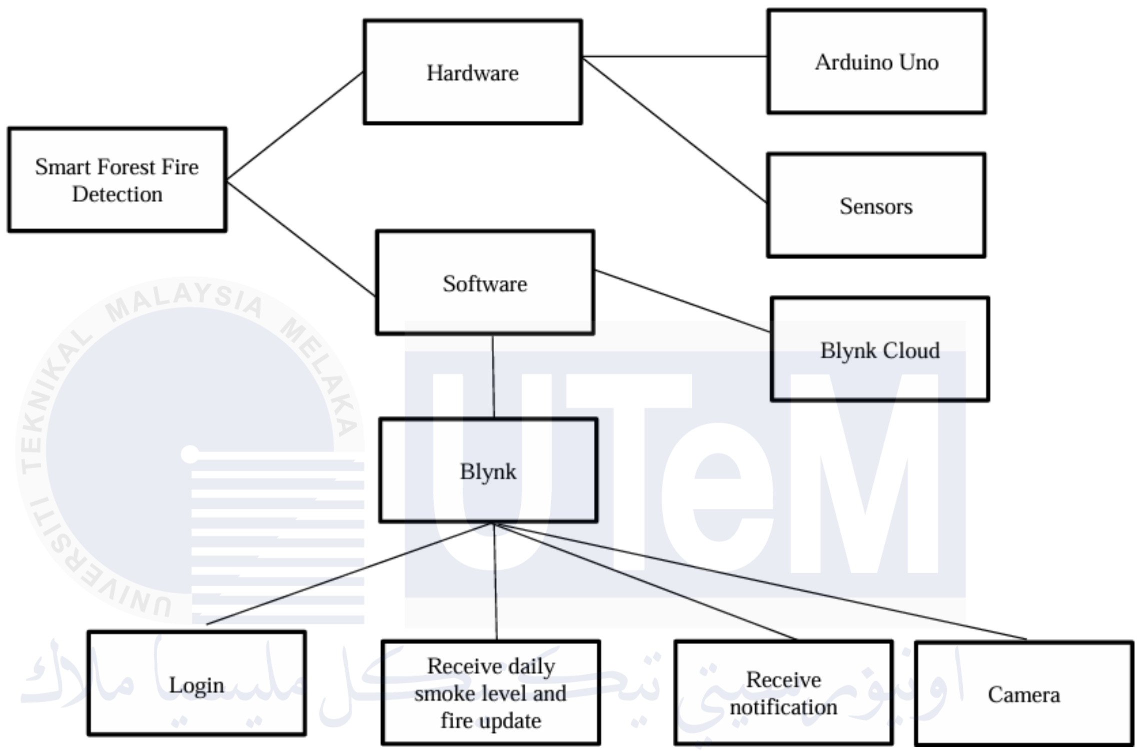

Figure 20: Functionality of Smart Forest Fire Detection system

An efficient and comprehensive method for early detection and warning systems is provided by the suggested forest fire detection sensor solution. The suggested strategy seeks to preserve life and forest for future generations by utilising the latest sensor technologies, data processing methods, and a strong communication network.

2.8 Conclusion

To sum up, the literature study is an essential chapter that plays a critical role in developing the project concept. It provides insight into the system's current features and a clear understanding of how to execute the system. The research and analysis can make the project's progression more understandable and seamless. In short, the goal of this chapter is to enhance the earlier research and enhance the problem's solution.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter discusses the methodology that will be applied in this project. Methodology refers to the systematic and theoretical analysis of methods, practices, and procedures used in a particular field of study or research. The methodology provides researchers with a framework for systematic research and ensures that the research process is rigorous, transparent and repeatable. A number of different methods are used in research and various research fields, including agile, rapid application development (RAD), spiral, waterfall, and prototyping. This project uses the prototype modeling methods. The prototype modelis a software life cycle model (l) that involves creating a functional model (prototype) of a system before developing complete software. The prototype model helps identify and validate user requirements. Creating a working prototype allows users to visualize and interact with the system virtually during the development process.

3.2 Research process

The research process is the continual evolution of the product. The procedures include data collection, analysis, design, implementation, and testing, to name a few. It is necessary to finish the project at every phase. The flow of the research process can be referred to the figure below.

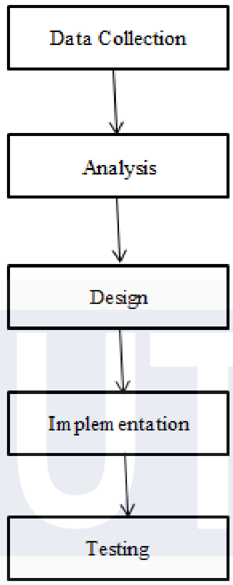

Figure 21 : Flow of research process

3.2.1 Data Collection

Collecting data for a smart forest fire detection system involves gathering relevant information and measurements to understand the conditions that contribute to forest fire detection. the first stage in the data-collecting process for a smart forest fire detection system is to identify relevant data sources. By this project, data is obtained via smoke and flame sensors before being sent to the user.

3.2.2 Analysis

According to the research and online analysis, it has been found that the existing forest fire detection system is not operating optimally. This is because some forest fire detection systems lack any applications for user to access. As a result, they cannot access the most up-to-date information about forest fires, smoke level and others. Then, this is dangerous if the user is around the burning place. This leads to both local user members and outsiders feeling unsafe during poor weather conditions. This inefficiency represents a significant misuse of resources.

3.2.3 Design

a. System Architecture Design

An internet-connected Android app with the ability to identify smoke level, flame, and possible forest fires is expected to come out of the smart forest fire detection system. Smoke and flame sensors linked to the ESP32 in the camp at the forest site will report the smoke level and depending on the data from the flame sensor, trigger an alert if there is a fire. After transmission, the data will be kept on the Blynk cloud. The data and report will be shown on the application's interface.

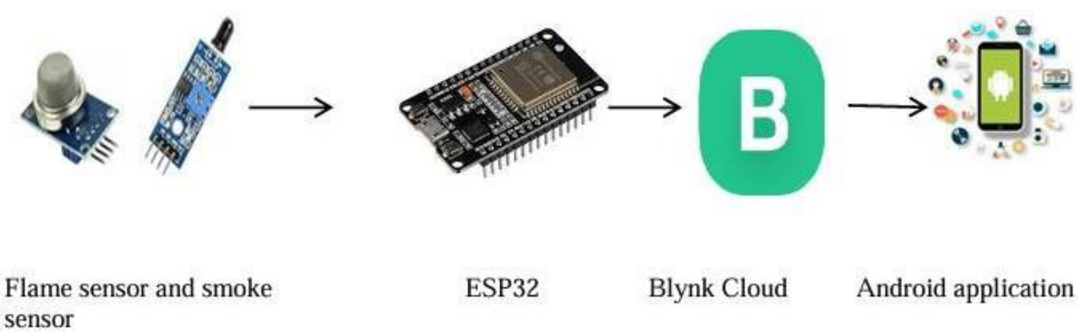

Figure 22: System Architecture

3.2.4 Implementation

Project implementation explains the process of turning the version and plans into a working system. The implementation process consists of two stages: the Android Studio application and the hardware connection. Chapter 5 discusses all the implementation specifics.

a. Hardware connection

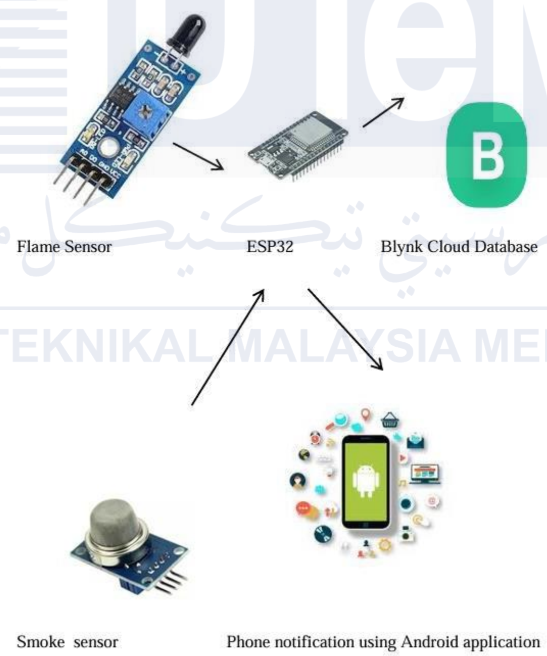

Figure 23: Hardware connection

The hardware connections used in this project are shown in the above diagram. Through the use of a male to female jumper wire, the sensor is connected to the ESP32 to enable data retrieval. It is necessary to programme the Blynk application to ensure that the ESP32 connection is established. To process the data, all of the information that the ESP32 collects must be sent to Blynk cloud.

b. Blynk application

Blynk is an application integrated development environment to monitor the Forest Fire Detection IOT system.

3.2.5 Testing

Testing is the process of running a system application and ensuring that everything is functioning as it should. Hardware testing and software application testing are the two phases that have received the most focus in the testing procedure for the Smart Forest Fire System. Every step of the testing process has to stick to the proper protocol. In chapter 6, a detailed explanation is covered.

Hardware testing: The ESP32 and sensors are included in hardware testing.

Software application testing: Every developed system feature is looked at to make sure everything is functioning correctly.

3.3 Prototyping Methodology

An early version known as a prototype gives stakeholders an actual picture of the anticipated final product so they can see it and engage with it prior to complete development. The objectives and specifications of the project determine a prototype's degree of usefulness and fidelity. Prototyping contributes to a development project's increased overall success by lowering risks and fostering better collaboration.

For a number of reasons, a smart forest fire detection project may find the prototype model suitable. Understanding the specific needs and specifications of the users, such as the kind of data to be gathered, the level of accuracy needed, and the ideal response time, is essential to the development of a smart forest fire detection system project. End users and domain experts are important sources of input on the efficacy of a landslip detection system. The prototyping project consists of six stages: requirement gathering, quick design, building prototype, customer evaluation, refining prototype, and engineer product. Figure displays the model of prototyping.

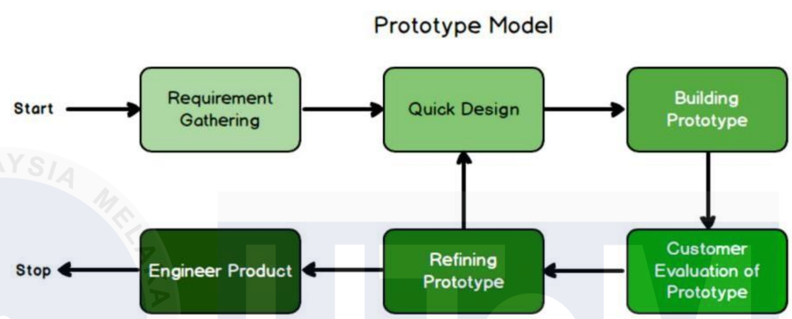

Figure 24:Prototyping Model

3.3.1 Requirement Gathering

As part of the prototyping methodology, gathering project requirements is the first step. Hardware and software needs are the two particulars that are absolutely necessary. These needs will be explained in detail in the sections that follow.

3.3.1.1 Hardware Requirement

i. ESP32 32

ii. Light Dependent Resistor

iii. Flame sensor

iv. Smoke sensor

v. Buzzer

vi. Breadboard

vii. Jumper Wire

viii. Resistor 330hm

ix. 12C LCD Display 16x2

x. Battery 9v and battery connector

For hardware connection, Light dependent resistor, flame sensor, smoke sensor and buzzer will connect to a Wi-Fi module, while the Wi-Fi module will connect to an NodeMCU board. The 330 Ohm resistor will be used to limit the current flow through the LCDs, while the 9V battery and battery connector will be used to power on the microcontroller without requiring a USB cable connection. All of the connection components for this project will be wired using breadboard and jumper wire.

3.3.1.2 Software Requirement

i. Arduino IDE

ii. Blynk

iii. Blynk Cloud

All of the hardware that will be used to accomplish the software described above will be programmed using the Arduino IDE. The creation of an Android application using Wi-Fi connectivity to retrieve smoke and data on forest fires from a Wi-Fi module will begin with Blynk. The Android application will have several interfaces designed, such as those for reporting, notifications, the current state, and risk level information. The capabilities of an IoT system for smart forest fire detection is shown in the image below. To save information on the smoke level and status of forest fires, Blynk will establish a direct connection with blynk cloud.

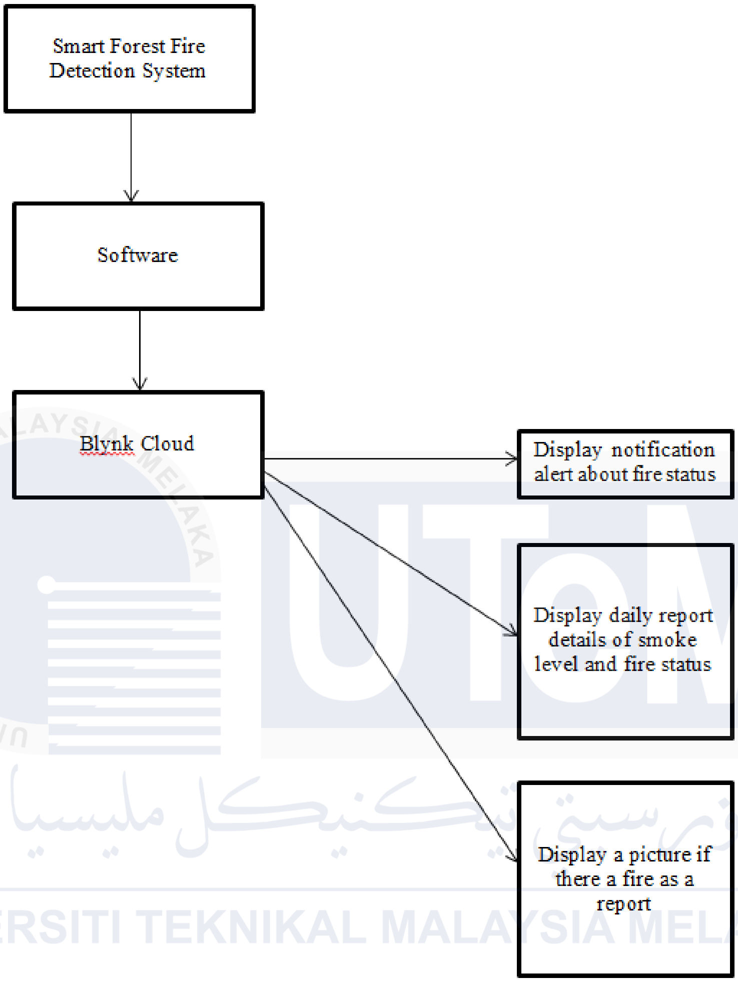

Figure 25: Functionality of Smart Forest Fire System

Login- User who have registered can login to the android application

Register - This process is to create a new account for the new user.

Display notification alert about fire status – Notify the user of forest fire update if it have a abnormal activities. Then, user will be presented with the instruction and required action.

Display daily report details of smoke level and fire status – Report status will generate detail report in user interface.

Display a picture if there a fire as a report – Display the picture at burning place for aware the users not go to that area.

3.3.2 Quick Design

The design of an IoT-enabled smart forest fire detection system will be used in the second phase of the prototyping model to create a concept for the project's execution, along with a flowchart outlining the project's intelligent development. The project's logical and physical components should be included in the design.

3.3.3 Building Prototype

The third stage will involve installing and configuring all hardware in line with the previously executed design. Furthermore, the chosen software will be installed for this project. The Arduino IDE, version 46, is the first and most important piece of software. It is used to write the programming for any hardware component that uses a NodeMCU microcontroller. The construction of Android application interfaces subsequently made use of the installation of Blynk. The purpose is to use a Wi-Fi connection to get the notification status from received data in Arduino and send it to a mobile device. Smoke level and data about forest fires will be stored using Blynk cloud, the third installed programme.

3.3.4 Customer Evaluation

In order to confirm the produced system's advantages and disadvantages, the finished suggested system will be assessed by an assessor and put to the test by many users in the fourth step. They will evaluate the system and application's created functionality and performance. To meet user requirements and ensure satisfaction, performance testing will assess data transfer speed and accuracy. The prototype will be tested for functionality by having the user select the location of the sensors and determine whether the data being collected is successful and functioning as planned, or if there are any issues. Any feedback and recommendations made by an assessor will be gathered, edited, and examined in order to improve the way the project is carried out. To assess and confirm the usefulness of the built smart forest fire detection with IoT system, a questionnaire regarding the aims of the forest fire detection system will be used for system usability testing.

3.4 Project Milestone

This endeavor's implementation plan will be based on the project timetable and milestones. Because they guarantee that the project will be performed properly and within the allocated time frame, including the anticipated completion date, the project timeline and milestones are crucial. It will therefore be beneficial since the project won't get behind schedule. The following figure shows the project's scheduled completion points.

