IOT-BASED SMOKE ALARM SYSTEM

HO EN XIANG

UNIVERSITI TUNKU ABDUL RAHMAN

IOT-BASED SMOKE ALARM SYSTEM

HO EN XIANG

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Electrical and Electronic Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2019

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature	:	
Name	:	HO EN XIANG
ID No.	:	1402282
Date	:	

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"IOT-BASED SMOKE ALARM SYSTEM"** was prepared by **HO EN XIANG** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours.) Electrical and Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _________________________

Supervisor : Mr See Yuen Chark

Date : _________________________

Signature : _________________________

Co-Supervisor : _________________________

Date : _________________________

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Mr See Yuen Chark for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement throughout the entire process.

ABSTRACT

Fire can be made useful for various purposes. However, uncontrollable fire may result in property damage and human death. The major factor of fire deaths is due to excessive smoke inhalation. Therefore, early detection of fire is crucial in fire detection systems. The conventional fire detection system does not come with a false alarm prevention system. Besides, the system is unable to tell the exact location of the fire. In this project, an Internet of Things (IoT) – based smoke alarm system using Message Queuing Telemetry Transport (MQTT) communication protocol was developed to overcome these problems. The proposed system consists of three major parts which are the detector, processing unit and surveillance. The detector unit is an integration of ESP32, carbon monoxide sensor, ionisation smoke detector, buzzer, temperature and humidity sensor. As the processing unit, Raspberry Pi is used to run Node-RED application which processes the data and performs monitoring. A surveillance unit is where a camera is installed to monitor the condition of the surrounding. The response of the system is based upon the sensor's values or the user's response. Once it is confirmed as a fire breakout, the system will immediately sound the alarm and the Global Positioning System (GPS) coordinates and the accommodation floor plan will be sent to the nearby fire station. A Floor plan is developed to track the exact location of the fire. Experiments are carried out on the proposed system, and encouraging results are produced. Accordingly, where the system is not only able to detect the fire in real time, but also allows the user or related party to monitor the involved scene through surveillance camera with the assistance of the floor plan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

3G	Third Generation of Wireless Mobile Communications
BLE	Bluetooth Low Energy
CO	Carbon Monoxide
CPU	Central Processing Unit
DWG	AutoCAD Drawing Database
ETSI	European Telecommunications Standards Institute
FACP	Fire Alarm Control Panel
FRDM	Fire and Rescue Department of Malaysia
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HA	Home Assistant
IBM	International Business Machines Corporation
ID	Identification
IDE	Integrated Development Environment
IFTTT	If This Then That
IoT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
LPDDR2	Low Power Double Data Rate 2
MATLAB	Matrix Laboratory
MCU	Microcontroller Unit
MQTT	Message Queuing Telemetry Transport
NFPA	National Fire Protection Association
NoSQL	Not Only Structured Query Language
OS	Operating System
PDF	Portable Document Format
RAM	Random Access Memory
RF	Radio Frequency

RFID	Radio-Frequency Identification
SD Card	Secure Digital Card
SDK	Software Development Kit
SMS	Short Message Services
SVG	Scalable Vector Graphics
TCP	Transmission Control Protocol
UI	User Interface
UMTS	Universal Mobile Telecommunication System
USB	Universal Serial Bus
UWB	Ultra-Wideband
VoIP	Voice Over Internet Protocol
WiMAX	Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Network

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Fire has been discovered by mankind many years ago. Until today, it is still important to humans in their daily lives. There are many purposes. A fire which is under control can be very useful. However, an uncontrolled fire might lead to destruction.

Figure 1.1 shows the total number of building fires from 2012-2016 in Malaysia. All these data are obtained from the annual reports provided by the Fire and Rescue Department of Malaysia (FRDM). It can be observed that there is not much decrease in the number of cases in these 5 years and the average is above five thousand. The types of buildings can be further broken down into few categories such as residential, retail, factory, institution, office, public place, and others. Among these categories, residential premises contributed to the highest building fire incidents each year.

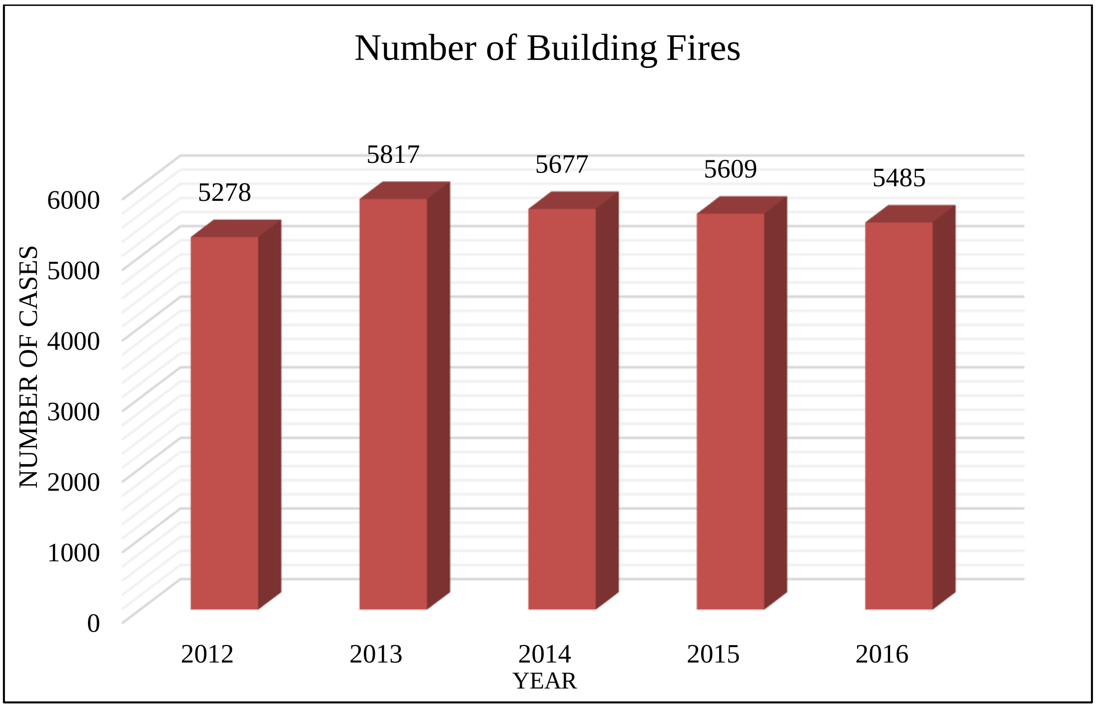


Figure 1.1: Building Fire Statistics (FRDM, 2019)

Referring to the National Fire Protection Association (NFPA) (2018), most fire deaths are caused by the inhalation of smoke rather than burns. Even before the victims are able to reach the exit of the building, they are already killed by the smoke itself. One of the reasons is that most of the oxygen was consumed by the fire itself during burning, which causes the oxygen level in the air to decrease. Besides, carbon monoxide may also be present which will displace oxygen in the blood. With the lacking of oxygen, the victims will initially feel some dizziness. When the oxygen content keeps on decreasing, the victims might fall into unconsciousness even facing the risk of death. Therefore, it is difficult for the victims to escape a fire especially when the fire occurred during night time when most of them are sleeping and unconscious of the fire. Table 1.1 shows the symptoms when the oxygen level in the air decreases.

Table 1.1: Oxygen Levels and Their Effects (NFPA, 2018)

When oxygen levels are at...	...a person experiences:
21 percent	Normal outside air
17 percent	Impaired judgment and coordination
12 percent	Headache, dizziness, nausea, fatigue
9 percent	Unconsciousness
6 percent	Respiratory arrest, cardiac arrest, death

This proved why smoke inhalation is the major factor for most of the fire deaths rather than burns. If smoke can be detected earlier, the victim will be able to have sufficient time to escape the building. Therefore, it is important to install fire safety system with smoke detectors which can provide early detection of fire and alert the victims.

There are various types of smoke detector available in the market. Two commonly used smoke detectors are photoelectric and ionisation smoke detectors. Both smoke detectors use different technology in smoke detection. This will be further discussed in Chapter 2.

1.2 Importance of the Study

According to the report from FRDM (2019), the total number of deaths caused by fire from 2012-2016 are shown in Figure 1.2.

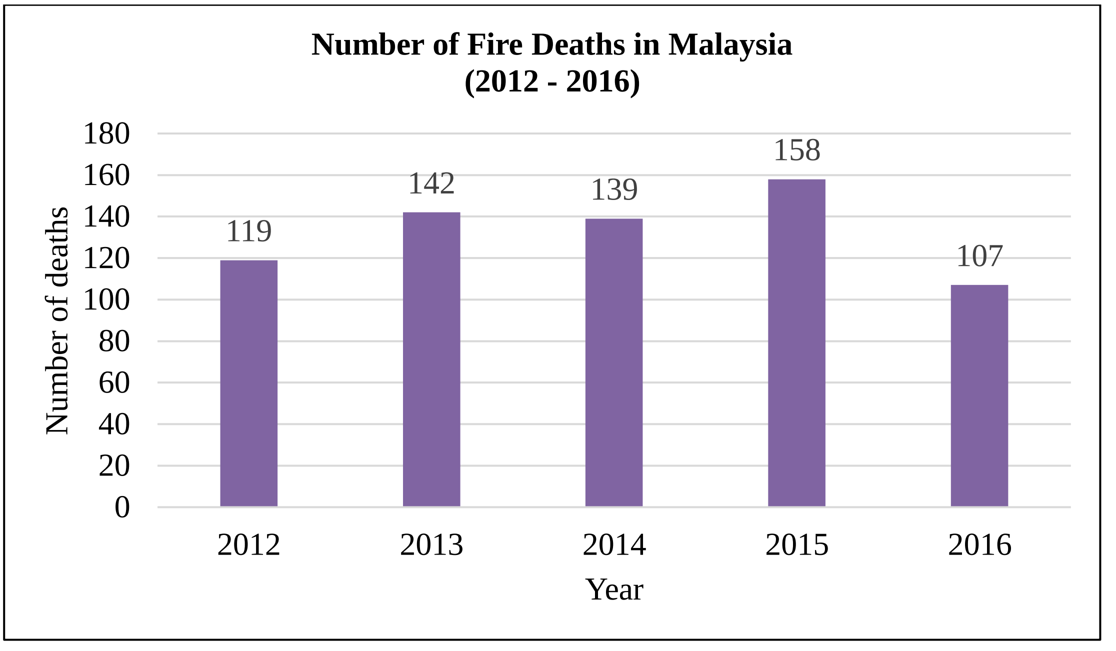


Figure 1.2: Statistics of Fire Deaths (FRDM, 2019)

Figure 1.3 shows the fire losses in Malaysia from 2012-2016. The total losses among these 5 years are estimated at about RM 13.1 billion.

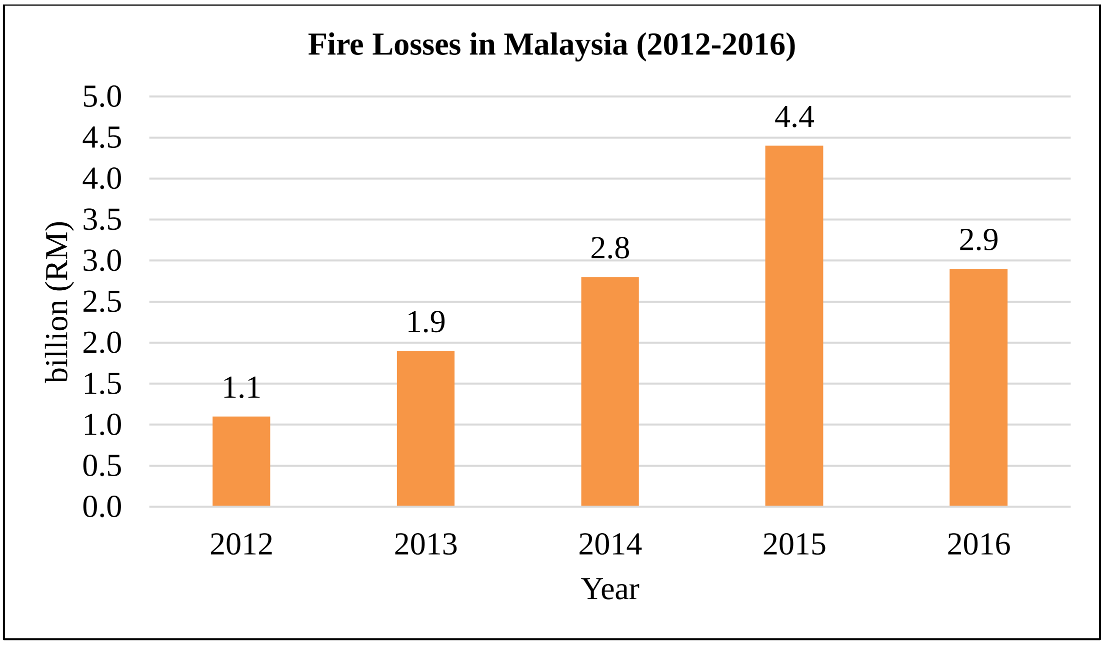


Figure 1.3: Statistics of Fire Losses (FRDM, 2019)

All the statistics presented in Figures 1.2 and 1.3 show the seriousness of the issue of building fire towards society. Not only it will threaten human life but also causes property damage. Therefore, it is important to develop a more reliable smoke alarm system which will be able to reduce the losses of property and life. In this project, the Internet of Things is used to create a comprehensive smoke alarm system which is able to monitor and alarm the user under various conditions. IoT is a network of things which are connected to the internet. A system equipped with IoT technology is smart enough that each device can communicate and share information with one another to carry out a certain action without any human intervention.

1.3 Problem Statement

A conventional fire alarm system consists of various smoke detectors or call points which are connected to the control panel using wires. All the components and devices in this system are controlled or monitored by the fire alarm control panel (FACP) based on zones.

According to Vijayalakshmi and Muruganand (2018), one of the disadvantages of the current fire detection system is not able to provide the location of the fire. It only provides the information on which zone is on fire.

Hasan and Razzak (2016) reported that the conventional fire detection system is not equipped with a false alarm prevention system. False alarm means the system is triggered by high humidity, dust or insects, chemical odors or cigarette smoke.

1.4 Aims and Objectives

The main objective of this project is to develop a smoke alarm system using IoT technology. The details of the objectives are:

i. To propose a system which is able to store data such as temperature, humidity and carbon monoxide concentration on a cloud storage system.
ii. To inform the authority about the location of the fire.
iii. To avoid false detection of fire.

1.5 Scope and Limitation of the Study

The study will focus on developing an IoT-based smoke alarm system. The process of developing the proposed system involved a combination of hardware and software. This are discussed as follow:

i. Hardware: Raspberry Pi 3, ESP32, Temperature and humidity sensor, buzzer, carbon monoxide (CO) sensor, Raspberry Pi Zero W, Raspberry Pi camera v1.3, and servo motor.
ii. Software: Arduino IDE, Home Assistant, Inkscape, Node-red, If This Then That (IFTTT), and Blynk.

The prototype consists of a processing unit, detector, and surveillance. This system allows the user to monitor the condition of their building remotely. Besides, the user will be informed through Voice Over Internet Protocol (VoIP) call when any fire or smoke is detected. The condition inside the building can also be monitored using the webcam.

The focus of this project is on the application of IoT in smoke alarm system. Therefore, the effectiveness and efficiency of the detector will not be taken into consideration. Besides, the quality of the camera used in this system will not be emphasized.

1.6 Contribution of the Study

In this thesis, a new concept of smoke alarm system based on the Internet of Things approach is discussed. The main contributions are as follow:

i. Review the existing smoke/fire alarm system.
ii. Identify the problems and challenges of using the conventional fire alarm system.
iii. Use Node-RED as a visual writing tool for IoT.
iv. Use Message Queuing Telemetry Transport (MQTT) protocol to provide communication between the detectors and processing unit.
v. Use Firebase Realtime Database to store data from sensors.
vi. Use camera for further identification of fire.
vii. Develop floor plan using Home Assistant (HA) to determine the location where fire occurred.

1.7 Outline of the Report

This thesis contains five chapters. The first chapter includes the background study, the problem faced currently and the objectives of this project. Chapter 2 is about the reviewing of related works as well as all the theories and concepts required for the achievement of this project. In chapter 3, which is the methodology, describes the overview of the proposed system, detailed circuit design and flowchart. Besides, the methods or ways used for the development of the system will be discussed. Then, the results will be discussed in chapter 4 and concluded in chapter 5.

CHAPTER 2

LITERATURE REVIEW

This chapter of the report presents the researches and studies related to the project as well as the technologies and methods applied. Important knowledge and theories necessary for the implementation of this project are also included.

2.1 Types of Fires

Fires can be categorized into two types – flaming fire and smouldering fire. The details are explained in the following subsections.

2.1.1 Flaming Fire

Flaming fire produces a large amount of flames but smaller amount of smoke. Usually, flaming fire is due to the ignition of materials such as flammable curtain, liquid or wood.

2.1.2 Smouldering Fire

A smouldering fire is a type of fire which is slow in spreading, low in temperature and flameless. It is a condition in which the material is being burned slowly with smoke but without flame. The smoke produced by such a fire may contain toxic chemicals such as cyanide and carbon monoxide.

2.2 Fire Triangle

Figure 2.1 shows the fire triangle which illustrates the elements required for combustion to take place. The three elements are fuel, heat, and oxygen. Fuel can be any kind of combustible material in any state. However, with the presence of fuel combustion will not occur unless there is the presence of heat. All combustible materials will have their own ignition temperature or ignition point. Thus, when the temperature requirement is met, it will ignite. The presence of heat is needed for the initial ignition of a fire and for the maintaining and spreading of the fire. There are many ways in which the heat source can be generated such as a hot object surface, friction, sunlight or sparks. Oxygen helps to support the chemical processes occurred during a fire. The oxygen content in the free air is about 21 %, and 16 % of oxygen is considered sufficient for combustion to take place. Without any one of the elements, a fire will not occur or will not last long because all of them are interdependent. Therefore, to stop or prevent combustion, the only need is to remove any one of the three elements. For a small fire, removal of fuel can be done naturally or manually. Oxygen supplied can also be reduced by using carbon dioxide fire extinguisher or a fire blanket. However, in the case of a building fire, removal of fuel and oxygen may not be an effective way to stop the fire. One common method used by the firefighters to extinguish a fire is water. Water is a kind of cooling agent which will help to lower down the temperature and therefore remove the heat.

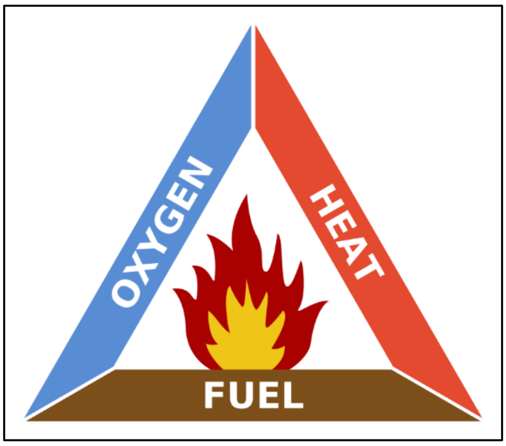


Figure 2.1: The Fire Triangle

2.3 Type of Smoke Detectors

The two most common smoke detectors used are photoelectric and ionisation smoke detectors. Ionisation smoke detectors are good in detecting flaming fires while photoelectric smoke detectors are better at detecting smouldering fires.

2.3.1 Ionisation Smoke Detector

Ionisation smoke detector mainly consists of a detector chamber and a small amount of radioactive element called Americium-241 with a half-life of 432 years. Such a radioactive material will emit radiation consisted of alpha particles. Under normal condition, electricity cannot flow through the air because air is considered as an insulator. However, when the alpha particles are produced, they will collide with the air particles inside the ionisation chamber. Such a collision causes the air particles to be ionised thus producing ions and electrons as shown in Figure 2.2. Ionisation is the process of liberating an electron from a gas atom or molecule to form a positively charged ion. The positive ion will move toward the negative electrode whilst the negatively charged electron to the positive electrode. Whenever there is the moving of electrons, there is current flow. As long as the radioactive material continues to emit the alpha particles, there will be a continuous flow of electric current.

However, when the smoke gets inside the chamber, it will interfere with the flow of the alpha particles. Fewer ions and electrons will be produced and therefore interrupts the current flow. When the current drops to a certain level, it will trigger the alarm. When the smoke is gone, the amount of current flows will return normal and the alarm will stop sounding. This type of smoke detector will be able to sense the small, invisible particles and therefore it provides better response toward a flaming fire.

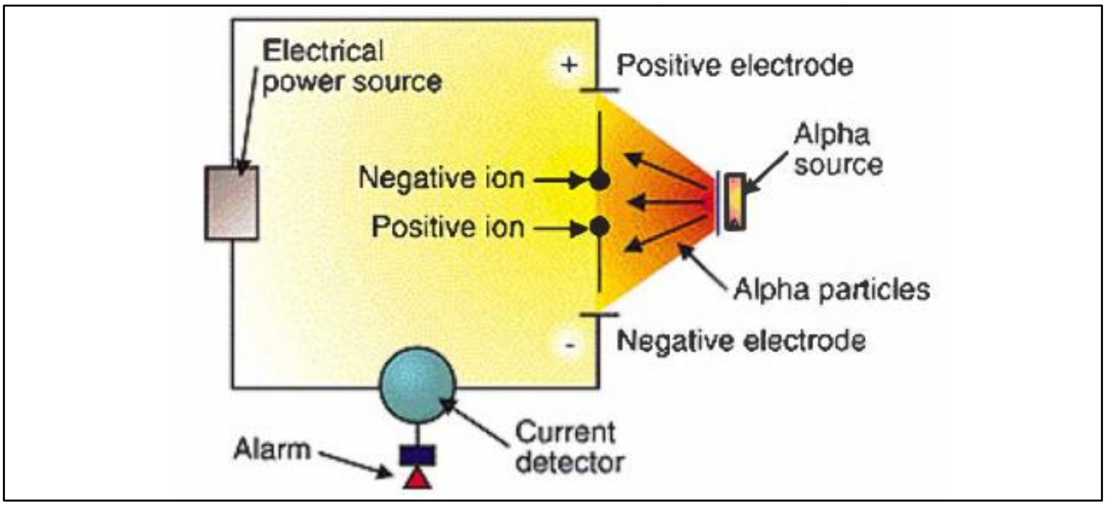


Figure 2.2: Circuit of an Ion Chamber Detector (Lipu et al., 2010)

2.3.2 Photoelectric Smoke Detector

Figure 2.3 shows the photoelectric smoke detector. There is a Light Emitting Diode (LED) and a photosensor inside the chamber. The function of the photosensor is to detect light. The purpose of LED is to generate a beam of light across the chamber but it is not directed towards the photosensor. When smoke enters the chamber, it will cause the light beam to be interrupted and scattered. Under this condition, some of the light might be scattered to the photosensor unintentionally. When the sensor detected a certain amount of light, it will trigger the alarm. The photoelectric smoke detector is better in detecting slow developing fire which produces more smoke instead of flames.

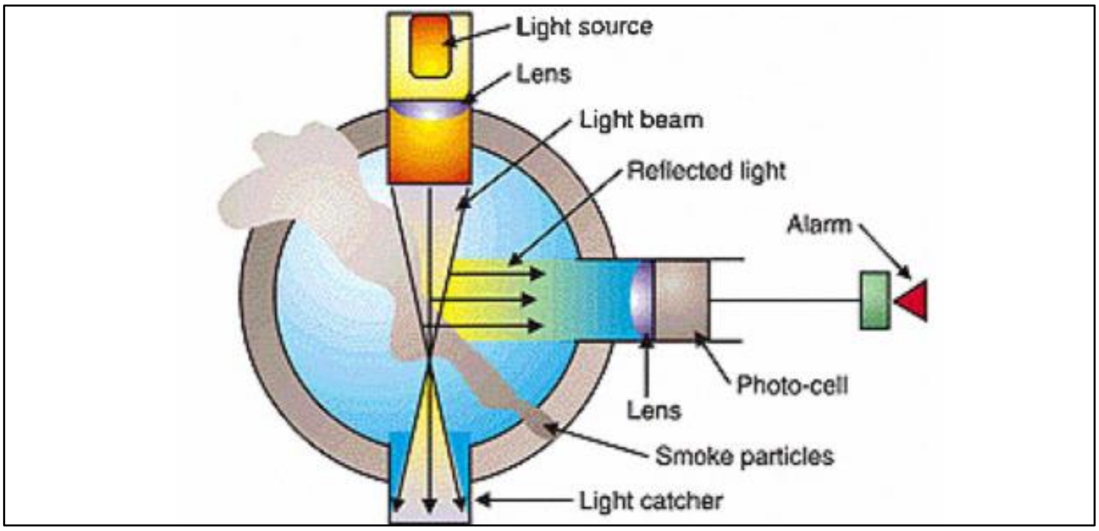


Figure 2.3: Photoelectric Smoke Detector (Lipu et al., 2010)

2.4 Related works

In this section, related works are reviewed and discussed according to the types of technology used.

2.4.1 Zigbee based Technique

Fuzi et al. (2014) developed a fire alert detection system utilising Zigbee wireless network. The system only has a temperature sensor for fire detection. When the sensor detected heat, it will send a signal to the Arduino Uno. The buzzer will be turned on when the temperature reached a pre-set value. The information will be sent to the laptop for monitoring via ZigBee wireless network using Xbee transmitter and Xbee receiver. However, one of the disadvantages is that the signal is limited to 10 m.

Another fire detection system was also developed by Islam et al. (2015) using the same wireless network. The system has a localizing capability. The authors used three sensors for the localization of a fire using Trilateration Technique. All the data are collected by the ZigBee network coordinator and sent to the Wi-Fi network using Arduino and Arduino Ethernet Shield. However, the authors stated that the system will have a relatively high initial cost.

2.4.2 Video or image processing

Instead of using sensors, a system that utilised camera image processing to detect a fire outbreak was implemented by Kwon et al. (2008). This was done by comparing the original image under the normal condition without fire and the current image captured. Fire can be detected based on different classification such as flame type, smoke type, and sparkling type.

Besides, a fire flame detection in surveillance video was developed by Kong et al. (2016). This system was implemented using logistic regression and temporal smoothing. A fire outbreak was determined based on the ratio of the colour component and pattern of the fire flame. Characteristics of fire such as size, motion, and colour information will be taken into consideration. Temporal smoothing was used to minimize the rate of getting a false alarm.