Table 6: Project Milestone

Method/ week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Requirement Gathering																						
Quick Design																						
Building Prototype																						
Customer Evaluation																						
Refining Prototype																						
Engineer Product																						

3.5 Conclusion

To sum up, this chapter, which outlines the project development technique, is crucial since it may control the project's flow structure. This chapter also includes a methodological description. The prototype model utilised for this project is simple enough that anyone will have no trouble understanding and using it. By following the prototype model's instructions, the developer can improve system development and boost client happiness. This is a result of the user's active participation in the project's development.

CHAPTER 4

ANALYSIS AND DESIGN

4.1 Introduction

This chapter presents the results of the investigation phase, both in the developed and developing steps of design. The details are analyzed to identify all project requirements in terms of what is required, both in hardware and software. This chapter involves a detailed discussion of the block diagram architecture with an analysis to ensure that the project design has been done perfectly and robustly. In describing every part of the project so exhaustively, from its foundation to intricate architectural design, we seek to provide a complete understanding.

4.2 Problem Analysis

Since it does not have any applications to give information for users, the functional smart forest fire detection system available in the market is not full. Therefore, the monitoring tool used in this project is Blynk application which is developed to monitor environmental changes and provide proper information for users.

4.3 Requirement Analysis

This section describes the input and output requirements of this undertaking. In addition, it demonstrates all the functional requirements for the complete construction of this undertaking.

4.3.1 Data Requirement

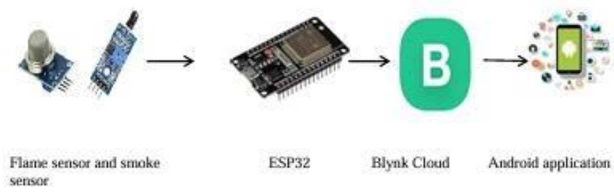

Figure 26: Data Flow

Figure 26 : System Data Flow The above figure illustrates the data flow of a system Overall project abstract A smart forest fire detection system that should end up as an Android app for the internet, that would be able to detect a bushfire by using flame and smoke sensors. These sensors, connected to the ESP32 module will monitor and send smoke level data for regions prone to forest fires, along with fire stability. The above information will be sent and saved into Blynk Cloud database. The application's main interface will showcase the latest reports along with data from the past few days, providing users with timely and historical information on forest fire conditions.

4.3.2 Functional Requirement

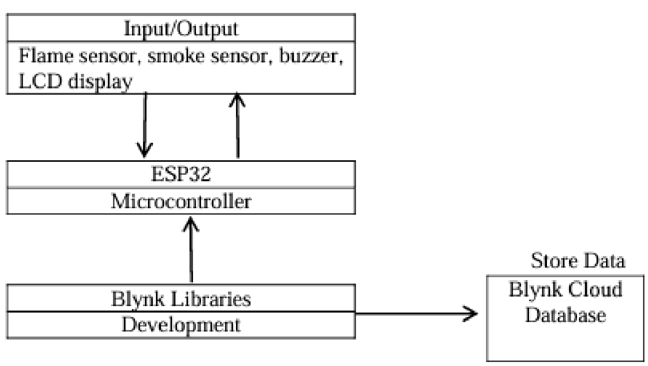

Figure 27: Block diagram of functional requirement

This project is divided into several sections namely microcontroller block, input/output block and development block. The figure 4.2 depicts the details of block diagram for this project.

1. The Blynk Libraries send the command to the NodeMCU microcontroller, which in turn sends that command to the sensors. The microcontroller sensor control code is all IDE programmed.

2. Input/Output sensors will measure smoke and flame in the forest area. It has to be directly connected to this microcontroller ESP32.

3.Blynk libraries : The project integrates the hardware components using the Blynk library for the smart forest fire detection system. This core operation is found within the development block, where Blynk Library controls to the ESP32 microcontroller. This always controlling ESP32 to check if smoke level is too high or there is any flame The Blynk Library which assists in making the Android application communicate with ESP32 microcontroller by sending them commands and receiving responses from them.

4.4 Hardware Requirement

1. ESP32

   ESP32 is a powerful and flexible development board based on the ESP32 microcontroller that has two cores of Tensilica LX6 processor running at 240 MHz. With its 520KB SRAM and up to 4 MB flash memory, it covers a lot of uses. The board works within the voltage range from 2.2V to 3.6V and has built-in Wi-Fi (802.11 b/g/n) and Bluetooth (v4.2 BR/EDR and BLE) capabilities. Besides supporting over 30 GPIO pins for digital input/output, PWM, ADC, DAC, etc., ESP32 also provides various interfaces such as UART, SPI, I2C and I2S for interactions with different sensors & peripherals between. Among other features that it has are a Hall sensor, touch capacitive sensors and integrated smoke sensor. The programming of ESP32 can be carried out using Arduino IDE; this brings in a lot of flexibility for the developers.

Figure 28:ESP32 Microcontroller

2. Flame Sensor

   Flame sensors, which are made to recognise certain light wavelengths released by flames, are a vital component of Internet of Things forest fire monitoring systems. These sensors come in two varieties: infrared (IR) detectors, which are used to detect infrared light, and ultraviolet (UV) sensors, which are used to detect ultraviolet light released by fires. This is made possible by filters that have been added to them to improve flame detection accuracy alone without interfering with ambient light. Thus, when combined with internet of things technology, devices like Raspberry Pis or microcontrollers may readily assess the presence of fire in real-time.

Figure 29:Flame Sensor

3. Smoke sensor

In order to identify forest fires early and stop them from spreading, smoke sensors are vital equipment. These sensors come in a variety of forms, such as optical sensors, which identify smoke particles using light, infrared sensors, which identify smoke and heat, and multi-sensor systems, which integrate many technologies for increased precision. These sensors' key characteristics include their great sensitivity to detect even minute levels of smoke, quick reaction times, and resilience to inclement weather. They frequently form a component of bigger networks that enable distant data processing and monitoring, as well as timely notifications for efficient management and suppression of forest fires.

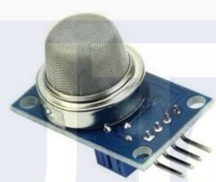

Figure 30:Smoke sensor

4. Buzzer

An electrical gadget known as a piezo buzzer uses the piezoelectric phenomenon to generate sound. Among many other electrical systems and gadgets, it is widely used to deliver audio alerts, notifications, and tones.

Figure 31: Piezo buzzer

5. LCD

The generation of text and images is produced on a flat-panel display which goes by the name LCD (Liquid Crystal Display) using liquid crystals.

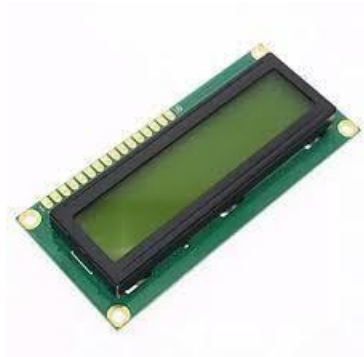

Figure 32: LCD

6. Jumper wire

One way of connecting device (e.g., sensor) to other devices is through the use of jumper wires. It's pretty simple. Jumper wires come in a variety of colors, lengths and types. Essentially there are three types of connector wires: male to male, male to female and female to female. Although all can be used interchangeably for the same purpose, each type is best suited for particular devices.

Figure 33:Male to female jumper wire

7. Resistor

In electronic circuits, resistors play the role of controlling or regulating current flow. They serve multiple purposes such as isolation of transmission lines or voltage division and signal frequency adjustment, and decreasing current flow which make them highly useful components in any circuit design.

Figure 34:Resistor

4.5 Software Requirement

1. Arduino IDE

An intuitive software framework called the Arduino Integrated Development Environment (IDE) was created to make using Arduino boards for prototyping and programming easier. For creating, building, and uploading code to Arduino microcontrollers, it offers an extensive toolkit. Both novice and expert users may easily use the Arduino IDE thanks to its user-friendly design and intuitive functionality. It is adaptable for a variety of projects because it supports a large selection of Arduino boards and shields.

Figure 35:Arduino IDE interface

2. Blynk Libraries

The foundation for smoothly connecting IoT devices with the Blynk platform is provided by Blynk libraries. These libraries come with a plethora of capabilities that make it easier to link hardware to Blynk's cloud infrastructure, hence promoting IoT development. Developers may quickly create connections, manage authentication, transmit sensor data, and receive commands via the Blynk mobile app by using Blynk libraries. Furthermore, these libraries enable developers to use a drag-and-drop interface builder to construct custom user interfaces for their Internet of Things applications, allowing them to customise the app to meet the demands of their particular projects. With the help of the Blynk mobile app and a thriving community, developers can create feature-rich, interactive Internet of Things applications that can be managed and observed from a distance.

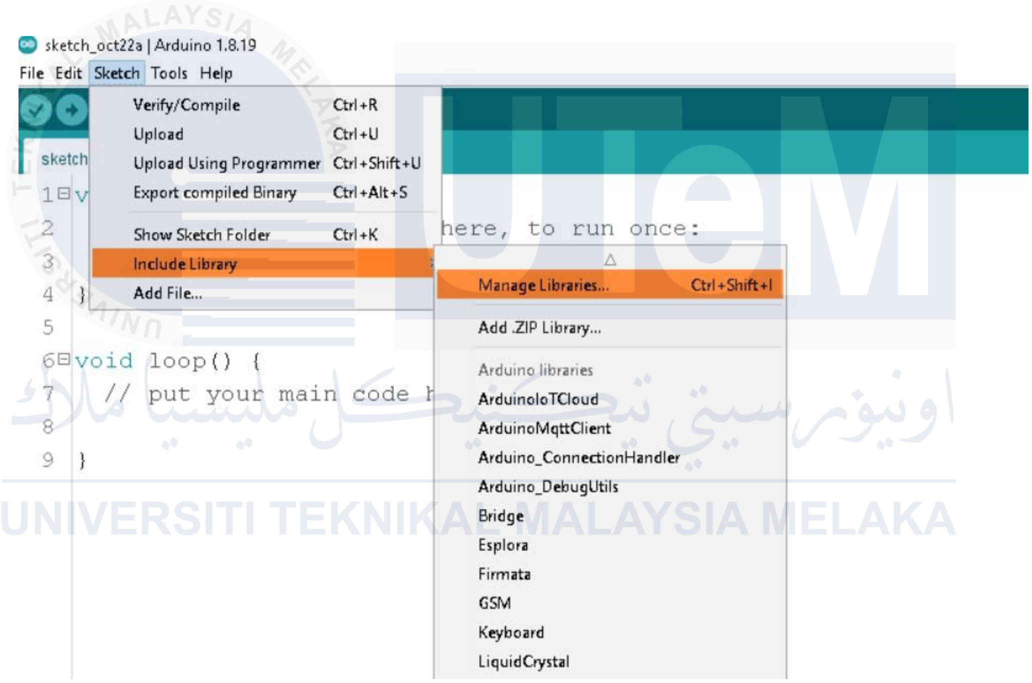

Figure 36: Blynk Libraries

3. Blynk Cloud

Users of the Blynk IoT platform can store and retrieve data from their IoT projects in the cloud with the help of the Blynk Cloud Database function. This feature makes data administration and analysis easier by allowing users to gather and safely store sensor readings, device statuses, and other pertinent data in the cloud. Through the usage of Blynk Cloud Database, customers may remotely monitor and operate their Internet of Things devices by accessing their data from any location with an internet connection. The platform also provides tools for analysing and visualising recorded data, enabling customers to obtain knowledge and decide wisely depending on the success of their IoT project.

Figure 37: Blynk cloud

4.6 High Level Design

In a project, high-level design (HLD) refers to the conceptual or architectural design of the system or solution under development. It describes the project's overall structure, components, interfaces, and main functionalities. It plays a vital role in assuring the well-structured and successful implementation of the project.

4.6.1 System Architecture

The Blynk platform for forest fire detection is used in the development of the system architecture. Blynk makes it easier to design and manage effective and user-friendly systems by making these tasks easier for users to accomplish. System architecture, which includes the design, development, and implementation of a system that satisfies all requirements to achieve the desired goals, is an essential component of product development. The system architecture uses Blynk's capabilities to link Internet of Things devices, gather sensor data, and offer remote monitoring and control features in the context of forest fire detection. The Blynk-powered Forest Fire Detection Internet of Things system architecture is depicted in the image below.

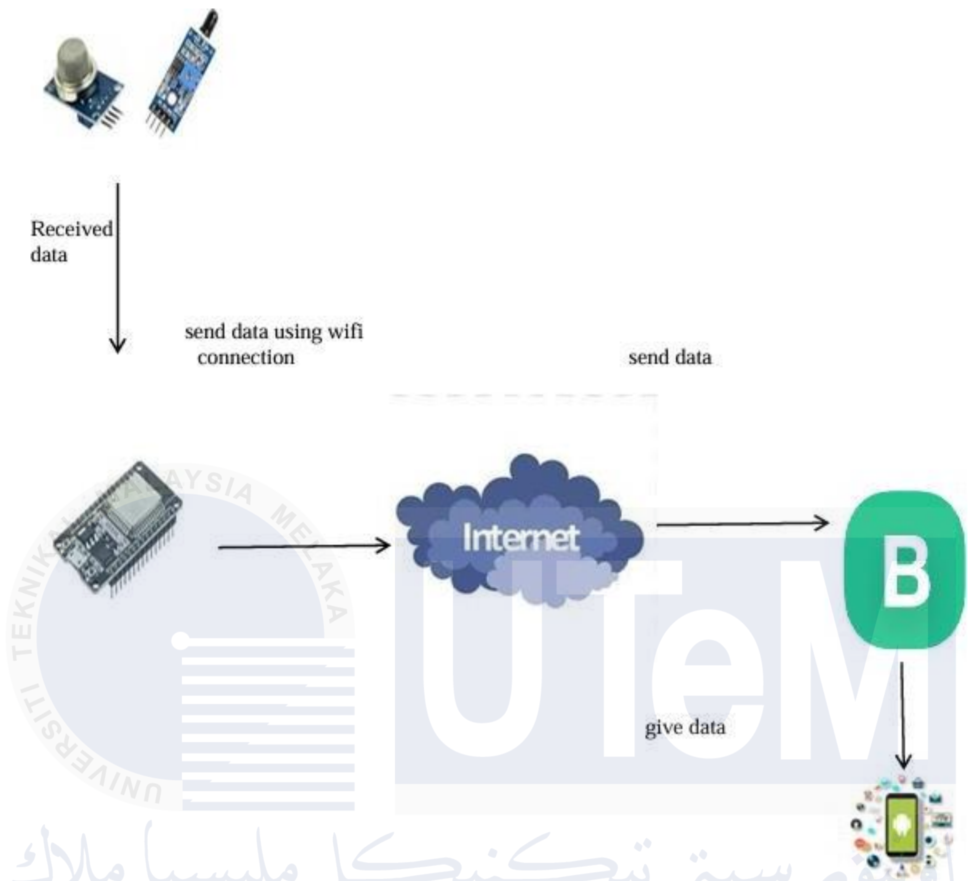

Figure 38:System Architecture

4.6.2 Interface Design

The suggested IoT system for forest fire detection would have an interface. In Human-Computer Interaction (HCI), user interface (UI) and user experience (UX) strategies are essential because they directly affect how users interact with and view computer systems and applications. The total HCI experience is influenced by them in terms of usability, efficiency, learnability, satisfaction, error management, accessibility, feedback systems, personalisation, and user engagement.