Another method for early fire detection based on video processing was done by using the smoke-pixel judgement consisted of two decision rules. The first one is the chromaticity-based static decision rule. The second one is the diffusion-based dynamic characteristic decision rule. The static decision rule was determined based on the greyish colour of the smoke during the burning process. The colour was categorized into two levels which are light grey and dark grey. For the dynamic decision rule, the spreading characteristics of the smoke were taken into consideration (Chen et al., 2006).

2.4.3 IoT based

An IoT based fire alarming system was developed by Imteaj et al. (2017). This system can detect fire as well as providing the location where the fire occurred. Raspberry Pi 3 was used as the main controller to control the Arduino which was integrated with sensors and camera. When a fire occurred, the system will send a message which includes the image captured and the Arduino’s location to the admin. The admin will be required to provide confirmation whether it is a fire outbreak. Once confirmed, the alarm will be raised and a message will be sent to the nearby fire brigade.

2.4.4 Others

A fire monitoring and control system was proposed by Gaikwad et al. (2016) making use of various sensors along with Global System for Mobile Communications (GSM) which is used to inform the user by sending messages. Besides, it also utilised the Global Positioning System (GPS) technology to locate the fire. When a fire occurred, the sensors will send a signal to the microcontroller ARM LPC2148. Then, the microcontroller will activate the emergency alarm as well as the water sprinklers for on-site control of fire. At the same time, the location information will be obtained using the GPS module and a Short Message Service (SMS) will be sent to the fire station. In this project, the authors did not take into consideration the possibility of having false detection.

Another system was developed by Asif et al. (2014) consisted of two sensors - temperature sensor and semiconductor sensor. The temperature sensor is to detect temperature beyond pre-set value while the semiconductor sensor is to detect smoke or gas. Both sensors were connected to the ATMega8L AVR microcontroller. The alert messages were sent using a SIM300CZ GSM kit-based network module. When any abnormality has been detected, the control unit will take actions by activating the local siren and the GSM module. Then the alarm message will be sent to the authority in the form of SMS. This system is quite similar to the one discussed previously but here the GPS module was not used.

2.5 Overview of IoT Architecture

Figure 2.4 shows the IoT architecture consisted of five layers.

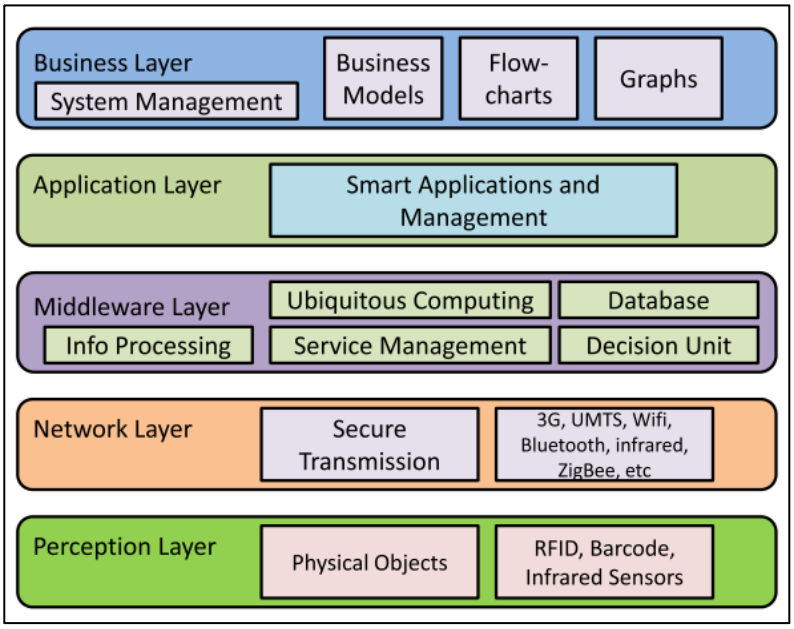


Figure 2.4: The IoT Architecture (Khan et al., 2012)

2.5.1 Perception Layer

The first layer is the perception layer or objects layer. This layer might consist of various sensors and actuators which obtain data or information from the environment. Various parameters can be measured such as pressure, temperature, position, acceleration, light intensity, humidity, motion, and many more. The information obtained from the physical environment will be converted into digital form for the use of digital devices.

2.5.2 Network Layer

This network layer can also be called as a “Transport Layer” or “Transmission Layer”. The function of this layer is to transfer the information from the sensor to the service management layer via secure channels. The information can be transported using wires or wirelessly by using technology such as Radio Frequency Identification (RFID), Third Generation Wireless Mobile Telecommunications (3G), GSM, Infrared, ZigBee, Bluetooth, etc.

2.5.3 Middleware Layer

The 3rd layer which is the middleware layer is also known as the processing layer. In this layer, all the information from the perception layer (which includes information from huge amounts of sensors) will be stored, analysed and processed.

2.5.4 Application layer

The application layer is on the management of the application based on the information or data being processed in the middleware layer. The purpose is to provide the services which can meet the customer’s needs. For example, the applications implemented can be smart home, smart city, smart healthcare or industrial automation.

2.5.5 Business Layer

The purpose of having the business layer is to manage the overall IoT system activities and services. A business model, flowchart, graph, etc can be built according to the data obtained from the application layer.

2.6 Wireless Network Protocol

There are few categories of wireless access technologies which are Wireless Metropolitan Area Network (WMAN), Wireless Wide Area Network (WWAN), Wireless Local Area Network (WLAN), and Wireless Personal Area Network (WPAN). The function of WPAN is to convey the information over short distances among devices usually about 10 m or less. Examples of protocols using this kind of technologies are Bluetooth, Zigbee and Ultra-wideband (UWB). WLAN can deliver information at a speed of about 200 Mbps up to 100 m. Wi-Fi is one of the examples of WLAN. WMAN is a network which provide communication between buildings or locations which is a few miles from one another within a region. The most popular wireless metropolitan area network is Worldwide Interoperability for Microwave Access (WiMAX), which can reach the speed of 70 Mbps over several kilometres. The network which can provide connectivity for a larger area is WWAN. The area can be between cities, or even countries. The technologies that used this kind of network are GSM, General Packet Radio Service (GPRS), and Universal Mobile Telecommunication System (UMTS).

2.6.1 Bluetooth

Even though Bluetooth has been implemented many years ago, it is still widely used in today’s wireless communication. Even recent smart watch technology also uses Bluetooth technology to communicate with smart phones. One of the reasons is that it only requires a small amount of power. However, it has low data rate which is about 1-2 Mbps and a connectivity range of 10 m or less.

2.6.2 ZigBee

ZigBee is known as a mesh Local Area Network (LAN) protocol operating on 2.4 GHz and it is mainly built for home automation. Due to this reason, it has been designed in such a way that it consumes very less power. However, the drawback of low energy consumption is the low data rate which is around 0.25 Mbps. In a mesh network, the connection is spreading out among intermediate devices. Therefore, data can be transmitted over longer distances.

2.6.3 Wi-Fi

Wi-Fi uses Internet Protocol (IP) to communicate between endpoint devices and LAN. In order to establish a Wi-Fi connection, it is required to have a wireless router which is connected to the network. One of the disadvantages of Wi-Fi is that it can be easily subjected to interference. It can be affected by microwave, cordless phone, and many other electronic equipment.

2.6.4 Wi-Max

Wi-Max is a technology which can provide long-range wireless networking. Usually, it will be used to provide portable mobile broadband connectivity across cities and countries. It has a longer data sending rate as well as longer connectivity range as compared to Wi-Fi.

2.6.5 GSM

GSM is a digital mobile telephone standard which has been developed by the European Telecommunications Standards Institute (ETSI). However, today GSM has already become the global standard for mobile communications.

2.7 Comparison of Characteristics for Different Wireless Protocols

Table 2.1 shows the comparison of characteristics for different wireless protocols which have been discussed in Section 2.6.

Table 2.1: Comparison of Characteristics for Different Wireless Protocols (Chakkor et al., 2014)

Protocols	**Bluetooth** [2], [14], [17], [18]	**UWB** [14], [19]	**ZigBee/IP** [2], [14], [17-23]	**Wi-Fi** [1], [2], [14], [24], [25]	**Wi-Max** [17], [25-28]	**GSM/GPRS** [29-33]
Frequency band	2.4 GHz	3.1-10.6 GHz	868/915 MHz; 2.4 GHz	2.4; 5 GHz	2.4; 5.1- 66 GHz	850/900; 1800/1900 MHz
Max signal rate	720 Kb/s	110 Mb/s	250 Kb/s	54 Mb/s	35-70 Mb/s	168 Kb/s
Nominal range	10 m	10-102 m	10 - 1000 m	10-100 m	0.3-49 Km	2-35 Km
Nominal TX power	0 - 10 dBm	-41.3 dBm/MHz	-25 - 0 dBm	15 - 20 dBm	23 dBm	0-39 dBm
Number of RF channels	79	(1-15)	1/10; 16	14 (2.4 GHz) 64 (5 GHz)	4;8 10;20	124
Channel bandwidth	1 MHz	0.5- 7.5 GHz	0.3/0.6 MHz; 2 MHz	25-20 MHz	20;10 MHz	200 kHz
Modulation type	GFSK, CPFSK, 8-DPSK, π/4-DQPSK	BPSK, PPM, PAM, OOK, PWM	BPSK QPSK, O-QPSK	BPSK, QPSK, OFDM, M-QAM	QAM16/64, QPSK, BPSK, OFDM	GMSK, 8PSK
Spreading	FHSS	DS-UWB, MB-OFDM	DSSS	MC-DSSS, CCK, OFDM	OFDM, OFDMA	TDMA, DSSS
Basic cell	Piconet	Piconet	Star	BSS	Single-cell	Single-cell
Extension of the basic cell	Scatternet	Peer-to-Peer	Cluster tree, Mesh	ESS	PTMP, PTCM, Mesh	Cellular system
Max number of cell nodes	8	236	> 65000	2007	1600	1000
Encryption	E_0 stream cipher	AES block cipher (CTR, counter mode)	AES block cipher (CTR, counter mode)	RC4 stream cipher (WEP), AES block cipher	AES-CCM cipher	GEA, MS-SGSN, MS-host
Authentication	Shared secret	CBC-MAC (CCM)	CBC-MAC (ext. of CCM)	WPA2 (802.11i)	EAP-SIM, EAP-AKA, EAP-TLS or X.509	PIN; ISP; Mobility Management (GSM A3); RADIUS
Data protection	16-bit CRC	32-bit CRC	16-bit CRC	32-bit CRC	AES based CMAC, MD5-based HMAC, 32-bit CRC	GPRS-A5 Algorithm
Success metrics	Cost, convenience	Throughput, power, cost	Reliability, power, cost	Speed, Flexibility	Throughput, Speed, Range	Range, Cost, Convenience,
Application focus	Cable replacement	Monitoring, Data network,	Monitoring, control	Data network, Internet, Monitoring,	Internet, Monitoring, Network Service,	Internet, Monitoring, control

2.8 MQTT

MQTT stands for Message Queuing Telemetry Transport. It is known as a lightweight publish or subscribe messaging transport protocol for machine-to-machine communications that runs on top of the Transmission Control Protocol (TCP) stack (Karagiannis et al., 2015). Initially, MQTT was developed by Andy Stanford-Clark and Arlen Nipper in 1999 which is mainly for the used in the oil and gas industry. However, until today it has been widely used in many applications such as home automation and Facebook messenger. MQTT is a many-to-many communication protocol thus enabling information exchange between multiple clients (Heđi et al., 2017). Besides, MQTT is suitable for constrained environments such as limited bandwidth, limited computation capability, and low power. Therefore, it is suitable to be implemented on the devices that run on battery (Yassein and Shatnawi, 2016). There are some important terminologies of MQTT which are publisher, client, broker, subscriber, and topic.

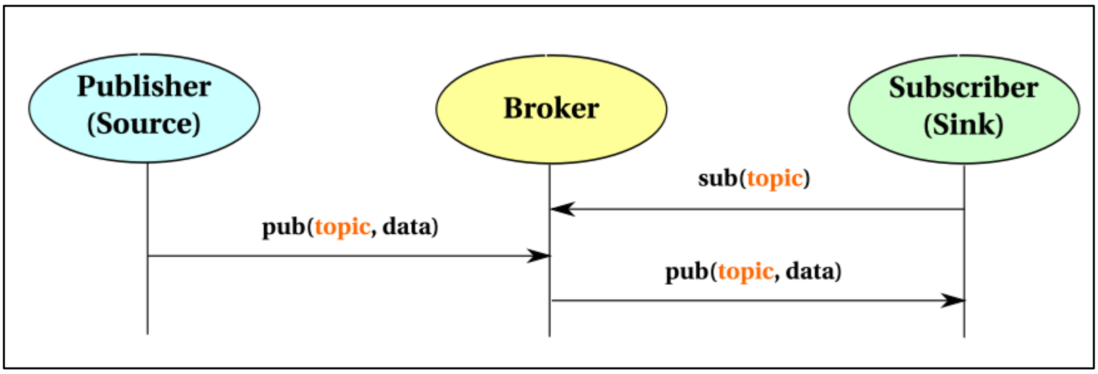


Figure 2.5: Communication Model (Hunkeler et al., 2008)

Figure 2.5 shows the communication model for MQTT protocol. Both the publisher and subscriber are called MQTT clients. A publisher refers to the client which publishes messages or information to the broker. A subscriber refers to the client which is subscribed to a topic in order to receive the message. MQTT client can be any devices as long as it is able to connect to the MQTT broker. There will be no direct communication or connectivity among the clients. All messages will be published to and send forth from the broker.