Figure 39: Home page interface

Figure 40: Fire Alert interface

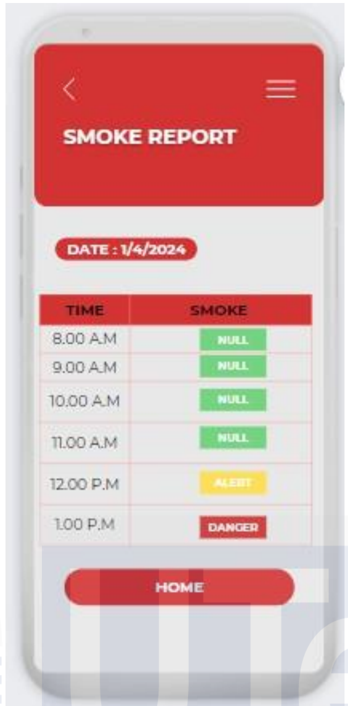

Figure 41: Smoke Report interface

Figure 42: Smoke Level Info interface

Figure 43: Graph

Figure 44: Report of Fire Incidents interface

4.7 Flowchart

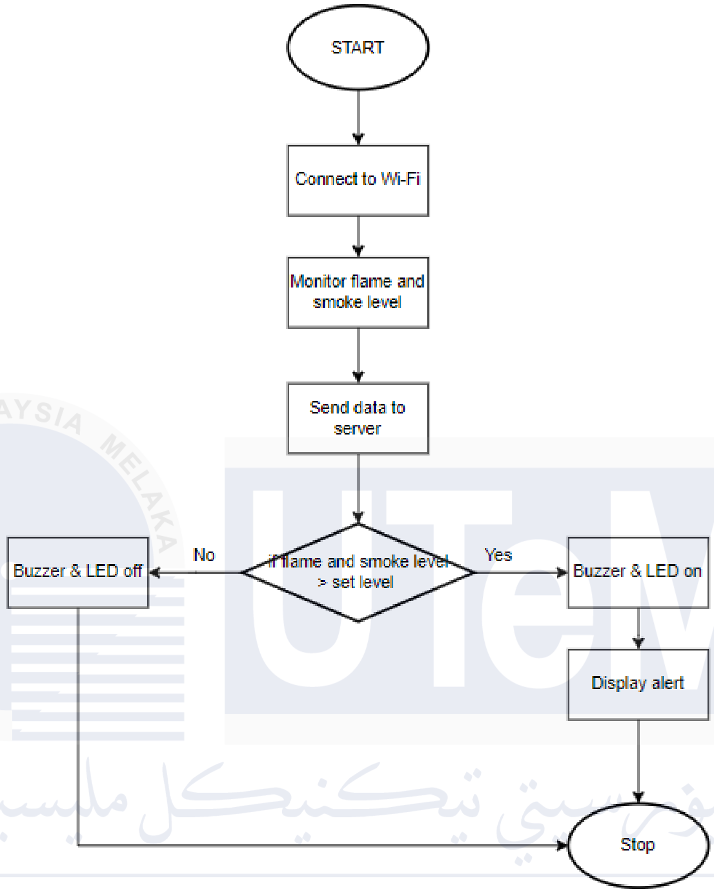

Figure 45: Flowchart

4.8 Conclusion

The steps of planning and designing a project are essential to its completion. Before starting a project, every piece of hardware and software needs to be carefully identified and inspected. This chapter acts as a preamble to the implementation process, providing an overview of the entire system flow to improve understanding prior to implementation. The next chapter, Implementation, explores the project's execution plan and anticipated results. Thus, a seamless implementation process is ensured in large part by the analysis and design phases.

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

The main focus of this chapter is on integrating Forest Fire Detection with Android smartphone applications using both hardware and software. To guarantee the project runs efficiently, it offers a thorough explanation of configuration management. The project builds up a Forest Fire Detection system using the NodeMCU ESP32 microcontroller. Notifications are received on the application, and alerts are sent when fire and smoke levels exceed critical levels. This configuration demonstrates the prototype's successful operation. Version control protocols, software configuration management, software development environment setup, and implementation status are among the activities involved.

5.2 Development Environment Setup

There are hardware and software requirements for setting up the development environment for the Smart Forest Fire Detection IoT System. Every stage is explained step-by-step and with clear instructions. Chapter 4 goes into detail on the hardware and software requirements, and the parts that follow provide more details on the connections.

5.2.1 Hardware Development Setup

Figure 47 below shows how the hardware that have setup. There are flame sensor, smoke sensor, LED, LCD and buzzer that is attached to the NodeMCU ESP32 microcontroller. NodeMCU ESP32 microcontroller is received command from arduino IDE to control the hardware.

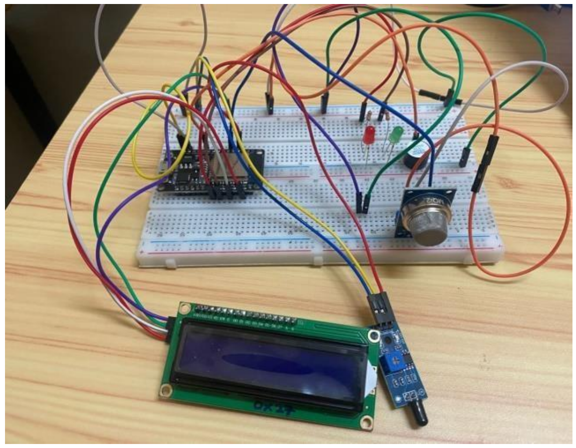

Figure 46: Hardware Setup

5.2.1.1 Hardware Installation

Connect the flame sensor to the NodeMCU ESP32 microcontroller using a male-to-male jumper wire, making that the male end is attached to pin 34. Next, pin 32 and a male-to-male jumper are used by the smoke sensor. Use pin 14 and a male-to-male jumper for the buzzer. Other than that, I use pins 18 and 5 for the green and red LEDs, respectively. You can refer to figure 48 below for this step. The detail pins that the hardware inserts into the NodeMCU ESP32 microcontroller are displayed in Table 7. Figure 48 shows the GPIO pins on the NodeMCU ESP32, and Figure 49 shows what the finished product looks like.

Table 7: Details of each pins number

Hardware	Pins
Flame Sensor	34

Smoke Sensor	32
LCD	22
LED Light (Green)	18
LED Light (Red)	5
Buzzer	14

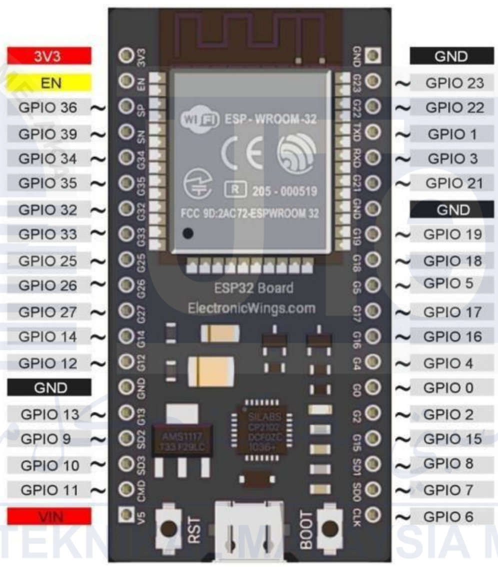

Figure 47:GPIO pins for NodeMCE ESP32

Figure 48: The final product

5.3 Software Development Environment Setup

The specifications for creating the prototype for this Smart Forest Fire Detection project are outlined in this subtopic. The construction of the hardware and software components requires adherence to these criteria. These components also give users the opportunity to actively contribute to the project's creation. The following elements make up the project's environment:

i. ESP32 NodeMCU environment setup

One of the main parts of the development environment built for the Smart Forest Fire Detection project is the ESP32. The Arduino IDE may be used to program the microcontroller and configure it, creating the ESP32 Smart Forest Fire Detection system. How to link this microcontroller to additional components is covered in the following categories (in the upcoming sections). The major component of this project is an ESP32 board, which is required for the detecting system and must be wifi capable.Thus, in order to produce a project that is successful, we also need to integrate the ESP32 with both hardware and software components. The necessary hardware and software for setting up an ESP32 environment are listed below:

a. Hardware Requirements

- NodeMCU 32 microcontroller
- Flame Sensor
- Smoke Sensor
- LCD
- LED
- Buzzer

- Power Supply cable
- Jumper wire
- Resistor

b. Software Requirement

- Arduino IDE
- Blynk

These components are essential to the development environment for configuring the Smart Forest Fire Detection system, according to the hardware and software requirements mentioned above. With its integrated Wi-Fi, the ESP32 microcontroller makes it easy to connect to a user's smartphone and sends alerts to users about possible fire risks. Together, the smoke and flame sensors identify environmental dangers; the buzzer and LEDs give visible and audible alarms; the LCD shows data in real time. An overview of the system setup for the ESP32 environment built in this project is provided below.

Figure 49: ESP 32 NodeMCU environment setup

No	System Configuration	Specification
1	Operating System	No real Operating System
2	Hardware	• Operating Voltage: 3.3V • Input Voltage: 7-12V • Flash Memory: 4 MB • 2.4 GHz Wi-Fi, supporting WPA/WPA2 • Supports standard

		TCP/UDP Server and Client
3	Software	Arduino IDE

ii. Blynk environment setup

The Smart Forest Fire Detection system's user-friendly interface is made possible by the Blynk platform, which also enables real-time warnings and monitoring for users. This project makes use of Blynk to send alerts when smoke or fire is detected and to visualise sensor data. Every facet of the Blynk cloud environment is deliberated over and set up to guarantee seamless functioning.Users are guaranteed to receive real-time alerts via the Blynk app in the event that smoke or fire levels surpass safety criteria, thanks to the ESP32's interface. By keeping the data synchronised, this configuration enables customers to receive the most recent information straight to their cellphones.

iii. Smart Forest Fire Detection setup

For users to successfully monitor the Smart Forest Fire Detection system, the application needs to run smoothly. The user's smartphone receives notifications as soon as the system identifies a possible threat. Because of the application's timely alerting feature, users can react promptly to any fire dangers found in the forest environment.

5.4 Software Configuration Management

An overview of the configuration management system's design and setup for the Smart Forest Fire Detection project is given in this subtopic. The configuration deals with the technical instruments that oversee and facilitate the project's development. The code for the ESP32 microcontroller is written and assembled using the Arduino IDE. The smartphone app that connects to the system and sends out real-time alerts in the event of a fire or smoke detection was created using Blynk.

5.4.1 Configuration Environment Setup

The setup for the Smart Forest Fire Detection system is carried out step by step. The flowcharts below illustrate the process, including the setup of the ESP32 prototype and the Blynk integration. This configuration ensures that the system operates effectively, providing real-time monitoring and alerts as intended.

i. Development and configuration (ESP32 and Arduino IDE)

Table 8: Show the development and configuration of ESP32

Component Name	Software Implemented
ESP32	Arduino IDE
Flame Sensor	Data Collection and Monitoring
Smoke Sensor	Data Collection and Monitoring
LCD	Data Display
LED Light (Green)	Status Indicator
LED Light (Red)	Alert Indicator
Buzzer	Alert Notification
Blynk	User Interface and Notifications

The software and parts used in this project for the ESP32 are displayed in Table 8. Developing and uploading the code to the ESP32 requires the Arduino IDE. The prototype would not work properly if Port 10 COM was not available for uploading the code. Hardware elements included in the project include buzzers, LCDs, LEDs, smoke and flame sensors, and LEDs. Furthermore, Blynk is utilised to deliver notifications and an intuitive user interface to the user's smartphone, facilitating efficient system monitoring and control.

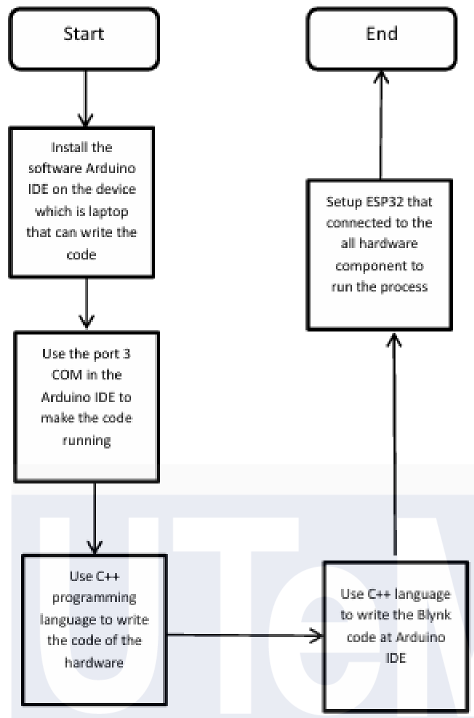

Figure 50: flowchart of the development and configuration ESP32

Below are show the step configuration for Arduino IDE environment which includes the libraries configuration ESP32 NodeMCU is shown below. This step by step helps the user to do the work with easily and faster.

Step 1 : Download the Arduino IDE from the Arduino website and run the installer. Figure 52 shows the website of arduino IDE.

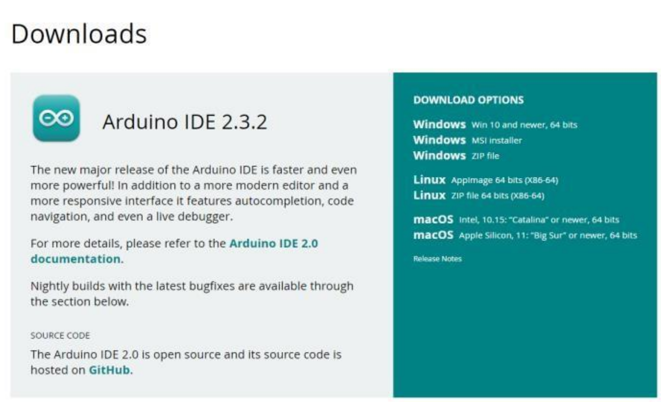

Figure 51:Website for Arduino IDE

Step 2 : Open the software after installation.

Step 3 : The interface of Arduino IDE. Figure 53 show the interface of Arduino IDE and need to put the source code to connect blynk and ESP32.

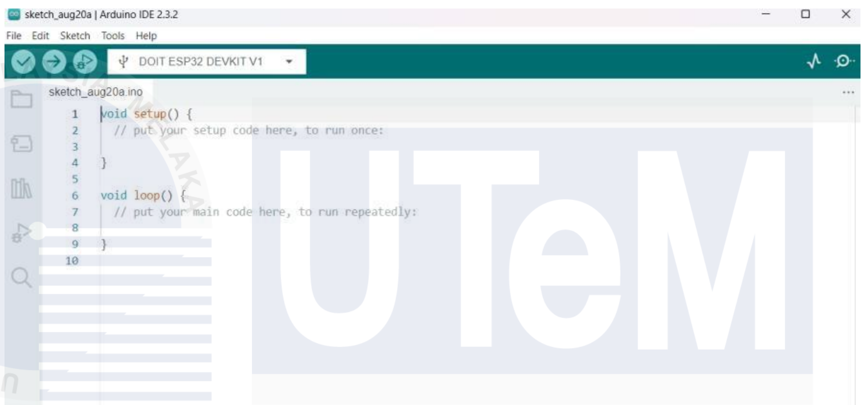

Figure 52:Arduino IDE interface

Step 4 : Install ESP32 board libraries to connect with Arduino IDE. For this step we need to go to the file and go to preferences.

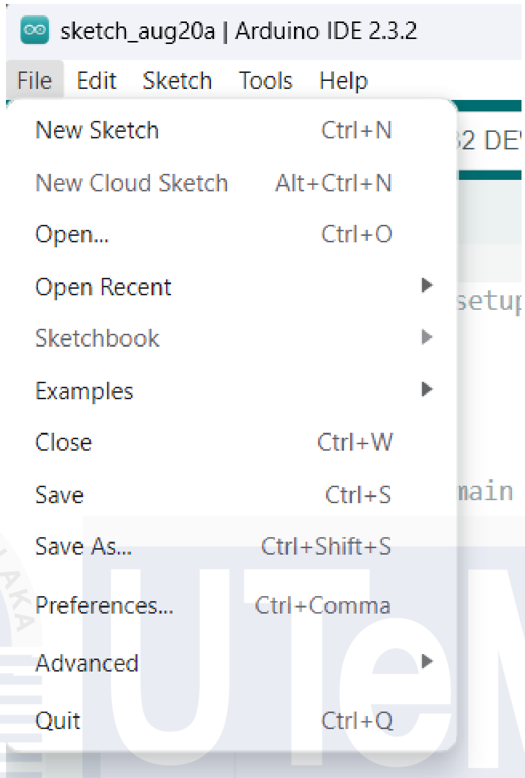

Figure 53:Press file and go to the preferences

Step 5 : Enter ―Additional board manager URLs‖ field‖ and copy the URL. Figure 55 shhows the preferences interface

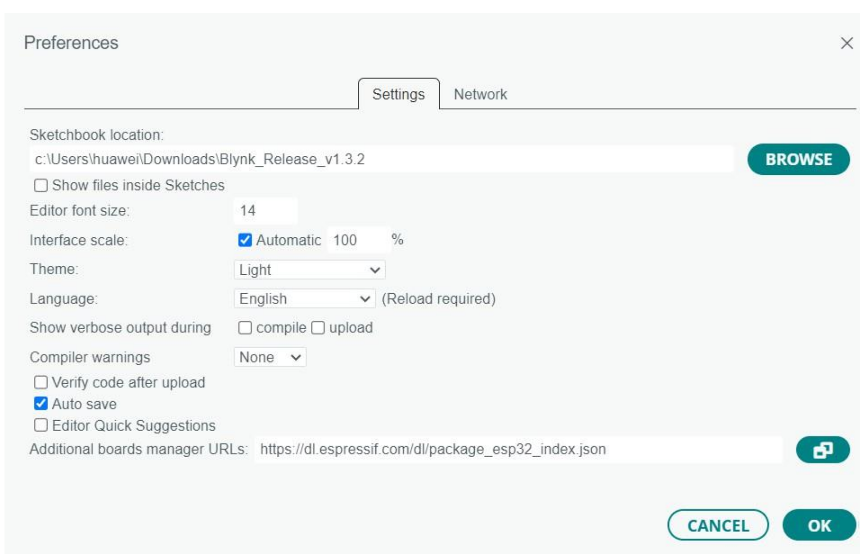

Figure 54:Paste the URL to board manager

Step 6 : Open the board manager and search for ESP 32 and click install. Figure 56 shows the library of Arduino IDE. Figure 56 shows the ESP32 library in boards manager

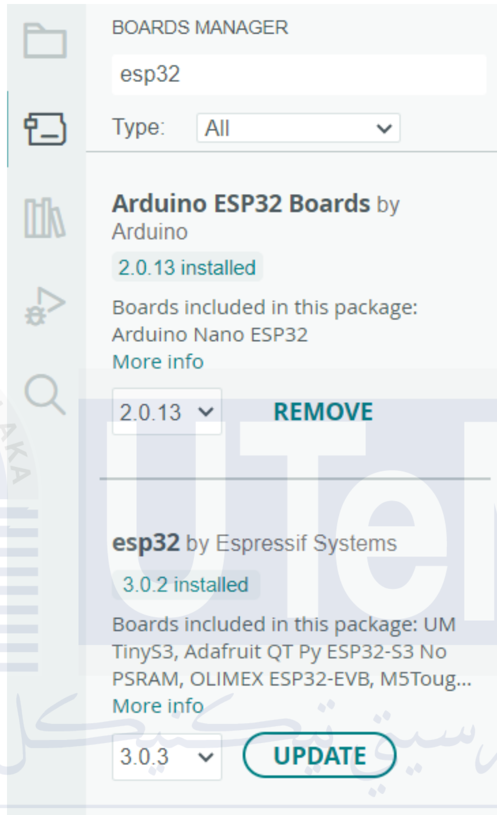

Figure 55: Select and install the ESP32 library

Step 7: Select the Port COM3. Figure 57 shows the port COM3 that we choose and the board

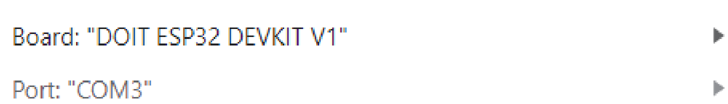

Figure 56: Select the Port COM3 and the board

ii. Development and Configuration (Flame and Smoke sensor)

Table 9: development and configuration of the flame and smoke sensor.

Component Name	Software Implemented
Flame Sensor and Smoke Sensor	Arduino IDE
	Port 3 COM
	C++ language

The development and configuration of the flame and smoke sensors are shown in Table 9. To ensure that the components were functioning properly, the flame and smoke sensors were configured using the Arduino IDE, connected via Port 3 COM, and programmed in C++. The creation and configuration code are then used to build the prototype.

Figure 57: Flowchart

Figure 58 is a flowchart that depicts the steps to ensure that the modules are operational. To operate this prototype, the flame and smoke sensors are connected to the ESP32 microcontroller. These sensors are essential for detecting fire hazards and triggering safety responses.

Figure 58: Flame sensor

Figure 59 shows the flame sensor connected to the ESP32 microcontroller. The flame sensor is responsible for detecting infrared light, which indicates the presence of fire. When a flame is detected, the sensor sends a signal to the ESP32, triggering an appropriate response such as activating an alarm or LED. This sensor is crucial for early fire detection, helping to mitigate potential hazards.

Figure 59: Smoke sensor

Figure 60 depicts the smoke sensor connected to the ESP32 microcontroller. The smoke sensor is designed to detect the presence of smoke particles in the air, which could indicate the onset of a fire. When smoke is detected, the sensor signals the ESP32 to take necessary action, such as triggering a buzzer.

iii. Development and configuration (Blynk)

Data is stored and synced in real time by this Smart Forest Fire Detection project using Blynk. Users can keep informed about potential fire threats and receive the most latest information by using the platform that Blynk offers, which enables continual updates. The diagram below displays the code that has been set up with Blynk to handle these data updates. To keep the system timely and correct, all data is updated with the date and time.

Step 1 : Sign Up/Log in to the Blynk. Figure 61 shows the interface of Blynk to Log in to the application

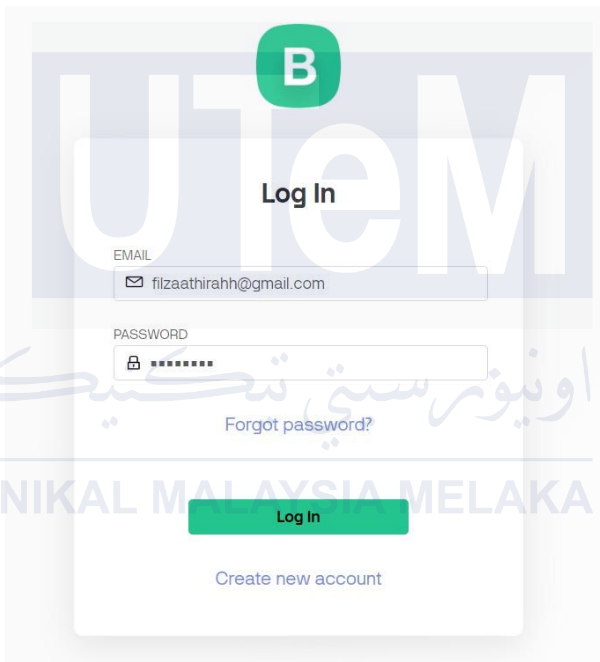

Figure 60: Blynk interface

Step 2 : Create new template and choose ESP32 for the hardware, WiFi for connection type . Figure 62 shows the interface of Blynk to create a new template for the project.

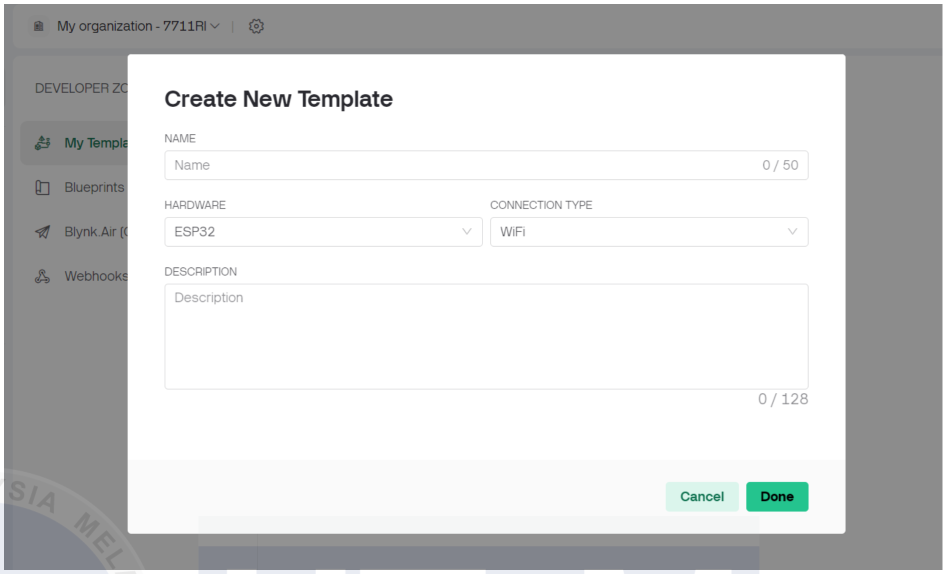

Figure 61: Blynk template

Step 3 : Add new device and choose —From template —. Figure 63 shows the interface to add device from template that we create

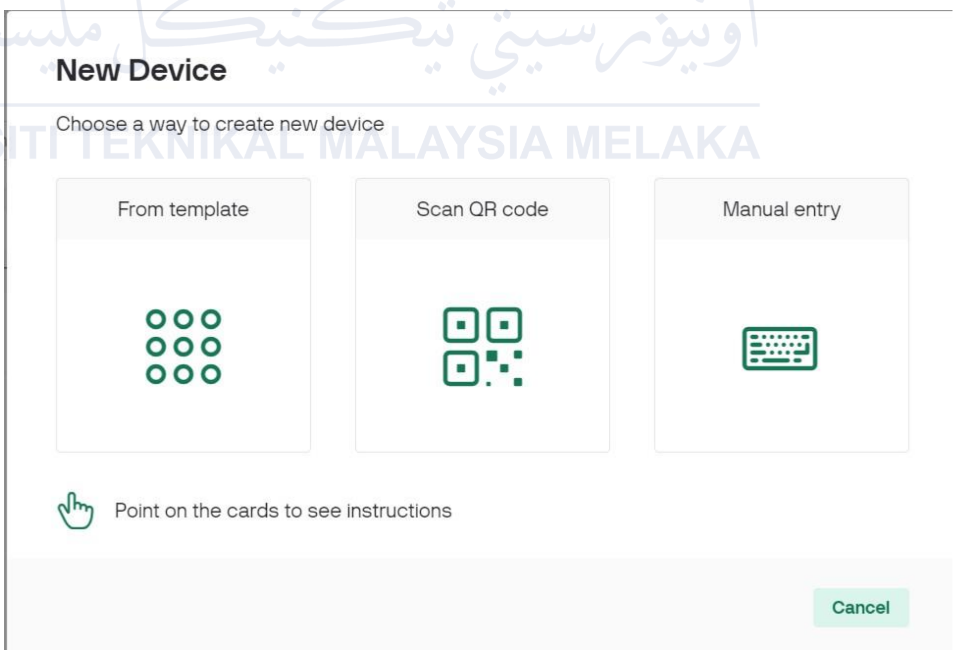

Figure 62: Add device from template

Step 4 : Choose the template that we create in the step 2 and type in a device name at device name dialog box. Figure 64 shows the interface to create new device.

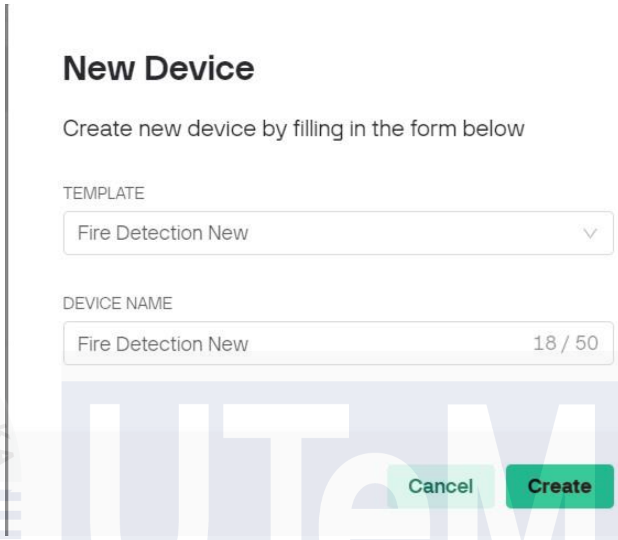

Figure 63: Create new device

Step 5 : We can see Template ID, Device name and Auth Token, and copy to clipboard to copy and save the header for information. Figure 65 shows the token and id that we need to connect with Blynk.

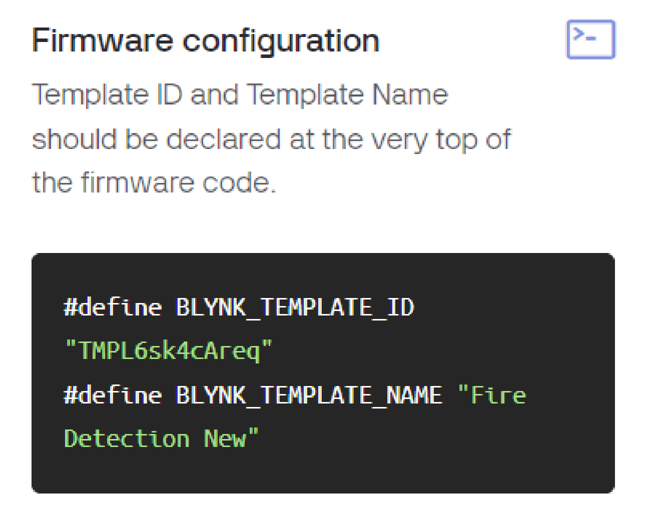

Figure 64: Token and ID from blynk

Step 6 : Open Arduino IDE and paste the Template ID, Device name and Auth Token. Then, put the ssid [] and pass [] variable and fill in the WiFi to send data to Blynk server. Figure 66 shows the Wi-Fi credentials that need to set up to connect with Blynk

```
#define BLYNK_TEMPLATE_ID "TMPL6sk4cAreq"
#define BLYNK_TEMPLATE_NAME "Fire Detection New"
#define BLYNK_AUTH_TOKEN "X-eZil88gr4WWL-y99-HYLjeRNOZzpJQ"

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <BlynkSimpleEsp32.h>

// WiFi credentials
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Fyp";
char pass[] = "Fyp12345";
```

Figure 65: Coding blynk at arduino IDE

Step 7 : Setup Blynk Web Dashboard then click datastream tab to setup data stream, click for the new Datastream followed by Virtual Pin . Figure 67 shows the datastreams in Blynk.

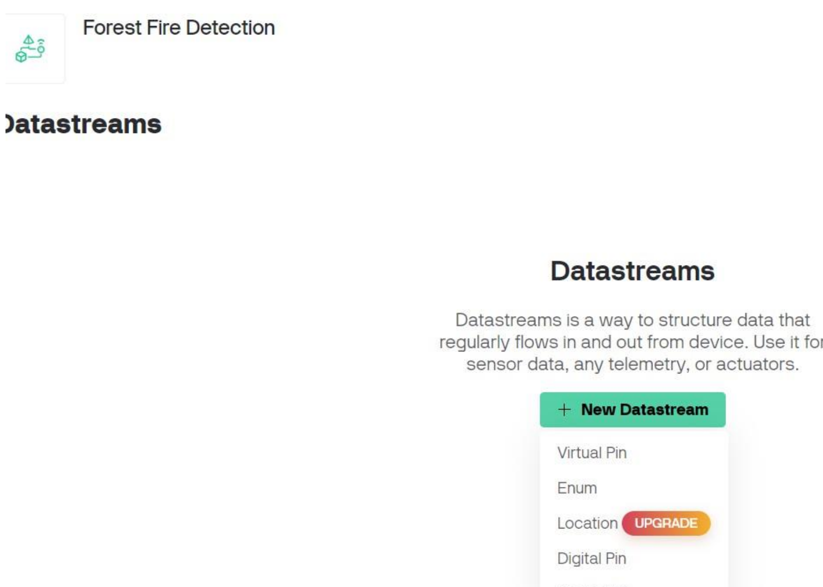

Figure 66: Create datastream

Step 8 : Setup the virtual pin in the datastream tab of blynk . Figure 60 shows the virtual pin to set up the Blynk.

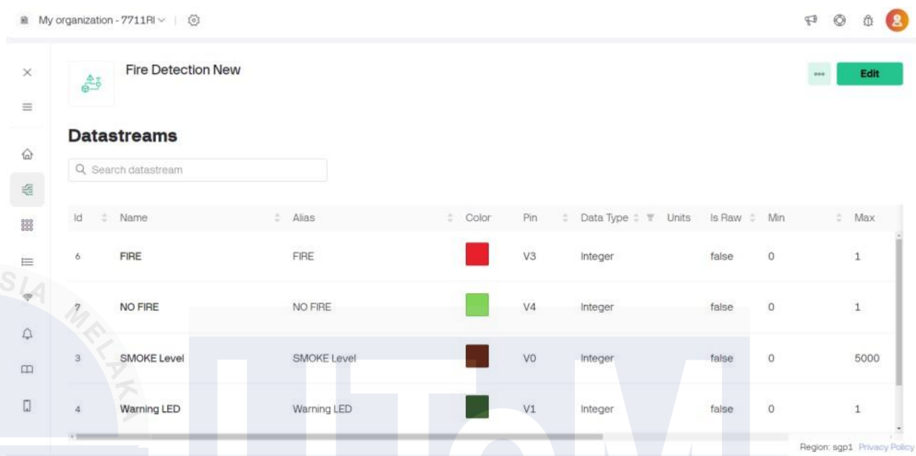

Figure 67: Setup virtual pin

Step 9 : Setup the Web Dashboard . Figure 69 shows the web dashboard of Blynk

Figure 68: Create web dashboard

Setup 10: Configure the coding for blynk code at Arduino IDE. Figure 70 shows the source code that need to connect blynk and ESP32

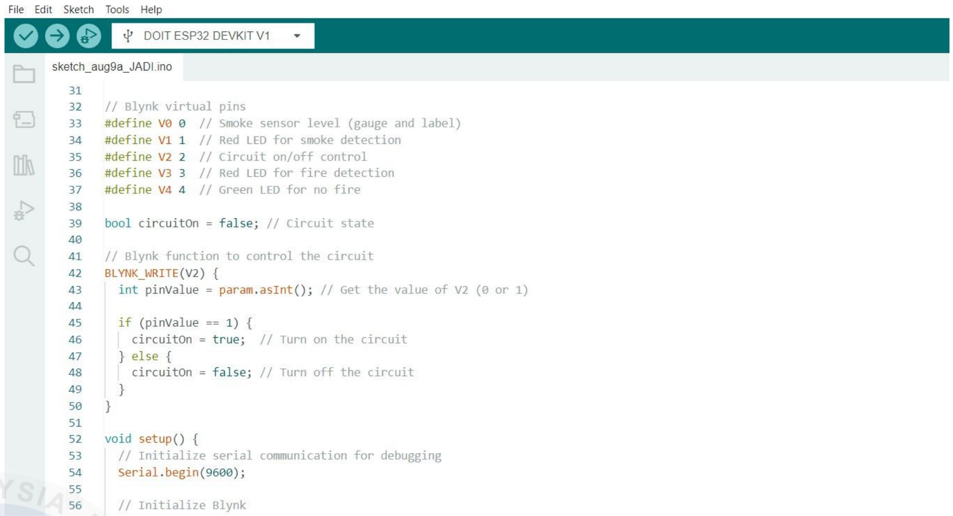

Figure 69: Coding blynk at arduino IDE

5.4.2 Version Control Procedure

The ESP32 board setup in the Arduino IDE and the source code control methods are covered in detail in this section. The port that is configured on the ESP32 board is COM3. Make sure the port is set to COM3 and that the ESP32 board is selected in the Arduino IDE. Using the Arduino IDE's Library Manager, install the required Blynk library. Connect the ESP32 to the Wi-Fi module using the supplied source code, then incorporate Blynk for real-time data management and notifications. Verify that every part, including the smoke and flame sensors, is linked to the ESP32 in the proper manner as specified by the source code. The settings and source code for integrating the ESP32 with Blynk to manage real-time data and notifications are shown in the figures below.