The MQTT broker is a software which function is to receive all the information and messages from the publishing clients. The, it will filter the messages and send the information to the subscribing clients. It is like the brain of the system. HiveMQ, AWS IoT, Mosquitto, and Mosca are some examples of MQTT brokers.

To communicate and transfer information between the clients via the MQTT broker, a 'topic' is very important. For example, device A wants to receive messages from device B. Now device B is the publishing client and device A is the subscribing client. For a successful transferring of information, device B and device A should be subscribed to the same topic.

CHAPTER 3

METHODOLOGY AND WORK PLAN

In this section, the methods and ways used for the development of the system are discussed. The lists of hardware and software required will also be explained.

3.1 Overview of the System

The proposed system consists of three major parts, namely the processing unit, detector, and surveillance. The block diagram of the system is shown in Figure 3.1. The detector is built by integrating the existing ionisation smoke detector with the temperature and humidity sensor (DHT11), carbon monoxide sensor (MQ-7), microcontroller (ESP32) and buzzer. ESP32 is responsible for obtaining the sensors' readings and triggering of the buzzer. The data collected from the sensors will be sent to the processing unit by ESP32 via Message Queuing Telemetry Transport (MQTT) communication protocol. The details regarding the communication protocol will be further explained in Section 3.2.

The primary processing unit used in this system is Raspberry Pi 3 Model B+ with Hass.io operating system (OS) running on it. Hass.io is an OS developed mainly for installing and updating Home Assistant (HA) – a python program with the ability to track, control and automate devices. Node-RED which is a programming tool for the Internet of Things (IoT) is installed in Hass.io. Node-RED is used to communicate with various services and devices. The information collected from the sensors is sent to the Node-RED and displayed on the dashboard in real time. At the same time, the values will also be stored into the Firebase Realtime Database. The processing unit will carry out responses according to the inputs received. The final decision of the system depends on both the sensor's readings and the user's response. The system flow will be further discussed in Section 3.5.

A phone call message will be delivered to the user's device via Voice Over Internet Protocol (VoIP) using the If This Then That (IFTTT) application and direct the user to the Node-RED dashboard. At the same time, the user can access the surveillance camera to identify whether it is a real fire or just a false detection. When the fire becomes uncontrollable, the Global Positioning System (GPS) location will be sent to the nearby fire brigade immediately.

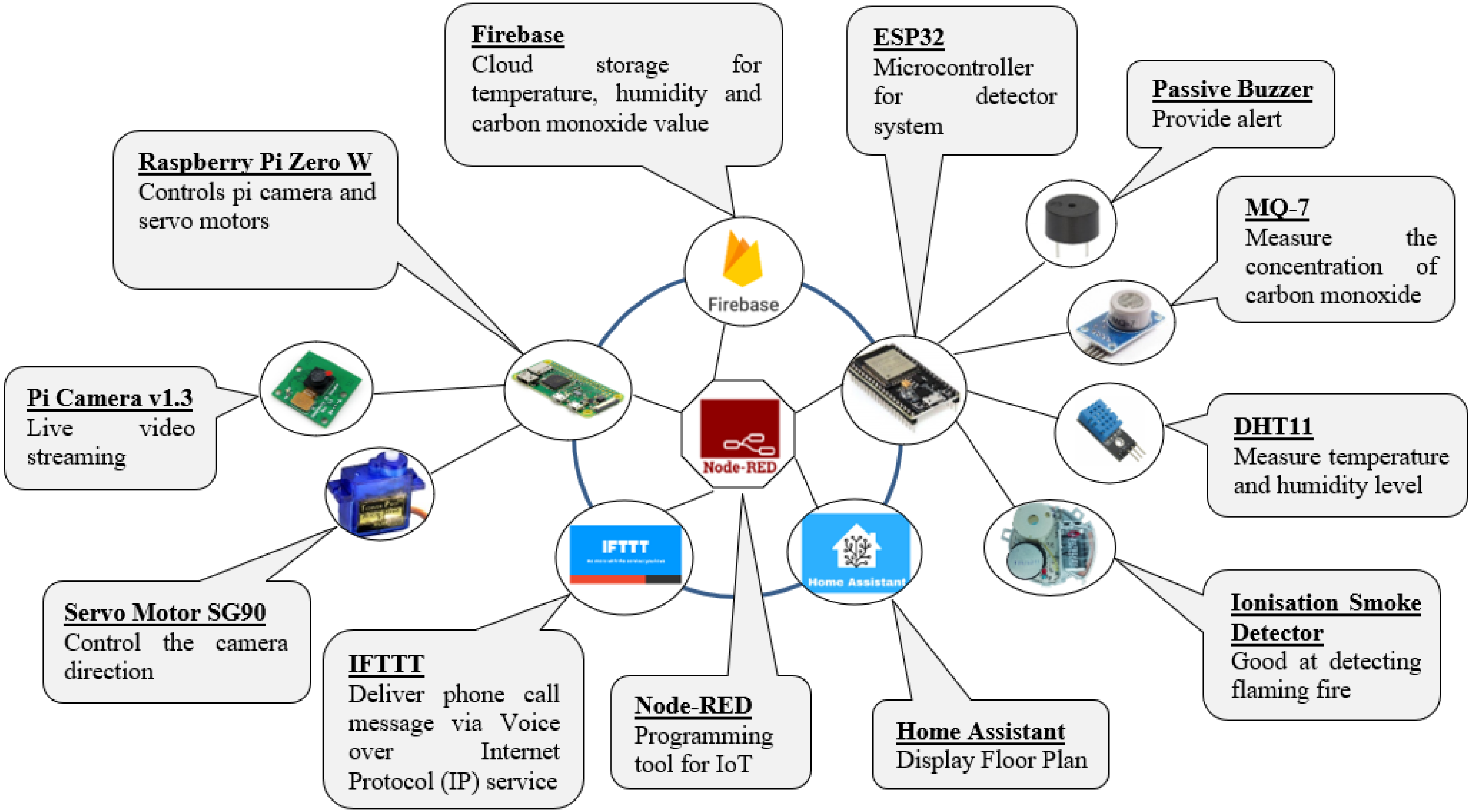


Figure 3.1: Block Diagram of the System

3.2 System Communication Model

Figure 3.2 shows the communication model of the system between ESP32 and Node-RED. Mosquitto broker will be installed on the Raspberry Pi to manage the information exchange among the clients. The red-coloured wordings represent the MQTT topics. ESP32 will obtain the values from the sensors and publish them on the topics subscribed by Node-RED. The information received will be updated and displayed on the Node-RED user interface regularly. When two or more sensors input is above the typical values, Node-RED will publish the message "on" to the topic "esp32/alarm" to tell the ESP32 to trigger the alarm.

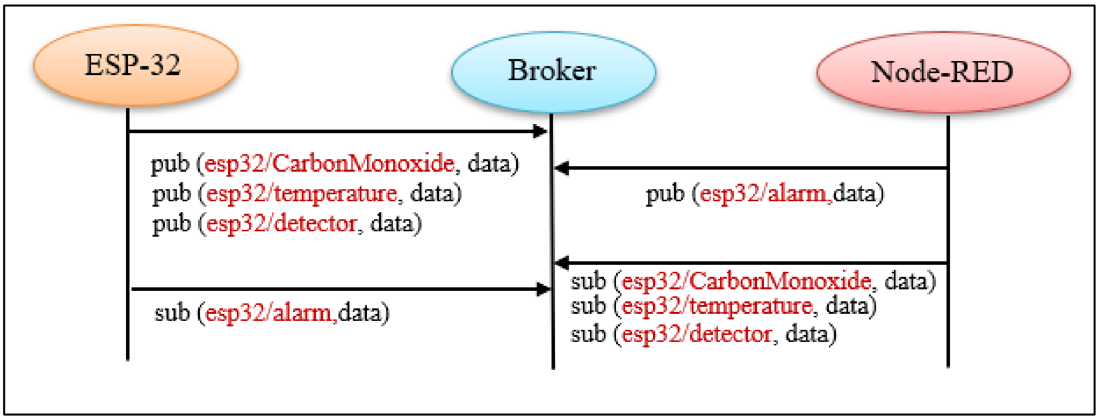


Figure 3.1: System Communication Model

3.3 Hardware

All the hardware used to develop the system are discussed in the following subsections.

3.3.1 Raspberry Pi 3 Model B+

As mentioned in the previous section, Raspberry Pi 3 Model B+ is used as the primary processing unit for the system. It is known as a powerful credit-card sized single board computer that can be used for various applications. The components of Raspberry Pi 3 are shown in Figure 3.3.

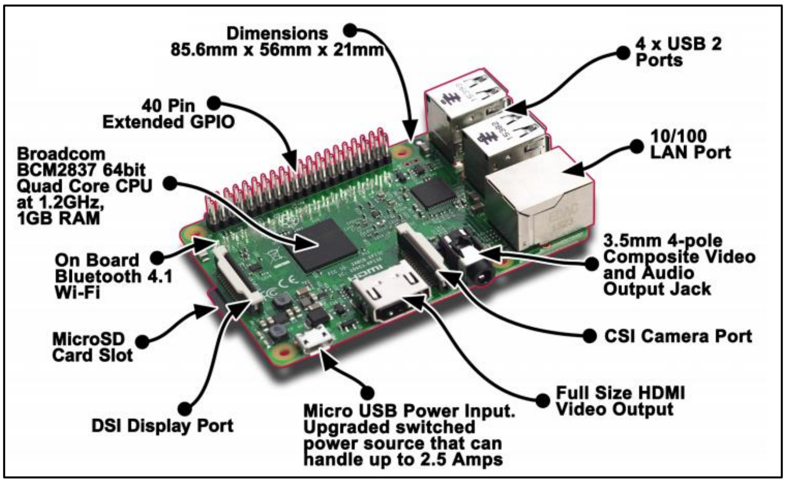


Figure 3.2: Raspberry Pi 3 Model B+

Figure 3.4 shows the core architecture used in this microprocessor. It has 1 GB of Low Power Double Data Rate 2 (LPDDR2) Random-Access Memory (RAM) and a Central Processing Unit (CPU) with a processing speed of 1.2 GHz. The OS used in this project is Hass.io. Figure 3.5 shows all the pinouts of the Raspberry Pi 3.

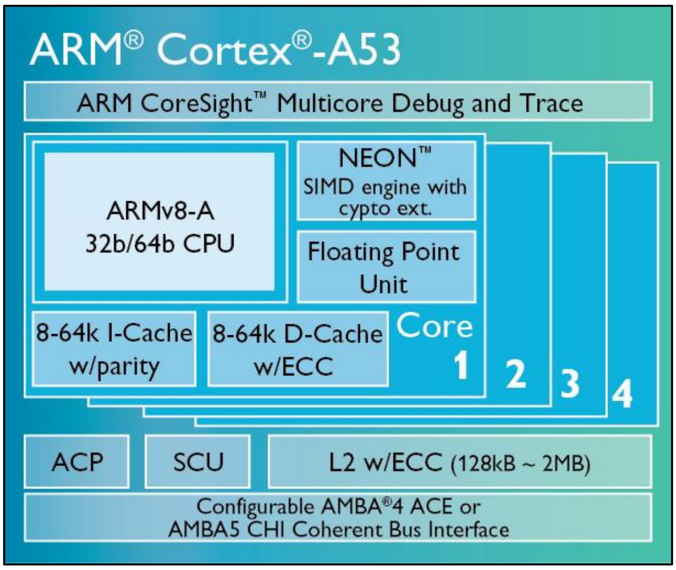


Figure 3.3: Raspberry Pi 3 Model B Core Architecture-Quad-Core ARM Cortex-A53

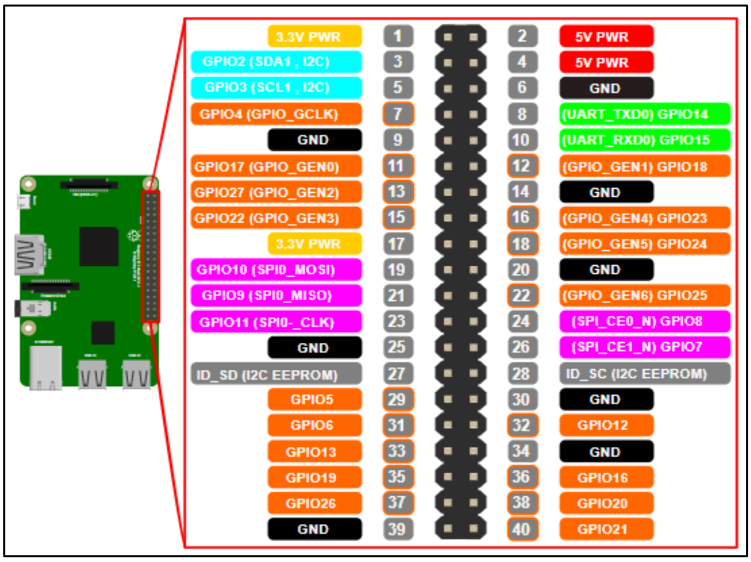


Figure 3.4: Raspberry Pi 3 Model B Pinouts

3.3.2 Raspberry Pi Zero W

Raspberry Pi Zero W is an improved version of the Pi Zero family which comes with added connectivity, consisting of wireless local area network (LAN), Bluetooth 4.1 and Bluetooth Low Energy (BLE). In the proposed system, it will be used to control the Raspberry Pi Camera v1.3 and servo motors. All the components and specifications of Raspberry Pi Zero W is shown in Figure 3.6 and Table 3.1 respectively.