i. Figure 71 shows the Wi-Fi credential that we need to connect with the Blynk sketch_aug9a_JADI.ino

```cpp
#define BLYNK_TEMPLATE_ID "TMPL6sk4cAreq"
#define BLYNK_TEMPLATE_NAME "Fire Detection New"
#define BLYNK_AUTH_TOKEN "X-eZil88gr4WWL-y99-HYLjeRNOZzpJQ"

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <BlynkSimpleEsp32.h>

// WiFi credentials
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Fyp";
char pass[] = "Fyp12345";
```

Figure 70: Show the code that use for connect component WiFi module and blynk in the Arduino IDE.

ii. For figure 72 shows that the source code for flame sensor to connect with ESP32

```cpp
// Print sensor values for debugging
Serial.print("Flame sensor value: ");
Serial.println(flame_detected);
Serial.print("Smoke sensor level: ");
Serial.println(smokeLevel);

// Map smokeLevel from 0-4095 to 0-5000 for the gauge
int mappedSmokeLevel = map(smokeLevel, 0, 4095, 0, 5000);

// Update Blynk with sensor values
Blynk.virtualWrite(V0, mappedSmokeLevel);      // Send smoke level to Blynk gauge and label
Blynk.virtualWrite(V0, smokeLevel);            // Send smoke level to Blynk label

// Logic to handle sensor readings
if (flame_detected == LOW) { // Flame detected
  // Alert state for flame detected
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("FIRE DETECTED");
  lcd.setCursor(0, 1);
  lcd.print("ALERT!");
  digitalWrite(buzzer, HIGH);    // Turn on buzzer
  digitalWrite(red_led, HIGH);   // Turn on red LED
  digitalWrite(green_led, LOW);  // Turn off green LED
  Blynk.virtualWrite(V1, LOW);   // Turn off green LED in Blynk
  Blynk.virtualWrite(V3, HIGH);  // Turn on red LED in Blynk
  Blynk.virtualWrite(V4, LOW);   // Turn off green LED in Blynk for no fire

  // Send notification to Blynk app
  fire_notification();
```

Figure 71: Coding for flame sensor

iii. Figure 73 shows the source code for smoke sensor to connect with ESP32

```
}
else if (mappedSmokeLevel > 3000) { // Smoke detected (you can adjust the threshold as needed)
  // Alert state for smoke detected
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("SMOKE DETECTED");
  lcd.setCursor(0, 1);
  lcd.print("ALERT!");
  digitalWrite(buzzer, HIGH);    // Turn on buzzer
  digitalWrite(red_led, HIGH);   // Turn on red LED
  digitalWrite(green_led, LOW);  // Turn off green LED
  Blynk.virtualWrite(V1, HIGH);  // Turn on red LED in Blynk
  Blynk.virtualWrite(V3, LOW);   // Turn off red LED in Blynk
  Blynk.virtualWrite(V4, LOW);   // Turn off green LED in Blynk for no fire
}
else { // No flame and no smoke
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("NO FLAME");
  lcd.setCursor(0, 1);
  lcd.print("NORMAL!");
  digitalWrite(buzzer, LOW);     // Turn off buzzer
  digitalWrite(red_led, LOW);    // Turn off red LED
  digitalWrite(green_led, HIGH); // Turn on green LED
  Blynk.virtualWrite(V1, LOW);   // Turn off red LED in Blynk
  Blynk.virtualWrite(V3, LOW);   // Turn off red LED in Blynk
  Blynk.virtualWrite(V4, HIGH);  // Turn on green LED in Blynk for no fire
}
```

Figure 72:Coding for smoke sensor

iv. Figure 74 shows the source code to send the notification from Blynk app to the user mobile phone

```
void fire_notification() {
  Blynk.logEvent("fire_notification", "Fire detected! Take immediate action!");
}
```

Figure 73: Coding for blynk to send notification to the user mobile phone

5.5 Implementation Status

This Smart Forest Fire Detection implementation status is a part of the time and status. In the table 10 is explain about the status of the development of each component or module. In this explanation, it comprises of module name, description, duration to complete and date completed.

Table 10: Show the implementation status project

No	Module Name	Description	Duration
1	Install and set up for the library installer	Installation and setting up the software Arduino IDE with the port and Blynk	3 Days
2	Assemble sensor	Setting up for the	7 Days

	and hardware	ESP32, Flame and smoke sensor, LCD, LED, buzzer and setup for the wire and pin	
3	Implement all source code	Implement code for all part hardware which is ESP32, sensor, blynk, android application and notification	40 Days
4	Develop application	This application is build the software and install it on the smartphone to ensure that the updates are received by the user.	15 Days
5	Test prototype	Check to see if the prototype is functional.	5 Days

5.6 Conclusion

To ensure efficient execution, the creation process is thoroughly examined during the Smart Forest Fire Detection project's implementation phase. To ensure the project's seamless and dependable operation, it is created through a number of clearly defined steps. To improve dependability, the configuration and environment are carefully considered. The setup's hardware and software components are covered in detail. In order to make a smooth transition to the next stage—detailed testing and analysis—all the information gathered is organised in Chapter 6.

CHAPTER 6

TESTING

6.1 Introduction

The Smart Forest Fire Detection project's testing phase is covered in this chapter to make sure the created system satisfies all specifications and performs as planned. To obtain the intended results and validate the project's aims, every component—including the ESP32 microcontroller, flame sensor, and smoke sensor—is put through a rigorous testing process. Ensuring the system's dependability and efficacy in real-time forest fire detection is the aim. Aspects of testing such as the test plan, test setup, test design, test results, and their interpretation are all covered in this chapter.

6.2 Test Plan

The main techniques and procedures used in product testing are covered in detail in this section. It also describes the roles and protocols followed by people taking part in the testing phase. The purpose of the test plan is to find any possible problems that might arise when the gadget is operating. Making sure that the system as a whole and the product design efficiently meet the needs of the user is the main objective. It is imperative to execute a comprehensive test strategy in order to ensure the dependability and efficiency of the systems and products prior to their release for user deployment.

6.2.1 Test Organization

Two people are in charge of completing the system testing procedure in this segment. The person in charge of developing the system, the system developer, is the first, and the end user is the second. In the testing process, each tester plays a different role:

i. System Developer

It is the duty of the system developer to carry out comprehensive system testing, detect any defects or mistakes, and guarantee the system's stability and seamless functioning.

ii. End user

It is the end user's responsibility to report any issues with the system, offer suggestions for enhancements, and offer comments on the functioning and user experience.

6.2.2 Test Environment

In order to verify that the prototype can transmit notifications for the Smart Forest Fire Detection system, this project tests it over Wi-Fi connectivity. The main control unit for monitoring the sensors and modules in the arrangement is the ESP32 microcontroller, which is powered by a 5V supply using a USB cable. For identifying fire dangers, flame and smoke sensors are prioritised above servo motors. The Blynk platform is linked to the system in order to handle data and deliver real-time changes. Throughout the project, a meticulously designed test environment has been implemented to guarantee accurate and dependable testing.

6.2.3 Test Schedule

Creating a test schedule and estimating the project's completion time are the main goals of this testing phase. During this time, a number of mistakes and problems surfaced, necessitating more time for implementation-related debugging and verification. The testing procedure will not end until the system is completely operational. The evaluation schedule looks like this: after the hardware is configured, the user uploads the code and tests the system to make sure everything works as it should. Therefore, it is anticipated that the prototype will operate effectively.

i. Start the prototype

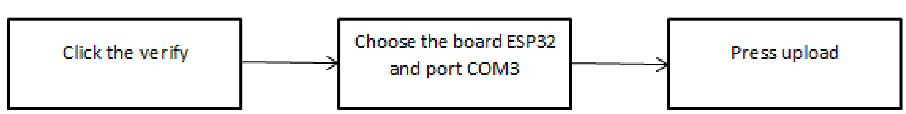

Figure 74: Show the role of the prototype

Figure 75 shows how to begin of the the prototype. Users can use the Arduino IDE's verify button to check the code have a error or not. To ensure that the prototype is work, the users should choose which board they want to use. The port is COM3 on the ESP32 board. The prototype is not be able to run if the users are unable to select this board and port. Then, to import the code into the prototype, click upload.

ii. Functionally of the prototype

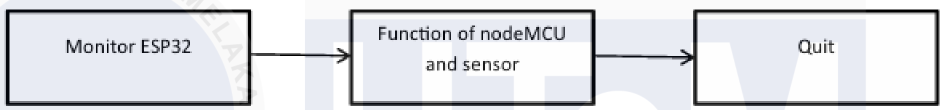

Figure 75: Show the functionally of the prototype

The prototype's functionality is illustrated in Figure 76. The key components involve the control of the ESP32 microcontroller, which connects all the sensors and modules. The fire and smoke sensors play a crucial role in detecting potential hazards. Once a fire or smoke is detected, the system triggers real-time alerts to notify the user. These functional components help ensure that users are promptly informed of any potential fire threats, allowing them to take immediate action.

iii. Connect with Blynk

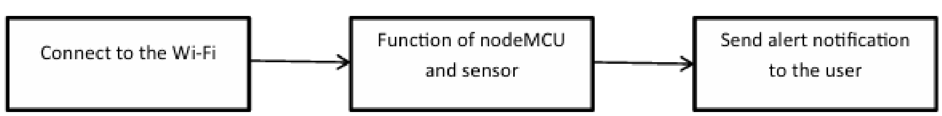

Figure 76: Show the connection with the Blynk

For this process, as shown in Figure 77, the user must be connected to Wi-Fi to connect to Blynk, which provides real-time data and sends notifications to the user's smartphone. The data, including the detection of fire and smoke events, is stored and managed through the Blynk.

iv. Start Smart Forest Fire Detection Application

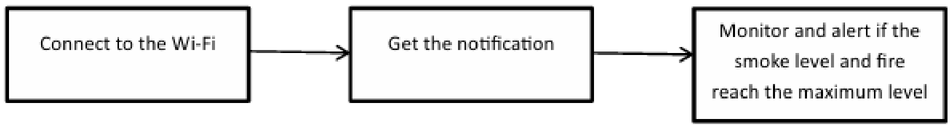

Figure 77: Show how to use the Smart Forest Fire Detection Application.

To use the Smart Forest Fire Detection application, as shown in Figure 78, the user must first connect to the Wi-Fi to receive real-time notifications from the system. Once the fire and smoke sensors detect a potential hazard, a notification will appear on the user's android, enabling prompt action to address the issue.

v. Test Strategy

This section outlines the testing strategy for the project, which aims to facilitate real-time notifications and monitoring of possible fire threats for users. Users need to install the Blynk application on their Android device in order to obtain updates and comprehensive information. The prototype used to test this capability is seen in Figure 79 below. Through the test, customers may confirm that notifications are appropriately received through the Blynk app and that the fire and smoke sensors function as intended. Prototypes have undergone extensive testing to guarantee excellent efficiency. The prototype's functioning and the Blynk app's notification system are the two areas that need to be tested.

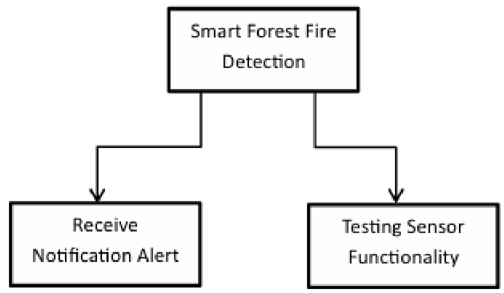

Figure 78: Two step to use in this category

i. Testing Notification Reception

Notifications are sent when the fire or smoke sensors detect a hazard, and they are delivered to the user's smartphone via the Blynk app. To receive these notifications, the android must be connected to the Wi-Fi.

ii. **Testing Sensor Functionality**

For testing the sensors' functionality, both an Wi-Fi connection and power supply are required. When the sensors detect fire or smoke, the system sends alerts in real-time, allowing users to respond promptly.

6.3.1 Classes Test

i. Functionality Test

The purpose of this functionality test is to confirm that the ESP32 microcontroller, flame and smoke sensors, and Smart Forest Fire Detection prototype all function as intended. The first task of the session was to test the ESP32 and make sure the sensors can reliably identify smoke or fire. Verifying that the sensors can accurately detect fire or smoke and that the ESP32 can initiate the proper action—such as turning on the LEDs, buzzer, or Blynk notifications—is the primary objective.

Moreover, real-time system monitoring is made simpler by the Blynk application's ability to display sensor data.

6.4 Test Design

This is one of the most crucial steps in the testing process, when the Smart Forest Fire Detection system's intended modules or components are examined for precision and efficacy. Every component or module is extensively tested to validate the procedure. To make sure the system satisfies the requirements and operates as intended, testing is essential. The project's functionality can be improved and its goals can be met by carrying out this analysis. Additionally, testing could help to improve system functionality in order to achieve the overall objectives. All parts and modules are extensively inspected to guarantee faultless operation.

6.4.1 Test Description

This section's goal is to identify the components of the Smart Forest Fire Detection system that require testing. Tables 11 to 6.3 provide specifics about the test cases used in this project. All of the parts and modules that were utilised to generate precise and fruitful outcomes are covered in this section. It is necessary to have thorough understanding of all the system components in order to use this prototype. Both the intended performance and the project's methods must be known to the users. The table below contains a list of every test case.

i. Table 11 shows the results of the Wi-Fi module testing

Table 11: shows the results of the Wi-Fi module testing.

Test	Test the Wi-Fi module's connectivity
Test purpose	To test the Wi-Fi module has connection with the internet
Test environment	For run this pre situation test, Wi-Fi module need to set up. Installation and

	setup procedure started in section 5.4.1.
Test setup	i. Download the Arduino IDE and install the library. ii. Create the source code at the ESP32 nodeMCU. iii. After run the code, the Wi-Fi module is function to test the internet connection.
Expected result	After running the code, the Wi Fi module is function to test the internet connection.
Error message	None
Result	Successful

ii. Table 12 shows the results of the flame and smoke sensor function

Table 12: shows the results of the flame and smoke sensor function

Test	Test the flame and smoke sensor connectivity
Test purpose	To test the flame and smoke sensors' ability to detect fire and smoke
Test environment	In order to run this test, the sensors must be set up as described.
Test setup	i. Download the Arduino IDE and install the library. ii. Connect the flame and smoke

	sensors to the ESP32 using jumper wires.. iii. Create the source code for the sensors iv. After running the code, verify the sensors' ability to detect fire and smoke, triggering the appropriate responses (LEDs, buzzer, Blynk notifications).
Expected result	After running the code, the sensors should accurately detect fire and smoke, and the system should respond accordingly. The green LED will turn on and the buzzer will off if there are not fire and smoke. If there are fire or smoke, it will triggered red LED and buzzer will turn on.
Error message	None
Result	Successful

iii. Table 13 shows the results of the Smart Forest Fire application on android testing.

Table 13: shows the results of the flame and smoke sensor function

Test	Test of the Smart Forest Fire Detection application’s connectivity
Test purpose	To test whether the user receives notifications about fire or smoke

	detection through the Blynk app.
Test environment	In order to run this test, the android needs to have an internet connection to connect with the Smart Forest Fire Detection system and receive notifications.
Test setup	i. Download and install the Blynk app. ii. Create a new project in Blynk and configure the widgets for sensors. iii. Connect the Blynk app to the ESP32 using the provided authentication token. iv. Create the source code to integrate Blynk with the flame and smoke sensors. v. After running the code, the user can view sensor data and receive alerts through the Blynk app.
Expected result	After running the code, the user should receive real-time notifications from the Smart Forest Fire Detection system.
Error message	None
Result	Successful

6.4.2 Test Data

i. NodeMCU ESP32 connectivity test.