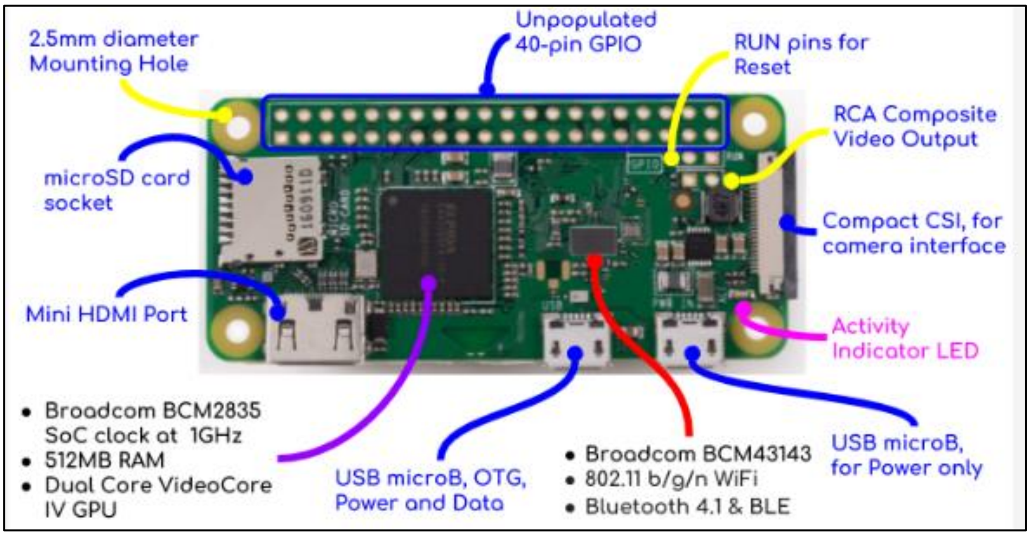


Figure 3.5: Raspberry Pi Zero W

Table 3.1: Raspberry Pi Zero W Specifications

Specifications	
Operating Voltage	5 V
Operating Current	2.5 A
Processor	BCM 2835 SOC
Clock Speed	1 GHz
RAM	512 MB
Built-in Wireless	BCM43143, WiFi + Bluetooth 4.1 + BLE (Bluetooth Low Energy)
Memory	Micro-SD
Display and Audio	Mini-HDMI
USB Port	1 × micro-B USB for data (with power too)
Power input	1 × micro-B USB for power (no data)
Camera interface	CSI camera connector
GPIO	Unpopulated 40-pin GPIO connector

3.3.3 Raspberry Pi Camera Board v1.3

The Raspberry Pi Camera is used to provide live video streaming to enable the user to monitor the conditions of surrounding. The specifications of the camera are shown in Table 3.2.

Figure 3.6: Raspberry Pi Camera Board v1.3

Table 3.2: Raspberry Pi Camera v1.3 Specifications

Specifications	
Weight	3 g
Still Resolution	5 Megapixels
Video modes	1080p30, 720p60 and 640 × 480p60/90
Sensor	OmniVision OV5647
Sensor resolution	2592 × 1944 pixels
Sensor image area	3.76 × 2.74 mm
Horizontal field of view	53.50 +/- 0.13°
Vertical field of view	41.41 +/- 0.11°

3.3.4 Servo Motor SG90

Servo motor is used to control the direction of the camera. The pinouts and specifications of the servo motor are shown in Tables 3.3 and 3.4.

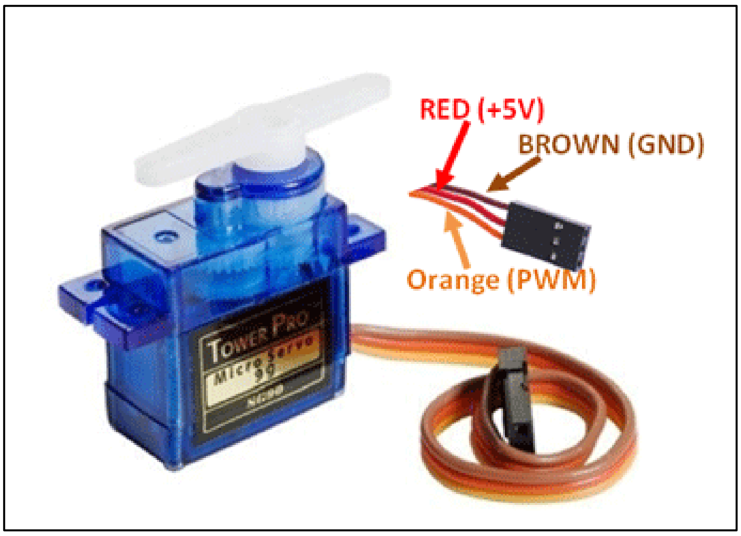


Figure 3.7: Servo Motor SG90

Table 3.3: Servo Motor SG90 Pinouts Description

Pin Colour	Description
Red	Connect to power supply 5 V
Orange	PWM signal is given in through this wire to drive the motor
Brown	Connected to the ground

Table 3.4: Servo Motor SG90 Specifications

Specifications	
Operating Voltage	+ 5 V typically
Torque	2.5 kg/cm
Operating Speed	0.1 s/60˚
Rotation	0˚ - 180˚
Weight	9 g

3.3.5 ESP32

ESP32 is an advancement of ESP-8266. It is a low-cost and lower power microcontroller integrated with Wi-Fi and dual-mode Bluetooth. Besides, it is also integrated with in-built antenna switches, Radio Frequency (RF) balun, filters, low-noise amplifier, power amplifier, and power management module. It can operate at a temperature between −40 °C to +125 °C. For the development of the system, ESP32-DevKitC development board of ESP32-WROOM-32 module will be used which is shown in Figure 3.9. ESP32 will obtain the data from the sensors and publish them to the MQTT topic in order to be received by Node-RED.

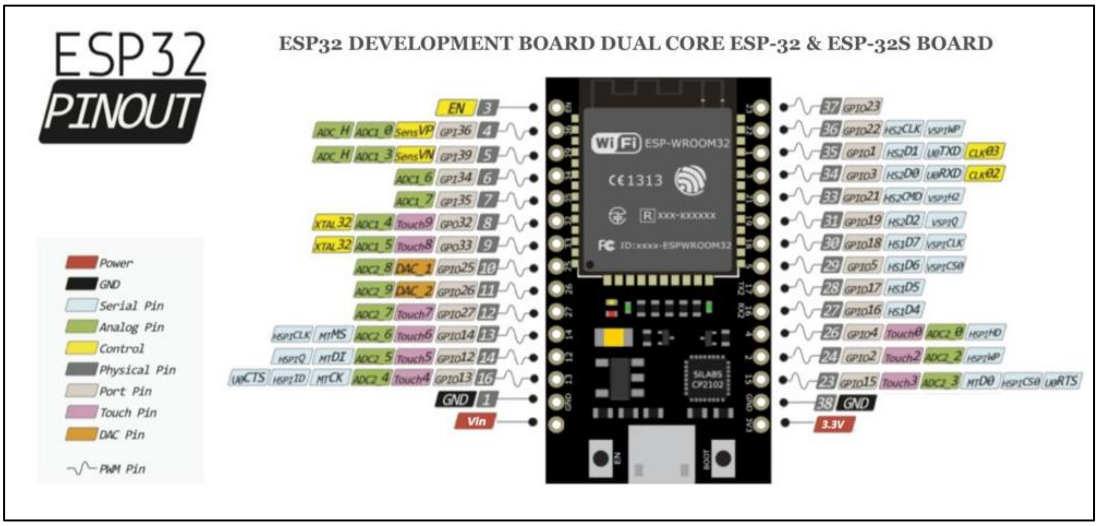


Figure 3.8: ESP32 (Espressif Systems, 2019)

3.3.6 Ionisation Smoke Detector

The ionisation smoke detector consists of 1.0 microcuries of Americium-241. It is powered up with 9 V battery. The circuit board of the smoke detector is shown in Figure 3.10.

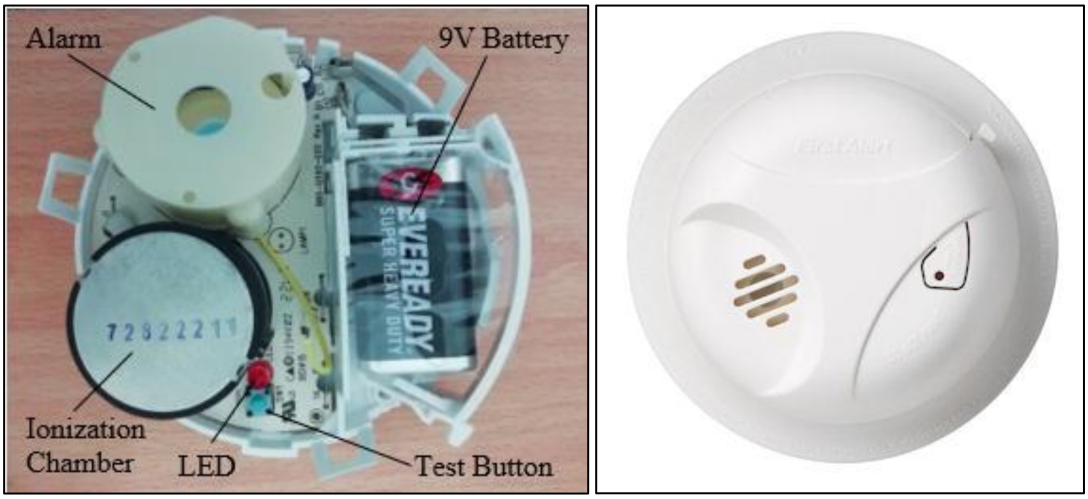


Figure 3.9: Ionisation Smoke Detector

3.3.7 DHT11 Temperature and Humidity Sensor

DHT11 is used to measure the temperature and humidity levels of the surroundings. The values will be sent and displayed on the Node-RED dashboard by the ESP32.

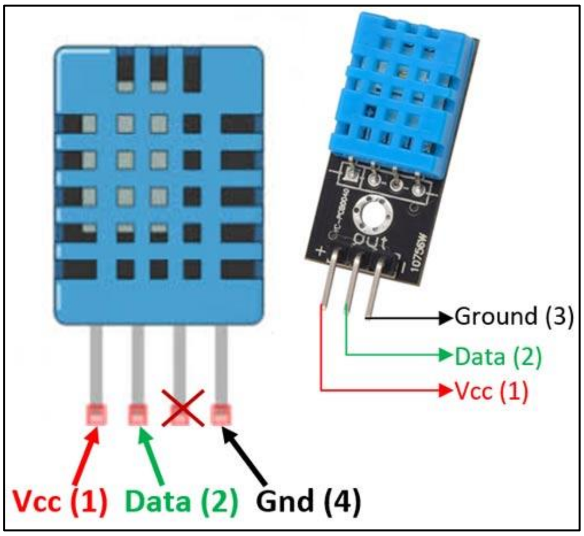


Figure 3.10: DHT11 Temperature and Humidity Sensor

Table 3.5: DHT11 Pinout Description

Pin Name	Description
Vcc	Power supply 3.5 - 5.5 V
Data	Outputs both temperature and humidity readings through serial data
Ground	Connected to the ground

Table 3.6: DHT11 Specifications

Specification	
Operating Voltage	3.5 V to 5.5 V
Operating Current	0.3 mA
Output	Serial Data
Humidity Range	20 % - 90 %
Temperature Range	0 °C - 50 °C
Resolution	temperature and humidity both are 16-bit
Accuracy	±1 °C and ±1 %

3.3.8 MQ-7

The MQ-7 shown in Figure 3.12 is used to measure the carbon monoxide concentrations in the air. The detection range of this sensor is shown in Table 3.8.

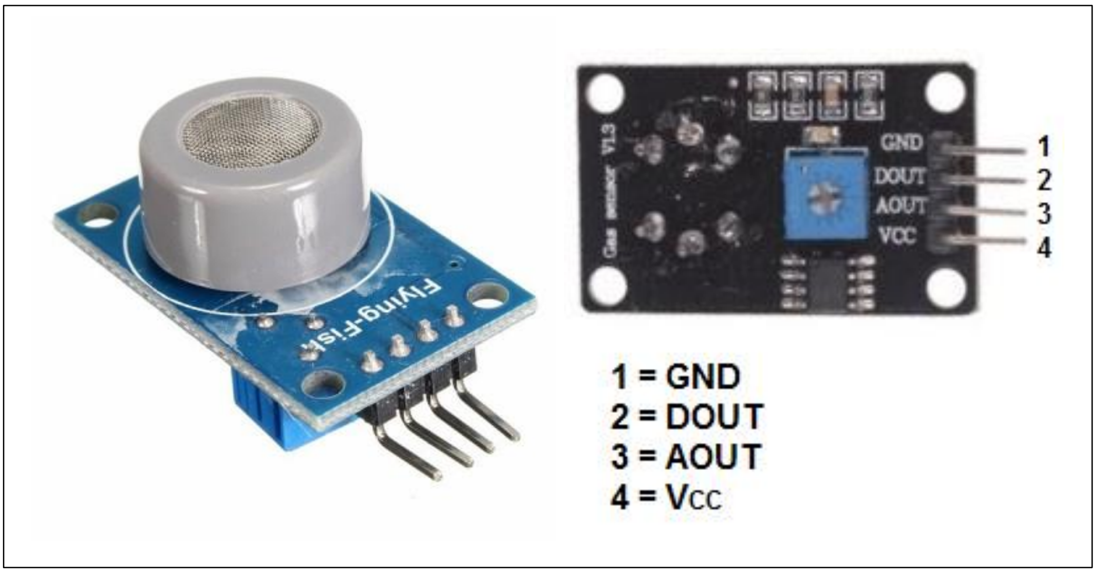


Figure 3.11: MQ-7

Table 3.7: MQ-7 Pinout Description

Pin Name	Description
Vcc	Connect to power supply 5 V
DOUT	Digital output
AOUT	Analog output
GND	To be connected to the ground

Table 3.8: MQ-7 Specifications

Specifications	
Operating Voltage	5 V
Operating Current	150 mA
Detection Gas	Carbon Monoxide
Detection range	20 – 2000 ppm
Temperature	-10 to +50 ˚C

3.3.9 Buzzer

The function of buzzer is to provide alert when fire occurred. The specifications are shown in Table 3.9.

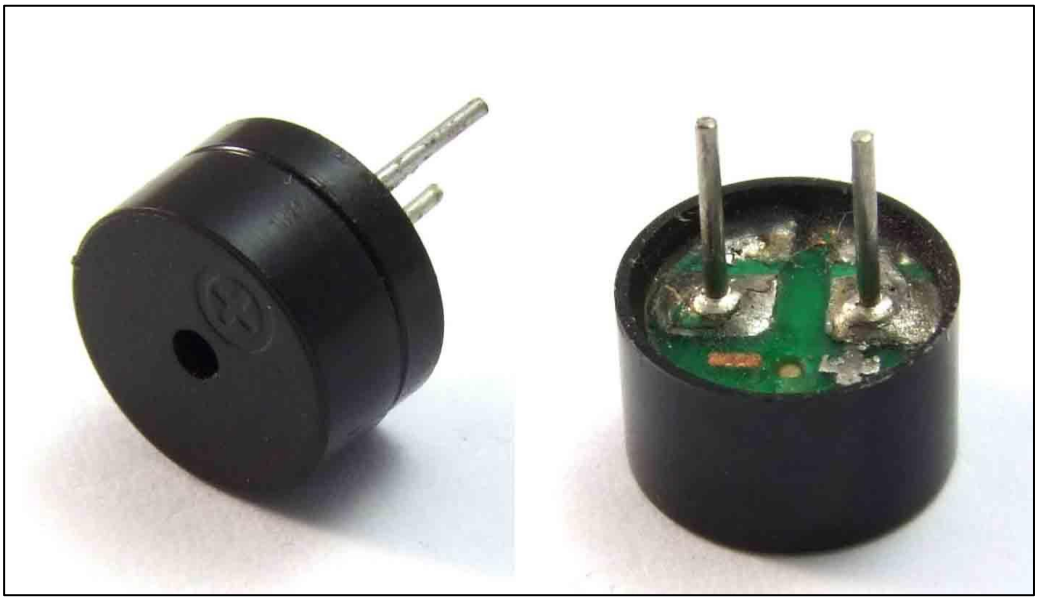


Figure 3.12: Passive Buzzer

Table 3.9: Passive Buzzer Specifications

Specifications	
Rated Voltage	3 – 5 V
Sound Output	≥ 85 dB
Frequency Range	50 ~ 14 000 Hz
Resonant Frequency	2048 Hz
Operating Temperature	−20 to 60 °C

3.4 Software

All the software used to develop the system are discussed in the following subsections.

3.4.1 Node-RED

Node-RED is a visual tool for providing the connections of different devices for the Internet of Things (IoT.). It was an open-source flow-based development tool. Flow-based programming defines an application as a network of black-boxes.

Figure 3.13: Node-RED

On the left side of Figure 3.15 shows the palette which consists of different type of nodes. Each of the nodes has its own purpose and function. With this flow-based development tool, it is easier for the user to create their own application without the need of having much knowledge about programming. Node-RED can be accessed through the Home Assistant (HA) by installing the third-party add-on. Then the Node-RED program flow can be created using the browser-based flow editor. The editor can be accessed through the browser from other machines as long as the Internet Protocol (IP) address is known and the Node-RED is running.

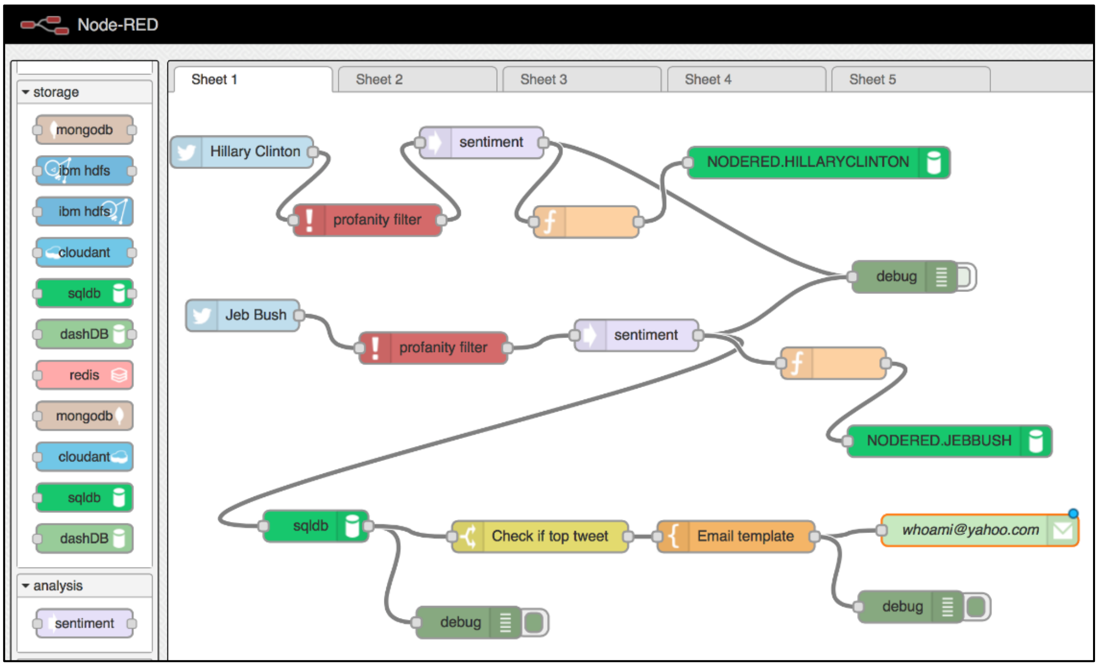


Figure 3.14: Node-RED Browser-Based Flow Editor

Besides, Node-RED's dashboard nodes allow the user to make their own user interface (UI). With this, the data collected can be shown in a more interactive way. The user interface can also be accessed through the browser which enables the user to monitor and control the system remotely. An example of Node-RED user interface is shown in Figure 3.16.

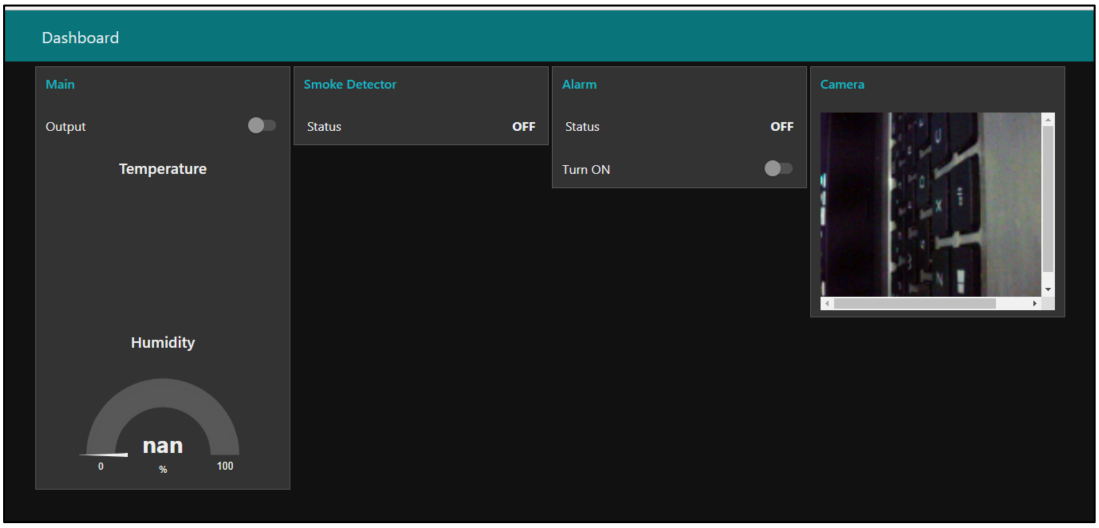


Figure 3.15: Node-RED User Interface

3.4.2 Home Assistant

Home Assistant (HA) is an open-source platform which allows the user to track, control and automate different types of smart devices. In this project, Hass.io is used as the operating system (OS) and install in Raspberry Pi 3 Model B+. The installation of Hass.io turns Raspberry Pi into an automation hub. Hass.io is a complete OS developed mainly for installing and updating HA. The architecture of Hass.io is shown in Figure 3.18.

Figure 3.16: Home Assistant

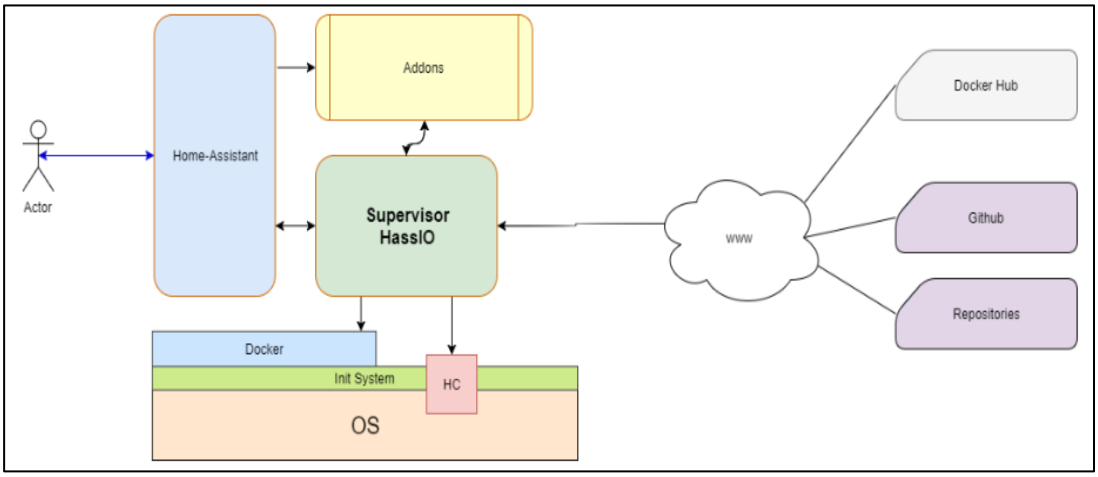


Figure 3.17: Hass.io Architecture (Home Assistant, 2019)

To allow Node-RED to work together with HA, the Node-RED add-on for Home Assistant is installed which is shown in Figure 3.19. The username and password for Node-RED is configured to be accessed through the web browser.

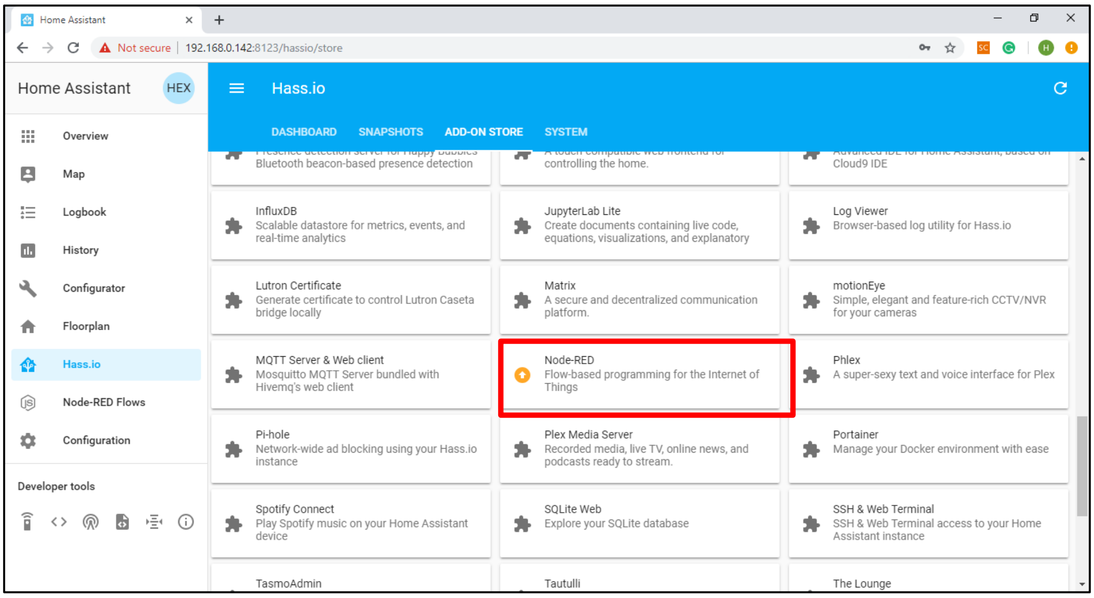


Figure 3.18: Hass.io Add-on Store

3.4.3 Inscape

Inkscape is an open-source vector graphics editor. This software can be used to create or edit the vector graphics such as charts, diagrams, illustration, and complex painting.