The ESP32 microcontroller board is connected to the laptop using micro-USB cable as shown in figure 80

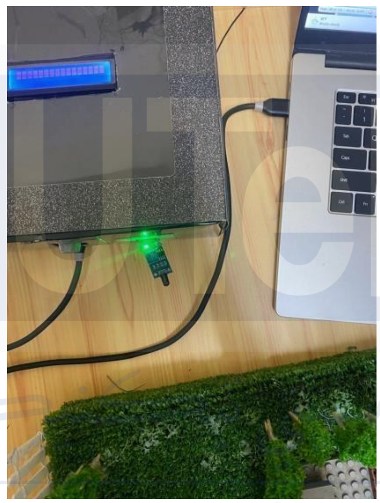

Figure 79: Connectivity testing of NodeMCU ESP32 to the laptop

The connection can be checked on the Arduino IDE to see whether the connection is successful or not. To program the Arduino IDE, go to Tools and choose Port COM3. The relation between the Arduino board and the USB port, which displays the port to upload the code, is shown in the diagram below. The console window would display that the uploading of source code was effective if there are no errors in the source code.

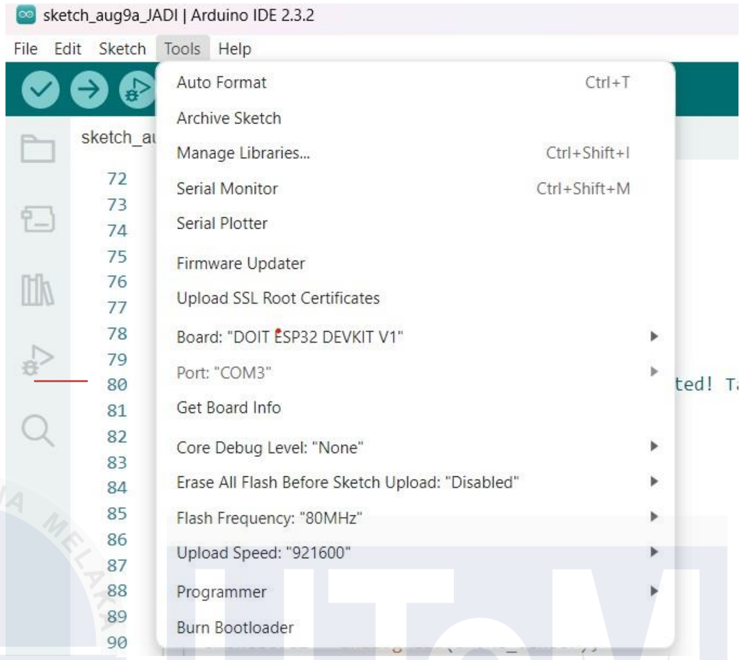

Figure 80: Arduino IDE shows the detail of port uses.

ii. Component connectivity test

Figure 82 shows the setup connection of the ESP32 and flame and smoke sensor component using the jumper wires to connect each other.

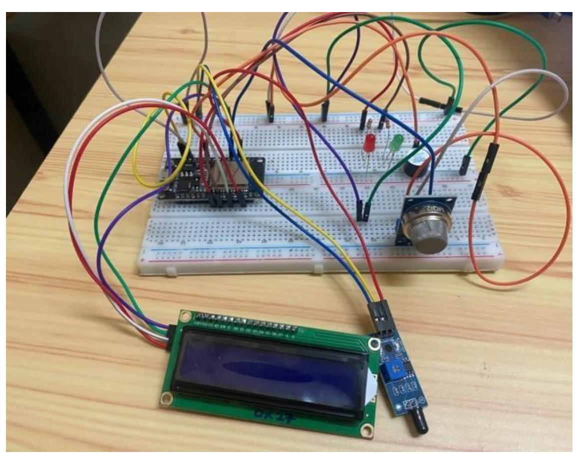

Figure 81: Component connectivity test

iii. Application connectivity test

The Blynk platform was used to develop the Smart Forest Fire Detection application for this project. On a android, this helps users receive alerts when fire or smoke is detected in the monitored area. Figure 83 shows the interface of Blynk.

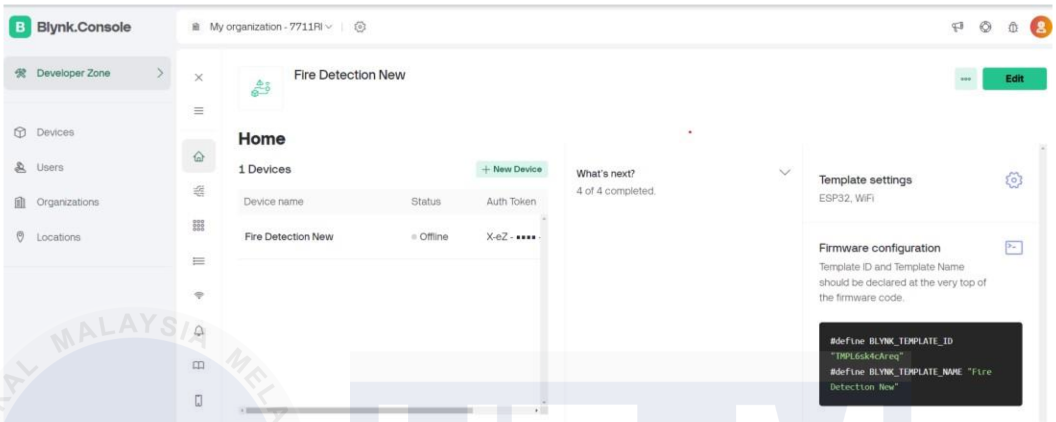

Figure 82: Blynk software.

Figure 84 shows the user information from the Blynk cloud database, where registered users receive notifications from the application.

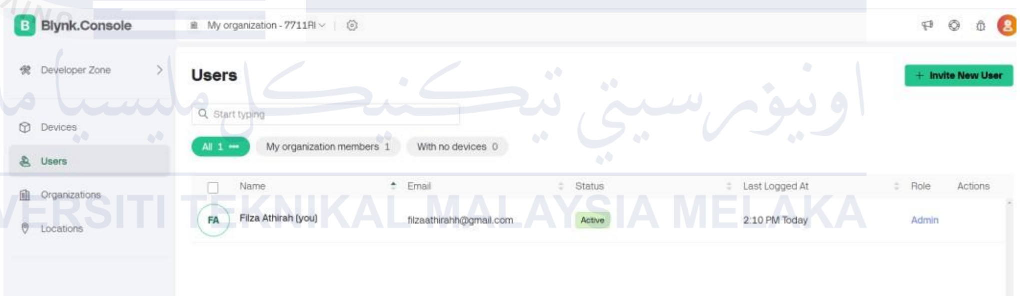

Figure 83: user information from Blynk users interface

Figure 85 displays the sensor status data received from the Smart Forest Fire Detection system, including real-time readings from the flame and smoke sensors.

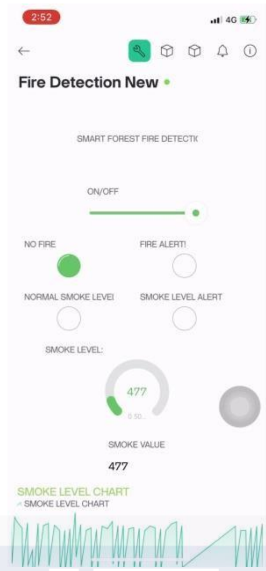

Figure 84: The sensor status data from smart forest fire detection system

6.5 Test result and analysis

Following the completion of the test schedule, strategy, and design outlined earlier, this section focuses on identifying test cases, specifying testers, and evaluating the results as either successful or unsuccessful. The detailed results of testing each component used in the Smart Forest Fire Detection project are presented below.

6.5.1 Test result on hardware

i. NodeMCU ESP32

Table 14: show the ESP32 function result and analysis

Test case identification	Test identification	Result Expectation	Success/fail
1	Uploading the codes and click run	The microcontroller is connect the	Success

		internet connection that have set	
2	When the codes are not uploaded, and the run button is not clicked.	The microcontroller have no internet connection	Success

Table 14 above shows the function of ESP32. This project uses an ESP32 Wi-Fi module to send and receive notification to the user's android. Without this module, users is unable to receive Smart Forest Fire Detection updates and has unaware to alert with the forest situation. The ESP32 must be configured in the Arduino IDE.

ii. Table 15 show the flame and smoke sensor function result and analysis

Table 15: flame and smoke sensor function result and analysis

Test case identification	Test identification	Result Expectation	Success/fail
1	Uploading the code and running the application	The sensors should correctly detect fire or smoke and trigger appropriate responses (e.g., LEDs, buzzer, notifications)	Success
2	When the code is not uploaded or the application is not	The sensors should not detect fire or smoke and no	Success

	run	notifications should be triggered	

The analysis of the flame and smoke sensor function is detailed in Table 6.5. The purpose of these tests is to ensure that the sensors accurately detect fire and smoke, and that the system responds correctly. This analysis is essential to verify the functionality and reliability of the sensors within the Smart Forest Fire Detection system.

6.5.2 Test result on application

i. Smart Forest Fire Detection application

Table 16: show the testing on the application

Test case identification	Test identification	Result Expectation	Success/fail
1	User receive the notification	Receives the notification and alert to the smart forest fire detection.	Success
2	User not receive the notification	Not receives the notification and alert to the smart forest fire detection.	Success

The testing on implementation of this project is shown in Table 16. An application provides users with the ability to quickly access and display data. The Smart Forest Fire Detection project was linked to an application that give the notification to the user to be alert. The figure 86 below show the notification that user get if there are flame or smoke in the area.

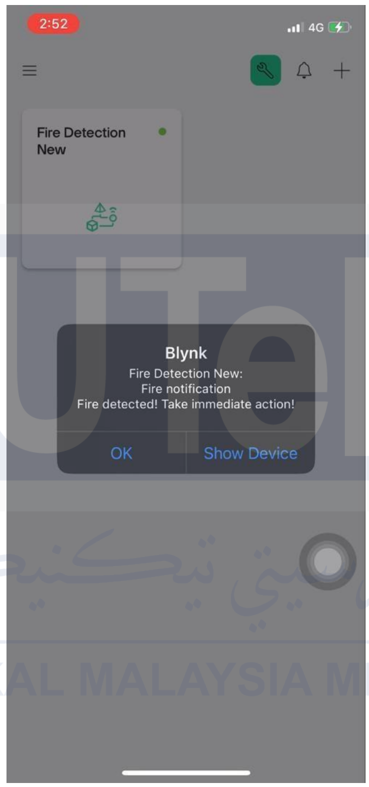

Figure 85: show the notification of Smart Forest Fire Detection application

6.5 Usability Test

The System Usability Scale (SUS), created by John Brooke in 1986, is used to measure the usability of the Smart Forest Fire Detection system. In "SUS: A Quick and Dirty Usability Scale" (ResearchGate), Brooke (1996) claims that SUS offers a quick, easy, and reliable way to assess a system's usability. This test evaluates the Smart Forest Fire Detection system's overall efficacy in detecting fire and smoke, as well as its ease of use and user satisfaction.

- **Ease of Use:** How intuitive is the system for users to set up and operate?

- **User Satisfaction:** How satisfied are users with their experience using the system?
- **System Effectiveness:** How well does the system perform its intended functions, including accurate fire and smoke detection and timely alerting?

For each of the ten items in the SUS questionnaire, respondents will choose from five options, ranging from Strongly Agree to Strongly Disagree. These enquiries are made to be readily and rapidly answered, making it possible to gather user input in an effective manner. Google Forms, which offers structured, real-time data and graphic charts for simple analysis, is used to conduct the survey. The ten questions that make up the questionnaire are listed below.

1. I believe I would use this Smart Forest Fire Detection system regularly.
2. I found the system more complicated than necessary.
3. I thought the system was straightforward to use.
4. I feel I would need technical support to effectively operate this system.
5. I found the functions within this system to be well integrated.
6. I encountered some inconsistencies while using this system.
7. I think most people would quickly become familiar with this system.
8. I found the system somewhat cumbersome to use.
9. I felt confident using the Smart Forest Fire Detection system.
10. I had to learn a lot before I could effectively use this system.

6.6.1 Results

According to Symk (2020), a minimum of five users is needed to obtain reliable data. For this reason, the questionnaire was completed by 14 respondents. Each participant underwent a usability testing session with the Smart Forest Fire Detection system and was given 1-2 minutes to fill out the questionnaire via Google Forms. The results of the testing are presented from figure 86 to figure 98 below.

Figure 86: Name of the responses

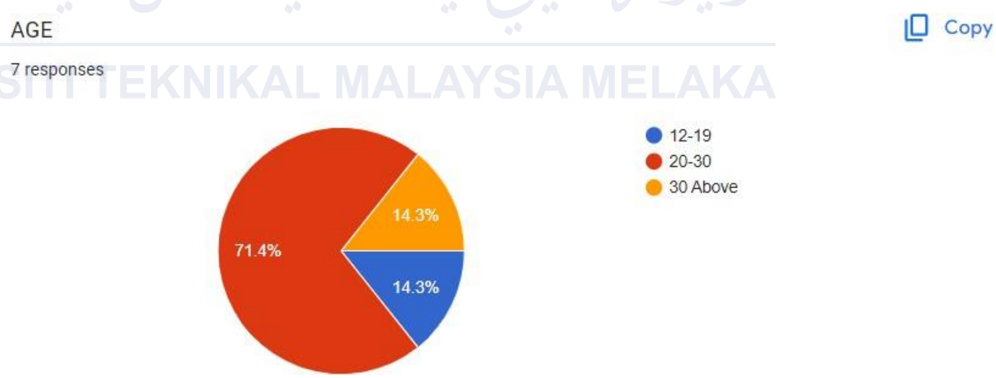

Figure 87: Age of the responders

Figure 88: Linear scale of the question

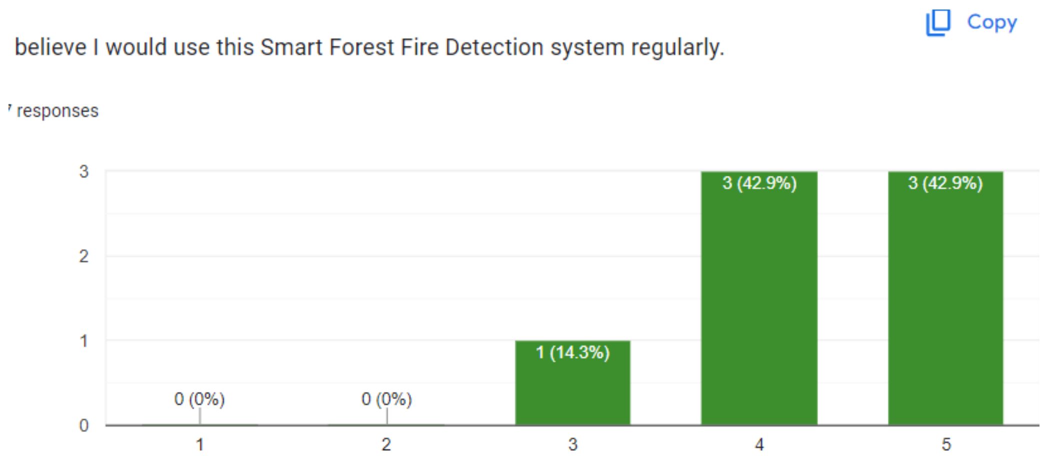

Figure 89: Answer of question 1

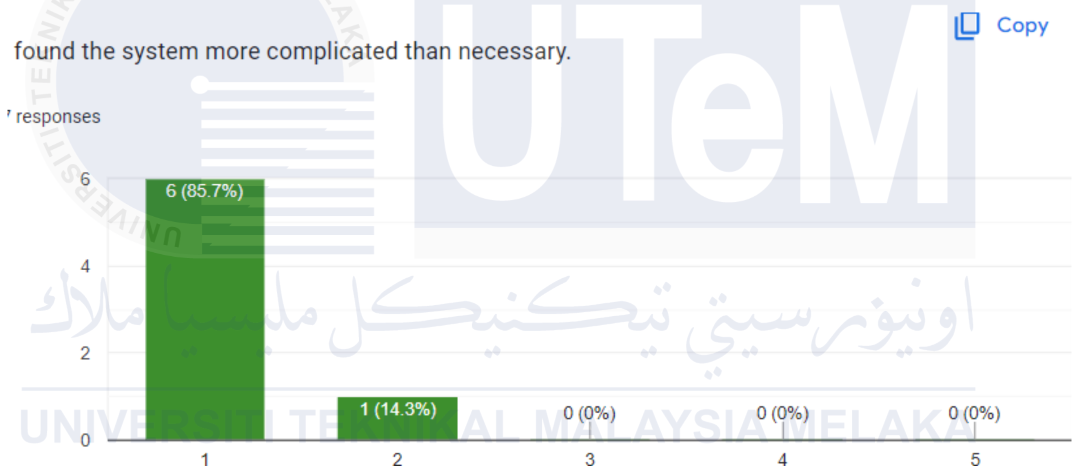

Figure 90: Answer of question 2

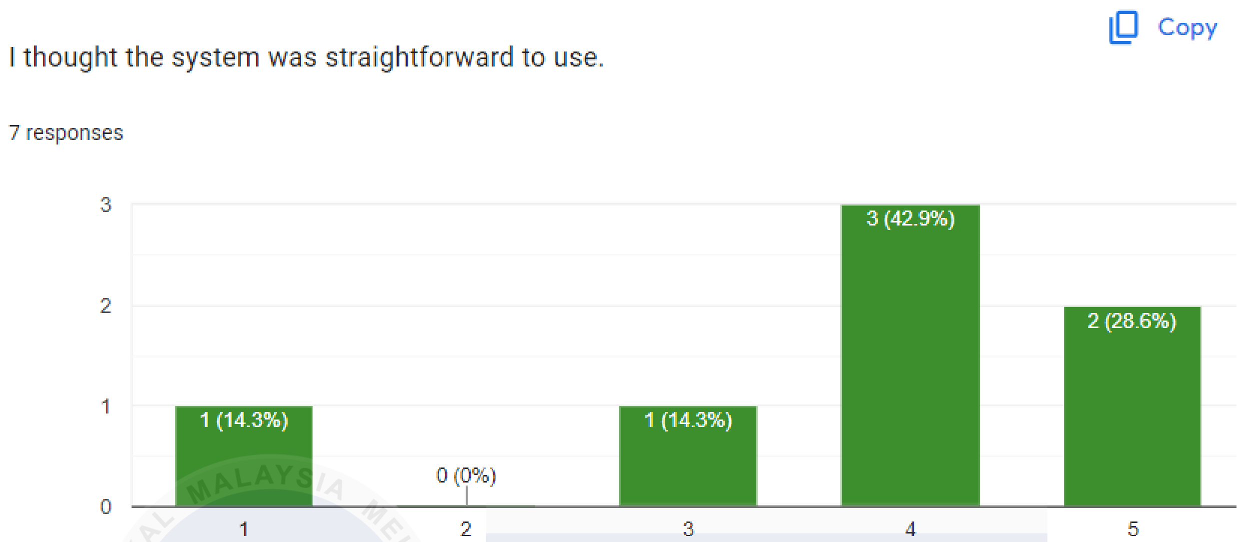

Figure 91: Answer of question 3

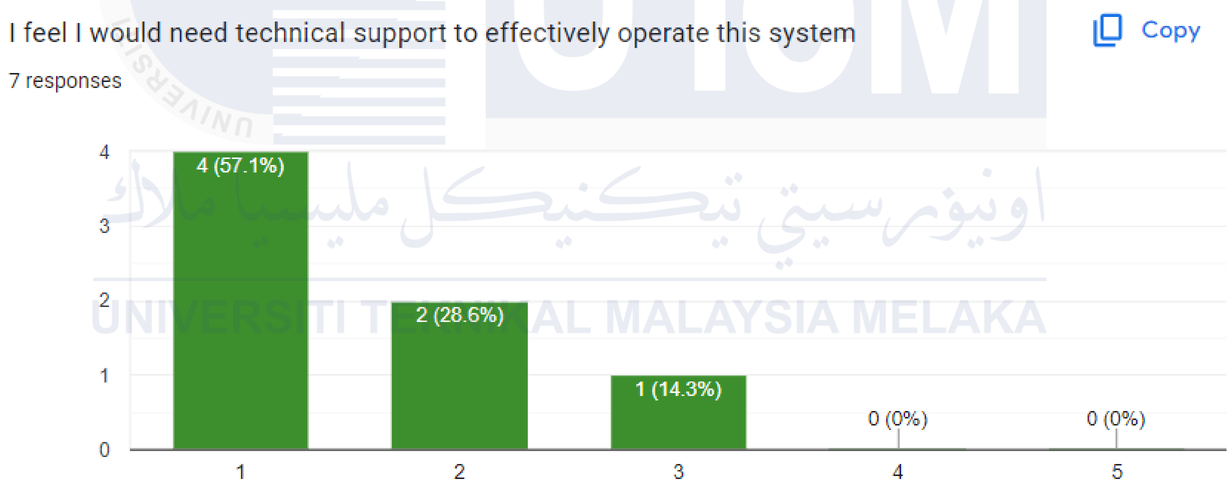

Figure 92: Answer of question 4

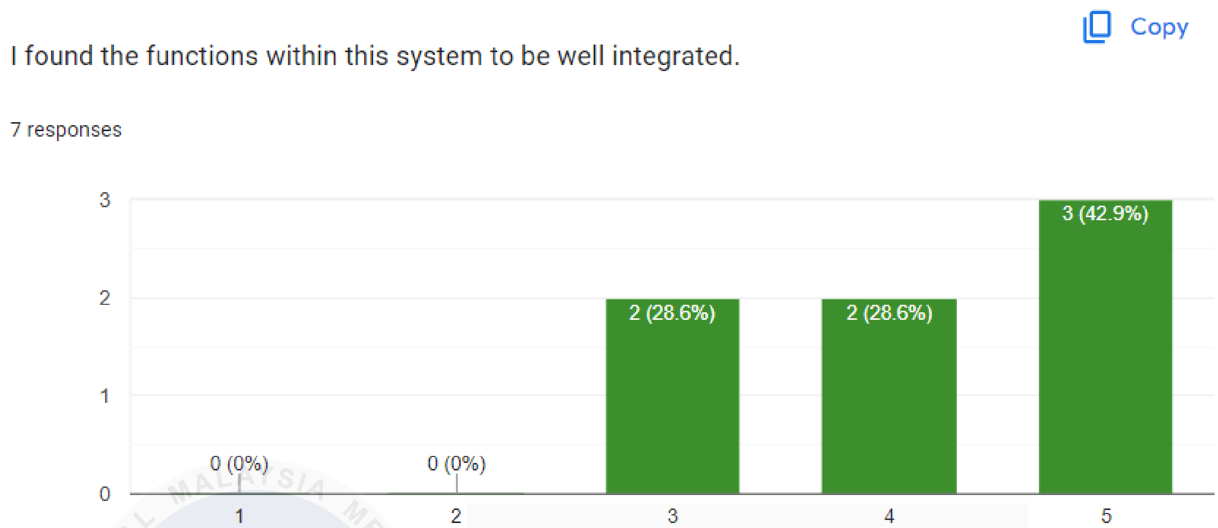

Figure 93: Answer of question 5

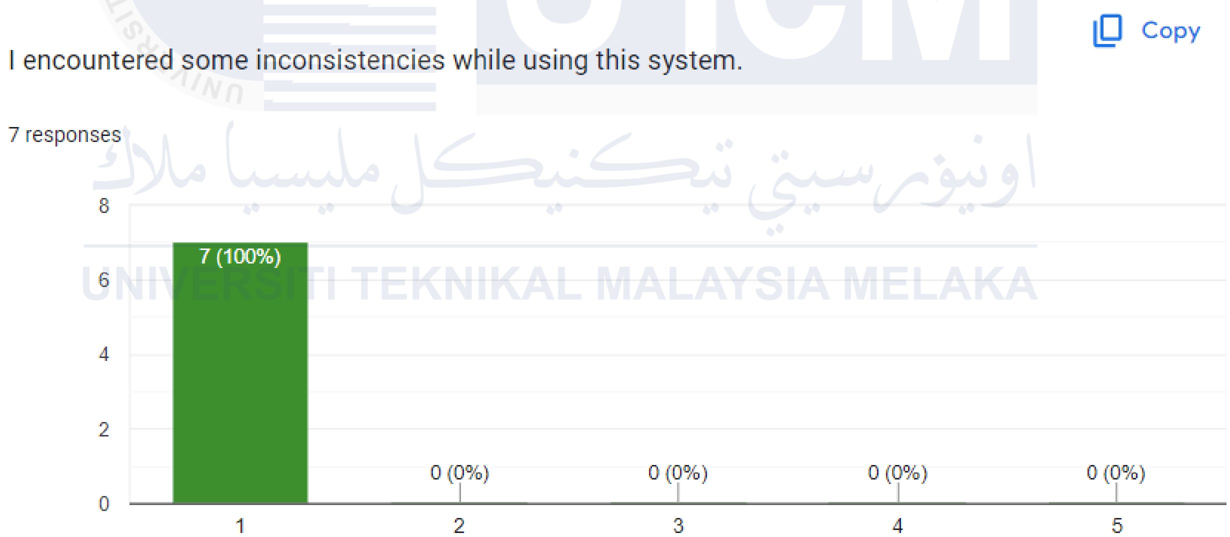

Figure 94: Answer of question 6

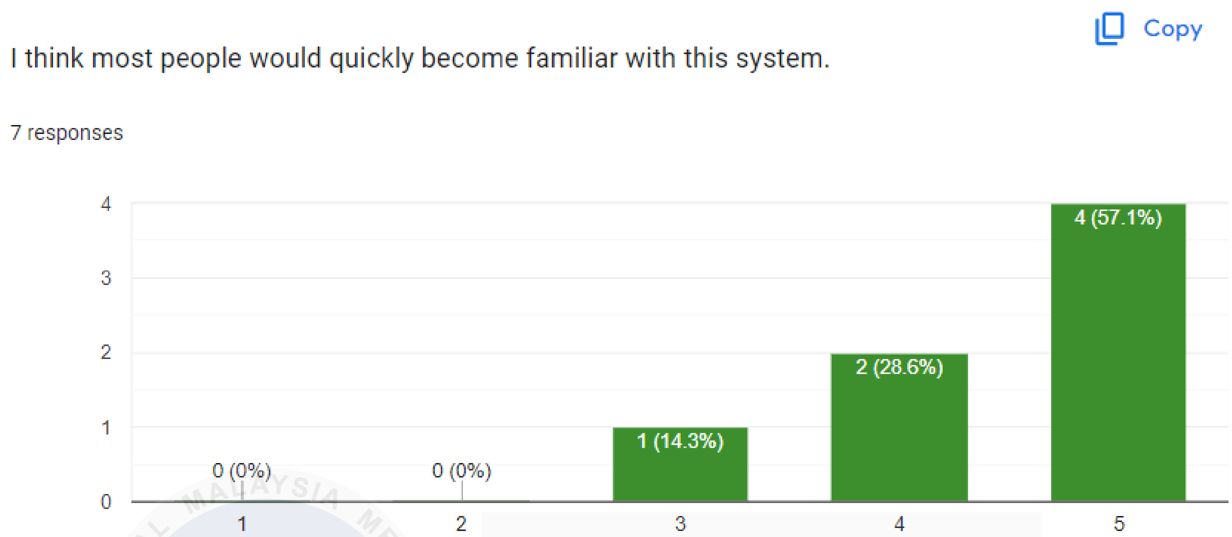

Figure 95: Answer of question 7

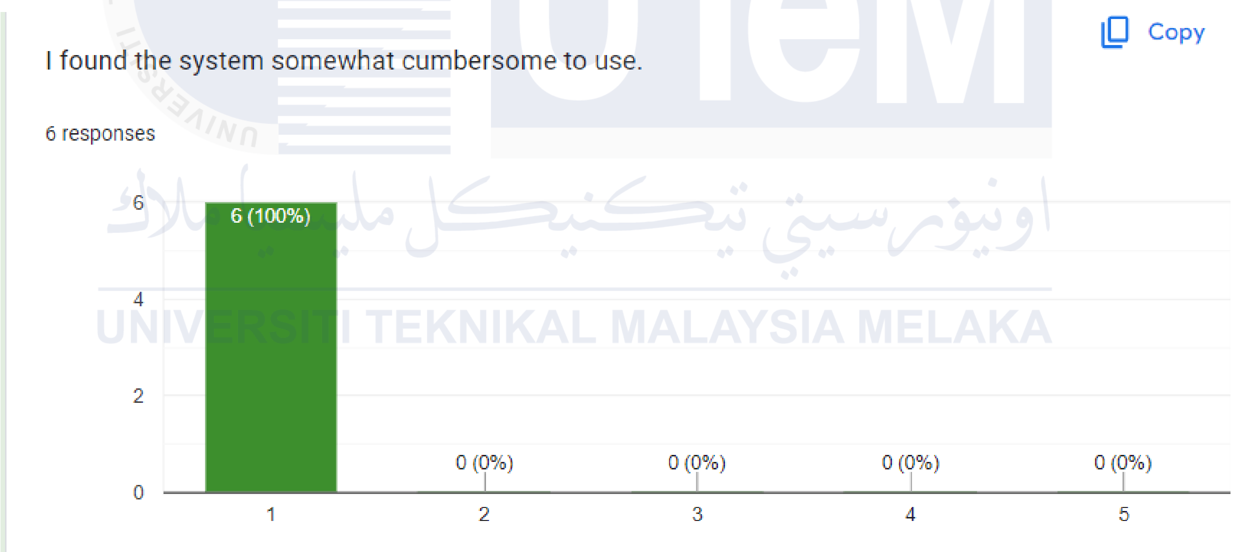

Figure 96: Answer of question 8

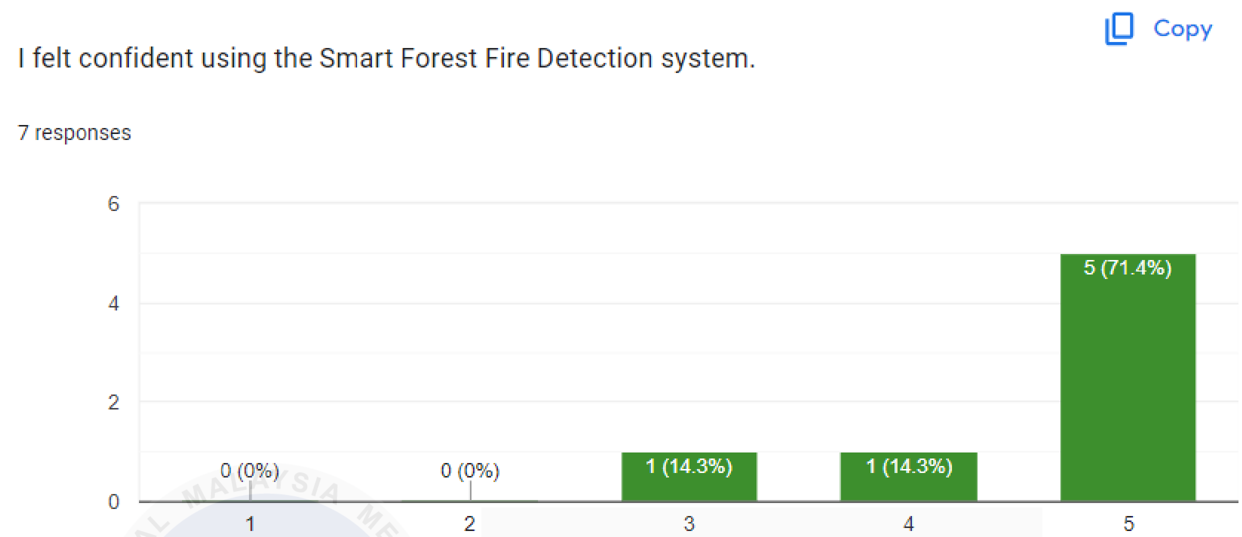

Figure 97: Answer of question 9

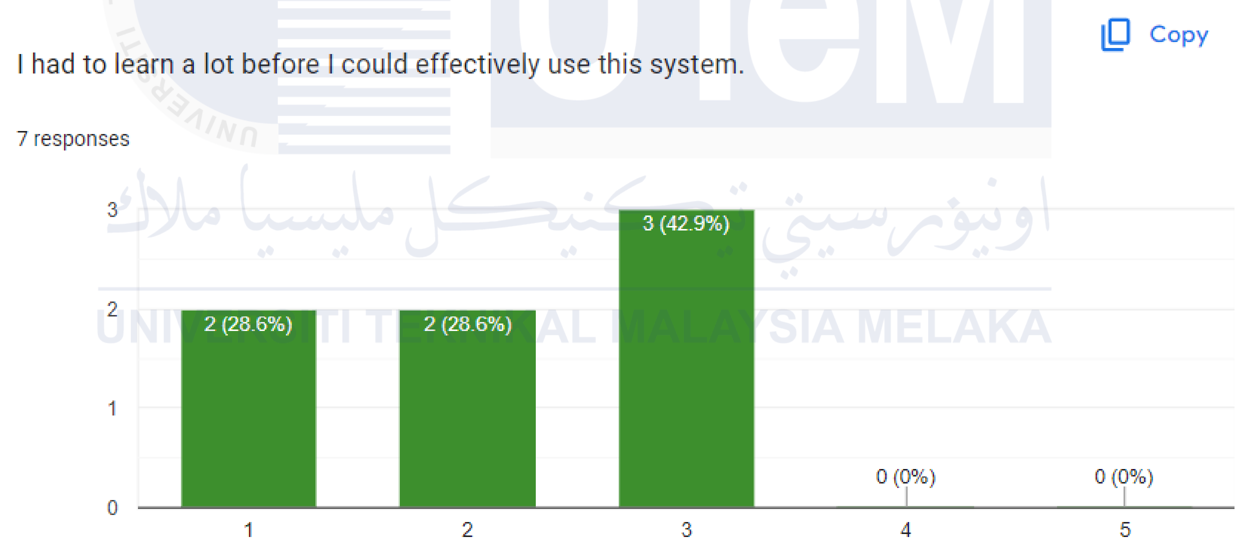

Figure 98: Answer of question 10

6.6.2 Interpreting Score

To determine the SUS score, a response scale with a range of 1 to 5 is utilised. Since the odd-numbered questions indicate positive sentiments and the even-numbered ones convey negative ones, add up the scores for all six respondents' odd-numbered questions (1, 3, 5, 7, and

9) and deduct 30 from the total to obtain (X). For each answer to an odd statement, deduct one from the SUS score. As there are five odd-numbered questions for every six responses in this instance, the total score for odd-numbered questions is deducted by thirty. Next, collect the points for all even-numbered questions (2, 4, 6, 8, 10) and deduct that sum from 150 to obtain (Y). To understand this, deduct the appropriate values from 5 in each even-numbered statement. In this case, there are 5 even-numbered questions for each 7 respondents, overall 150 is subtracted by the total of even-numbered questions. Lastly, Add up the total score of the new values (X+Y) and multiply by 2.5.