Figure 3.19: Inkscape

In this project, Inkscape is used to create the HA floor plan in Scalable Vector Graphics (SVG) format. The existing floor plan which is in AutoCAD Drawing Database (DWG) file format can be converted to Portable Document Format (PDF) and imported into Inscape to be saved as SVG format. For the interaction between the floor plan and Home Assistant, each object is required to be assigned with a specific entity of HA. This can be done by inserting the entity Identification (ID) in the object properties as shown in Figure 3.21.

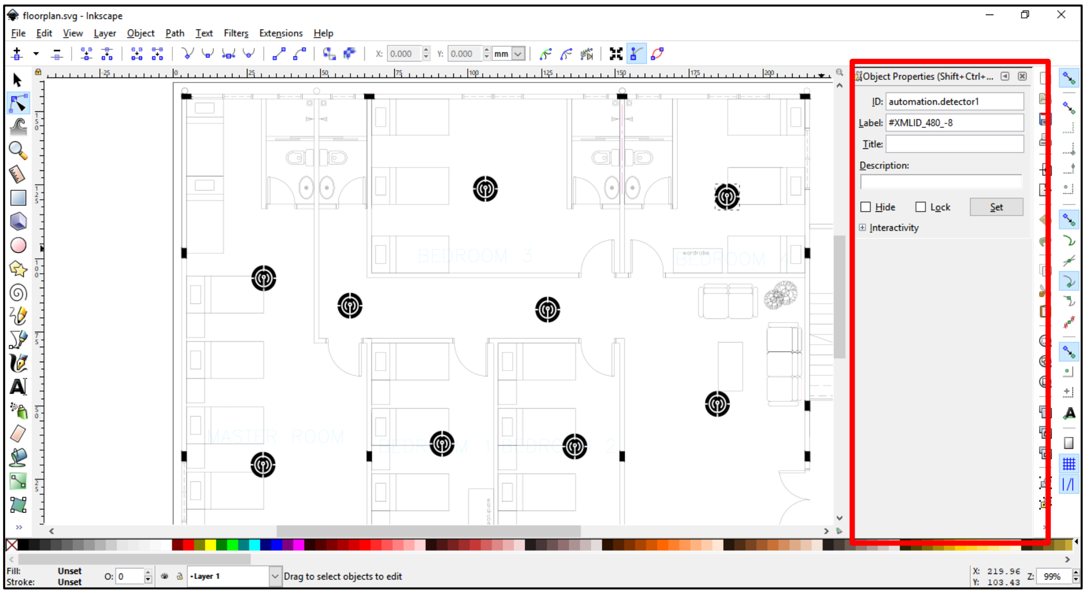


Figure 3.20: Floor Plan Icon Entity ID Setup

3.4.4 Arduino IDE

Arduino Integrated Development Environment (IDE) is a free platform application that is written using a simplified version of C++. It is mainly used to write and upload programs into the Arduino board. This IDE needed for writing and uploading codes to the ESP32.

Figure 3.21: Arduino IDE

3.4.5 IFTTT

If This, Then That (IFTTT) is a free web-based service used to create chains of simple conditional statements which enable various apps and devices to work together. Applets help to bring various services together to perform a task. Figure 3.23 shows an applet that was created by utilizing both “Webhooks” and “VoIP Calls” services.

With the VoIP Calls service, phone call messages will be delivered to the user when any fire or excessive smoke is detected.

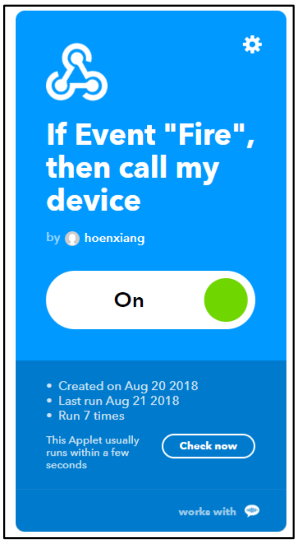


Figure 3.22: Applet Created to Deliver Phone Call Message

3.4.6 Firebase

The readings obtained from the sensors will be stored in the Firebase Realtime Database. It is a cloud-hosted Not Only Structured Query Language (NoSQL) database which allows the user to store and synchronise the data across all the clients in real time. In the case when the users are offline, the Realtime Database Software Development Kit (SDKs) will utilize the local cache on the device for storing. Once it is backed online, the data will be synchronised automatically.

Figure 3.23: Firebase

3.5 Circuit Design

The following subsections show the circuit design of the detector and surveillance units.

3.5.1 Detector

Figure 3.25 shows the schematic diagram of the detector unit. It is an integration of the ionisation smoke detector, MQ-7, ESP32, DHT11, and passive buzzer.

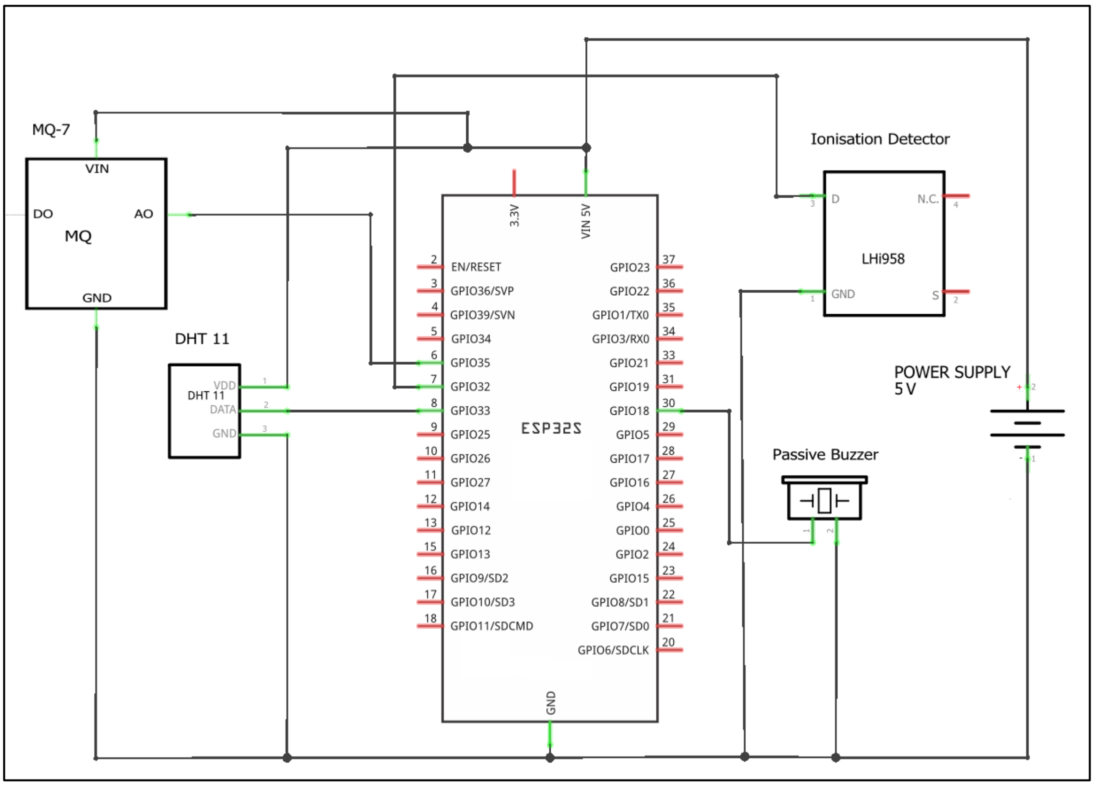


Figure 3.24: Detector Schematic Diagram

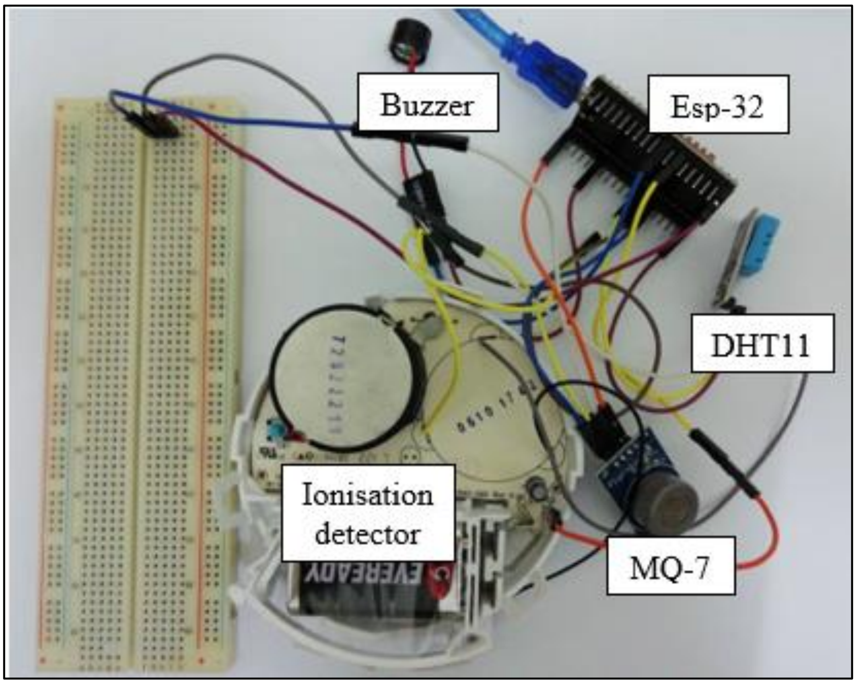


Figure 3.25: Detector Hardware Configuration

Figure 3.26 shows the hardware connection of the detector circuit. One of the pinouts from the ionisation smoke detector is connected to the ESP32. This is to allow the ESP32 to monitor the condition of the ionisation smoke detector.

3.5.2 Surveillance

Referring to Figure 3.27, two servo motors are connected to the Raspberry Pi Zero W. The purpose is to control the direction of the camera.

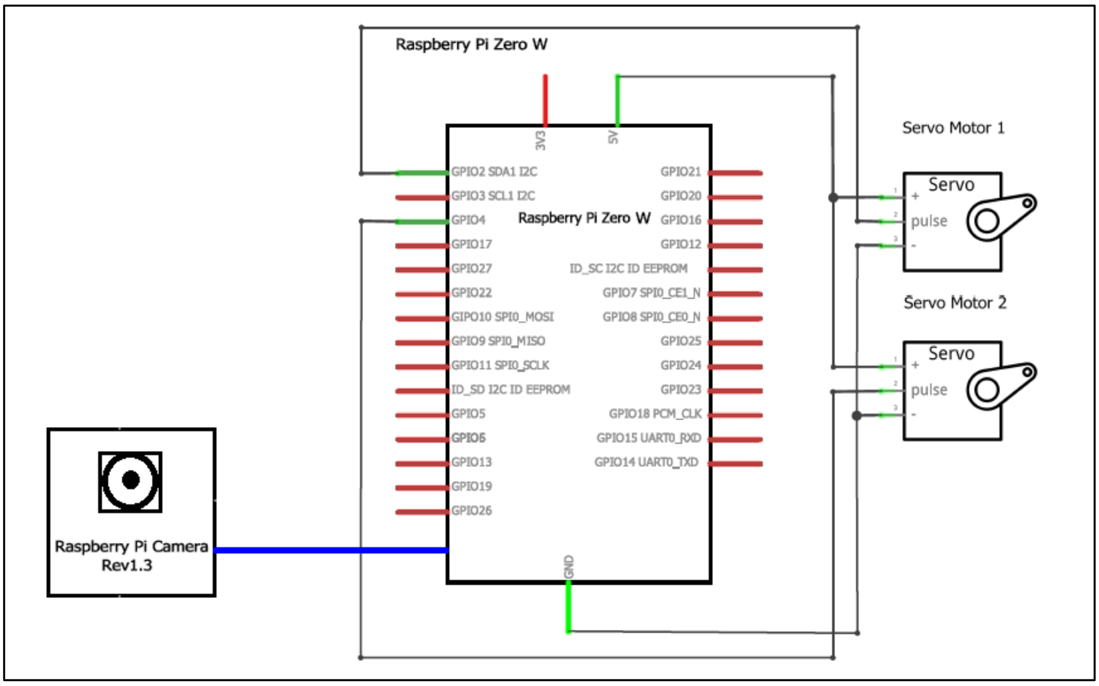


Figure 3.26: Camera Schematic Diagram

3.6 Flow Chart

Section 3.6.1 shows the responses of the system towards different combinations of sensors output. Section 3.6.2 explains the program flowchart of ESP32 (microcontroller of the detector unit).

3.6.1 Overall System Flow Chart

Based on the flowchart in Figure 3.28, when the ionisation smoke detector is triggered or the carbon monoxide (CO) concentration is above or equal to 35 ppm, an email will be sent to notify the user. However, when the detector is triggered and the concentration of CO is above or equal to 35 ppm or 100 ppm, the alarm will be raised automatically. Apart from the email and message notification, an immediate voice call will be sent to the user through the internet. When the temperature is higher than 49 °C, the GPS coordinates of the building will be sent to the nearby fire station.