X = Total odd - Numbered Questions – 30
= (30+26+29+31+32) -30
= 148-30 = 118
Y = 150 – Total even-numbered questions
= 150 – (8+11+7+7+15)
= 150- 48 =102

Total score = (X+Y) XX 2.5 600 x 100

= (118 + 102) x2.5 x100
600

= 220 x 2.5
= 550
= 550/600 x 100
= 91.6

When using the scoring method described above, the SUS score is calculated out of 100, resulting in a score of 91.6/100. However, this is not a percentage score. As noted by UsabiliTEST (2011), it is crucial to understand that raw scores should not be interpreted as percentages and must be normalized for accurate percentile rankings. SUS scores can also be translated into letter grades, which provide a clearer understanding of the results. Table 17 below shows the SUS score grading.

Table 17:SUS score grading

SUS Score	Letter Grade	Adjective Rating
Above 80.3	A	Excellent
Between 68 and 80.3	B	Good
68	C	OK
Between 51 and 67	D	Poor
Below 51	E	Awful

For the SUS score for Smart Forest Fire Detection is above 80.3. It shows that the Smart Forest Fire Detection get an A which is an excellent rating.

6.7 Conclusion

In conclusion, the Smart Forest Fire Detection application's goals were all accomplished. This comprises creating a tool prototype, testing the tool's efficacy, and building the system with a clear, simple, and useful design. To make sure the prototype worked as anticipated, feature testing was done. After a thorough analysis of the data, certain settings were taken into consideration while drawing findings. The project's overall overview, contributions, limitations, and a thorough description of its development will all be covered in the upcoming chapter.

CHAPTER 7

PROJECT CONCLUSION

7.1 Introduction

This chapter provides a comprehensive overview of the Smart Forest Fire Detection project, summarizing the key outcomes, contributions, limitations, and future activities. It reflects on how the project's objectives were achieved, the methods used for development and testing, and offers insights into areas for improvement and further development.

7.2 Project Summarization

Smart Forest Fire Detection is a project designed to monitor and alert users about potential forest fires using an ESP32 microcontroller equipped with flame and smoke sensors. The system delivers real-time alerts and updates through the Blynk, leveraging the ESP32's powerful processing capabilities and built-in Wi-Fi. When connected to Wi-Fi, users can receive accurate information about smoke levels and fire conditions directly on their mobile phones.

The project aims to achieve three key objectives: first, to study and design methods for delivering real-time alerts and updates about forest fire incidents via an application; second, to develop an application that provides users with accurate information such as forest fire smoke levels; and third, to test the features of the Smart Forest Fire Detection application. The ESP32 is programmed to monitor and control the sensors, ensuring smooth operation and reliable performance of the system. The successful integration of hardware and software confirms that the Smart Forest Fire Detection system meets its objectives effectively.

i. Project Weakness

- Limited Connectivity

  The system relies on Wi-Fi connectivity for real-time alerts and updates. If the Wi-Fi connection is weak or lost, the user may not receive timely notifications about fire incidents, which could delay the response time.

- Power Consumption

  The system is designed to operate continuously to monitor forest fire conditions. However, it requires a consistent power supply to function effectively. Any disruption in power could cause the system to fail, resulting in a lack of monitoring and alerts.

- Hardware limitations

  This project does not include advanced features like a camera for visual confirmation of fire incidents. As a result, users rely solely on sensor data, which may not provide a complete picture of the situation, potentially leading to false alarms or missed incidents.

ii. Project Strengths

- Easy Installation and Maintenance:

  There aren't many components needed for the system, making maintenance and installation simple. Because of its simplicity, the system can be widely implemented in forested areas at a lower total cost.

- Real-Time Monitoring and Alerts:

  Users are guaranteed to receive prompt notifications about potential fire accidents due to the system's real-time monitoring and alerting features. This makes it possible to respond to forest fire risks more quickly.

- Remote Accessibility:

The system can be accessed remotely via the Blynk app, allowing users to monitor forest conditions and receive alerts from anywhere. The user-friendly interface of the app makes it easy for users to navigate and respond to alerts effectively.

7.3 Project Contribution

The Smart Forest Fire Detection system is designed to provide critical support in monitoring and responding to potential forest fires. By enabling real-time detection and alerts, the system empowers users to act quickly, potentially saving vast areas of forest and wildlife. The IoT-based solution leverages the ESP32 microcontroller and is controlled via the Blynk app, which allows users to monitor forest conditions remotely, ensuring they are informed even when they are not physically present. The project is cost-effective, with essential features that provide accurate smoke and fire detection. The use of flame and smoke sensors ensures that users receive timely notifications, allowing them to take preventive measures. This project contributes significantly to environmental protection efforts by offering a reliable and accessible tool for forest fire management. Table 18 shows that the project objective references for all chapter.

Table 18: Project objective references

PS	**PQ**	**PO**	**Project Objective**	Reference (chapter)
PS1	PQ1	PO1	To study and design existing methods of delivering real-time alerts and updates about forest fire incidents via an application.	Chapter 2
		PO2	To develop an application of providing users withaccurate information such as forest fire smoke level.	Chapter 4

		PO3	To test features of Smart forest fire detection applications.	Chapter 6

7.4 Project Limitation

The primary limitation of the **Smart Forest Fire Detection** system is its dependence on a stable power supply to keep the sensors and the ESP32 microcontroller operational continuously. Any power disruption could hinder the system's ability to monitor and alert users, potentially leading to missed fire incidents. Additionally, the system requires a reliable Wi-Fi connection for sending alerts through the Blynk app. If the internet connection is weak or lost, there could be delays or failures in alerting the user, reducing the system's effectiveness in preventing or responding to fires. The absence of a visual confirmation feature, such as a camera, limits the system's ability to provide comprehensive situational awareness, relying solely on sensor data for decision-making.

7.5 Future Works

The Smart Forest Fire Detection project can be further developed by enhancing its features and introducing new implementations. Here are some potential future improvements to consider:

- **Integration of a Camera for Visual Confirmation:** Adding a camera to the system would allow users to visually confirm the presence of fire and assess the situation remotely. This addition would provide an extra layer of verification, reducing the chances of false alarms and giving users more accurate information.

- **Incorporating a Backup Power Source:** To ensure the system's reliability during power outages, a secondary power source, such as a battery backup or solar power, could be integrated. This would allow the system to continue functioning and monitoring for fires even during power disruptions.

- **Enhancing Sensor Accuracy and Range:** Future iterations of the project could include more advanced sensors with greater accuracy and range. This would allow for earlier detection of fires and better coverage of larger areas, making the system even more effective in preventing large-scale forest fires.

7.6 Conclusion

The **Smart Forest Fire Detection** project has been successfully completed and thoroughly tested. All the project's objectives have been met, and the system is functioning effectively, providing real-time alerts and monitoring to help prevent forest fires. This project offers significant benefits in protecting forests and wildlife by enabling rapid response to potential fire incidents. By providing a reliable and accessible tool for forest fire detection, the project addresses a critical environmental need. As the project continues to evolve with future enhancements, it holds the potential to become an even more powerful solution in the fight against forest fires.

REFERENCE

1. Arduino.cc. (n.d.). *Windows tutorial: Getting started with Arduino IDE 1.x*. Retrieved September 2, 2024, from https://docs.arduino.cc/software/ide-v1/tutorials/Windows/
2. Asenware. (n.d) Fire alarm control panel. Retrieved 05th March 2019 from https://asenware.en.made-in china.com/product/hXyJWoAThVcp/China4-Zones-ConventionalFire-Alarm-Control-Panel-withCompetitivePrice.html.
3. Asenware. (n.d) Fire alarm control panel. Retrieved 05th March 2019 from https://asenware.en.made-in china.com/product/hXyJWoAThVcp/China4-Zones-ConventionalFire-Alarm-Control-Panel-withCompetitivePrice.html.
4. Ansori, A. R. (2023, December 10). IOT-based fire detection tool using the BLYNK app. Hackster.io. https://www.hackster.io/achmadridho/iot-based-fire-detection-tool-using-the-blynk-app-fc0138
5. Internet of things-based Fire Alarm Navigation System. (n.d.). https://downloads.hindawi.com/journals/misy/2022/3830372.pdf
6. Tutoriales, C. (n.d.). Rogerbit. https://rogerbit.com/wprb/2018/01/sistema-de-alerta-de-incendios-forestales-gps-con-sim808-y-arduino-uno/#google_vignette
7. Schroll, R. C. (2007, December 1). *Fire detection and alarm systems: A brief guide*. *Occupational Health & Safety*. https://ohsonline.com/Articles/2007/12/Fire-Detection-and-Alarm-Systems-A-Brief-Guide.aspx
8. Wikimedia Foundation. (2024, March 6). Fire alarm system. Wikipedia. https://en.wikipedia.org/wiki/Fire_alarm_system
9. IOT based fire detection and automatic ... - IJEAST. (n.d.-b). https://www.ijeast.com/papers/312-317,Tesma612,IJEAST.pdf
10. Gupta, V. (2024, February 28). Forest fire detection using sensor network and IOT. PsiBorg Technologies Pvt. Ltd. https://psiborg.in/forest-fire-detection-using-sensor-network-and-iot/
11. *Forest monitoring and wildland early fire detection by a hierarchical wireless sensor network*. (n.d.). https://www.researchgate.net/publication/294577058_Forest_Monitoring_and_Wildland_Early_Fire_Detection_by_a_Hierarchical_Wireless_Sensor_Network
12. *Forest monitoring and wildland early fire detection by a hierarchical wireless sensor network*. (n.d.-a).

https://www.researchgate.net/publication/294577058_Forest_Monitoring_and_Wildland_Early_Fire_Detection_by_a_Hierarchical_Wireless_Sensor_Network
13. *Forest monitoring and wildland early fire detection by a hierarchical wireless sensor network*. (n.d.-b). https://www.researchgate.net/publication/294577058_Forest_Monitoring_and_Wildland_Early_Fire_Detection_by_a_Hierarchical_Wireless_Sensor_Network

APPENDICES

APPENDICES A

i. The figure in 99 show that source code of the ESP32 and connect to Blynk

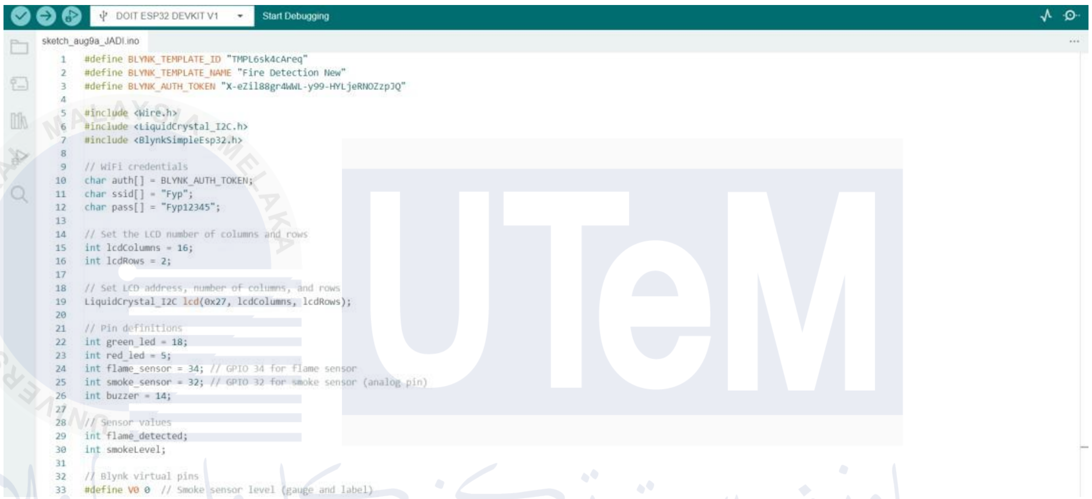

Figure 99: Source code ESP 32 connect with blynk

ii. The figure 100 show the source code virtual pin that we set in the blynk to connect with ESP32

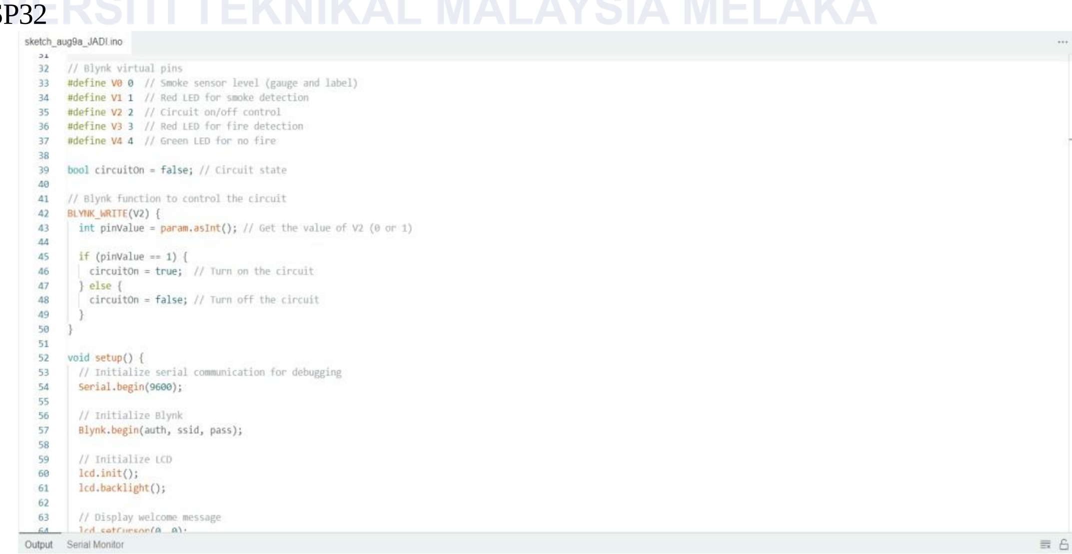

Figure 100: Source code virtual pin from blynk

iii. The figure 101 show the source code to get the notification from blynk to user mobile phone.

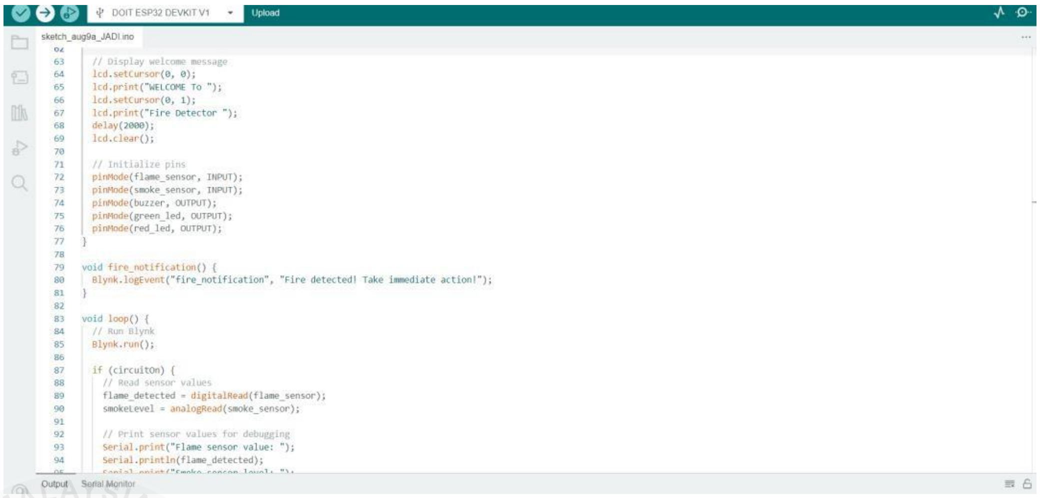

Figure 101: Source code notification from blynk to user

iv. The figure 102 show that the source code for flame and smoke sensor in arduino IDE

Figure 102: Source code flame and smoke sensor