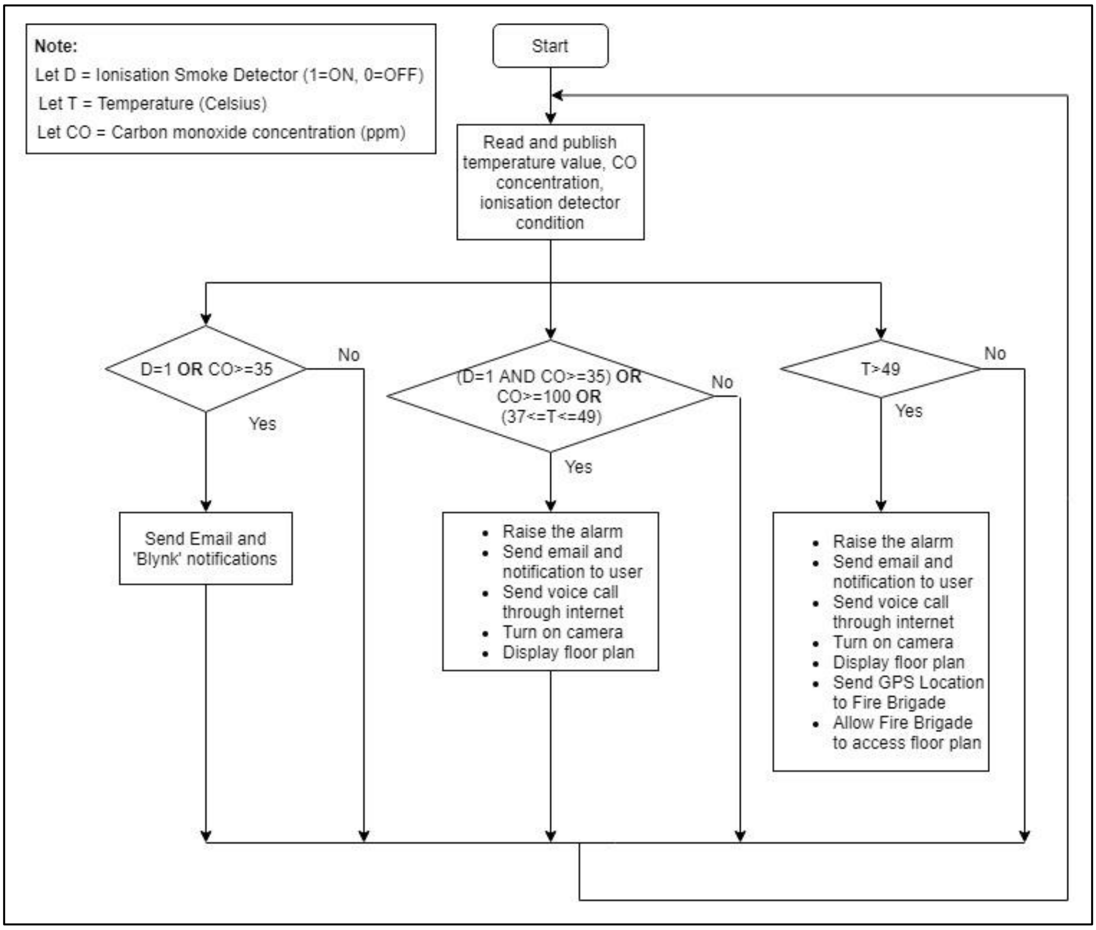


Figure 3.27: Overall Flowchart of the System

Gottuk et al. (2002) found that with only a photoelectric detector, false alarm will trigger more often. Therefore, ionisation smoke detector is used in the proposed system as it has been tested to reduce false alarm. Furthermore, the combination of ionisation detector and CO sensor is able to speed up the fire detection rate.

Figure 3.29 shows the monthly average minimum and maximum temperatures over a year in Kuala Lumpur, Malaysia which is obtained from the website “World

Weather and Climate Information". The maximum temperature is about 33 °C. Based on Appendix G, the sensor used in the system has a temperature accuracy of ±2 °C. Just to be safe, the threshold value of 37 °C is chosen. However, the threshold value for temperature should be adjusted according to the condition of the environment.

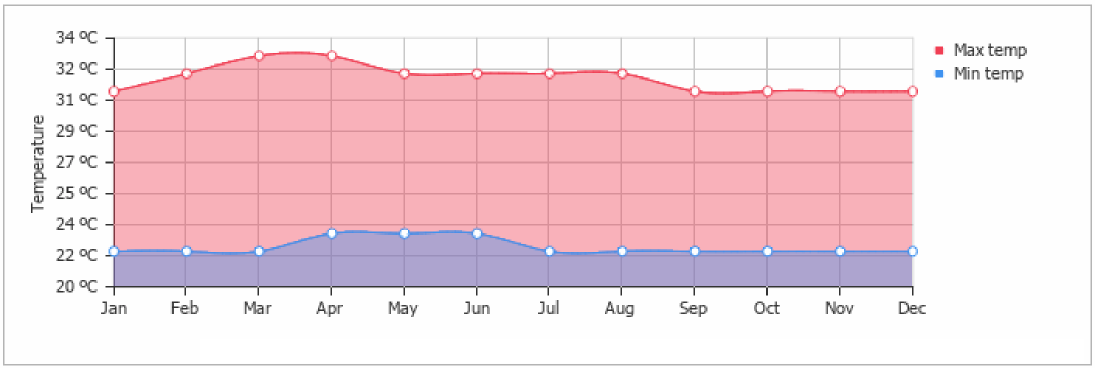


Figure 3.28: Average Min and Max Temperature in Kuala Lumpur, Malaysia (World Weather and Climate Information, 2019)

The threshold value of the concentration of CO are chosen based on the research paper by Goldstein (2008) as shown in Figure 3.30. Low threshold values are chosen so that the victim or user can have more time to response.

Carbon monoxide concentrations, COHb levels, and associated symptoms

Carbon monoxide concentration	COHb level	Signs and symptoms
35 ppm	<10%	Headache and dizziness within 6 to 8 h of constant exposure
100 ppm	>10%	Slight headache in 2 to 3 h
200 ppm	20%	Slight headache within 2 to 3 h; loss of judgment
400 ppm	25%	Frontal headache within 1 to 2 h
800 ppm	30%	Dizziness, nausea, and convulsions within 45 min; insensible within 2 h
1,600 ppm	40%	Headache, tachycardia, dizziness, and nausea within 20 min; death in less than 2 h
3,200 ppm	50%	Headache, dizziness, and nausea in 5 to 10 min; death within 30 min
6,400 ppm	60%	Headache and dizziness in 1 to 2 min; convulsions, respiratory arrest, and death in less than 20 min
12,800 ppm	>70%	Death in less than 3 min

Figure 3.29: Carbon Monoxide Concentration and Associated Symptoms

3.6.2 Detector Flowchart

The program code for the detector unit is written based on the task prioritization. It consists of a main loop and four tasks module.

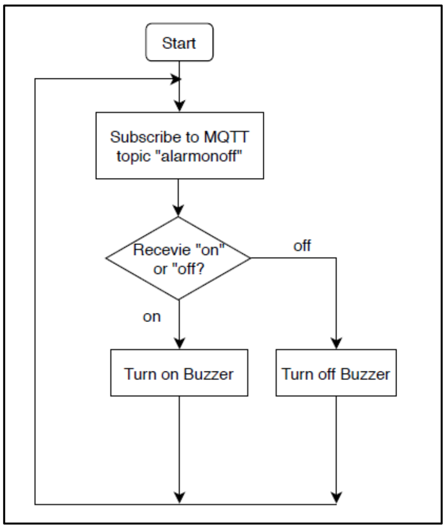


Figure 3.30: Main Loop

In the main loop shown in Figure 3.31, the ESP32 is subscribed to the MQTT topic "*alarmonoff*". It will keep on monitoring the message published to the topic. When the message published is "on" it will turn on the buzzer and vice versa.

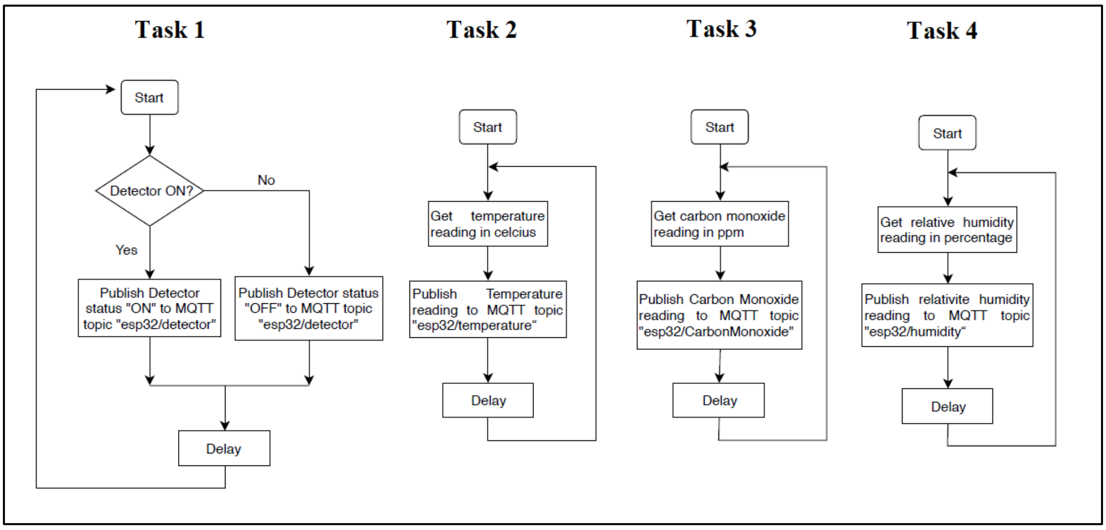


Figure 3.31: Four Tasks Modules

Among all the tasks shown in Figure 3.32, task 1 will be given the highest priority. The purpose of task 1 is to monitor the condition of the ionisation smoke detector. When the ionisation smoke detector is triggered, a string "ON" will be published to the MQTT topic "*esp32/detector*" and "OFF" when it is not triggered. In task 2, temperature reading will be obtained from the DHT11 sensor in the unit Celsius. After that, the value will be published to the MQTT topic "*esp32/temperature*". For task 3, the concentration of carbon monoxide in parts per million (ppm) will be obtained from MQ-7 and published to the topic "*esp32/CarbonMonoxide*". In task 4, the relative humidity of the surrounding will be obtained from the DHT11 sensor and published to the topic "*esp32/humidity*".

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Final Hardware Prototype

The hardware prototype of the system consists of three major parts which are the processing unit, detector and surveillance. Each of them is discussed in the following subsections.

4.1.1 Processing Unit

Figure 4.1 shows Raspberry Pi 3 Model B+ which is used as the processing unit for the system. The Universal Serial Bus (USB) flash drive is used to configure the Wi-Fi connection.

Figure 4.1: Processing Unit

4.1.2 Detector

The hardware prototype of the detector is shown in Figure 4.2. It is an integration of ESP32, ionisation smoke detector, buzzer, MQ-7, DHT11, and buzzer.

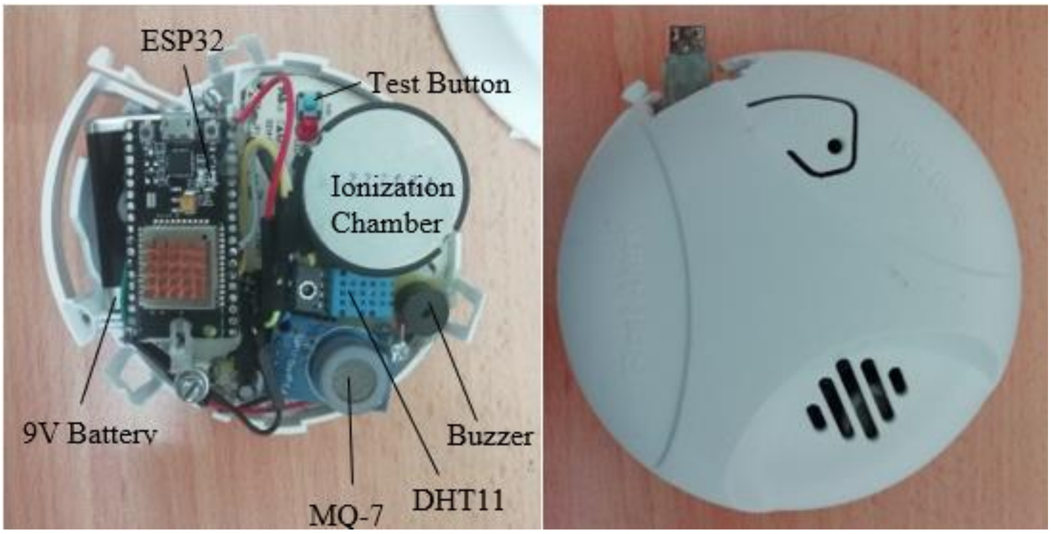


Figure 4.2: Detector Unit

4.1.3 Surveillance

In Figure 4.3, the Raspberry Pi camera v1.3 is attached to the servo motor brackets. This design allows the user to control the camera directions.

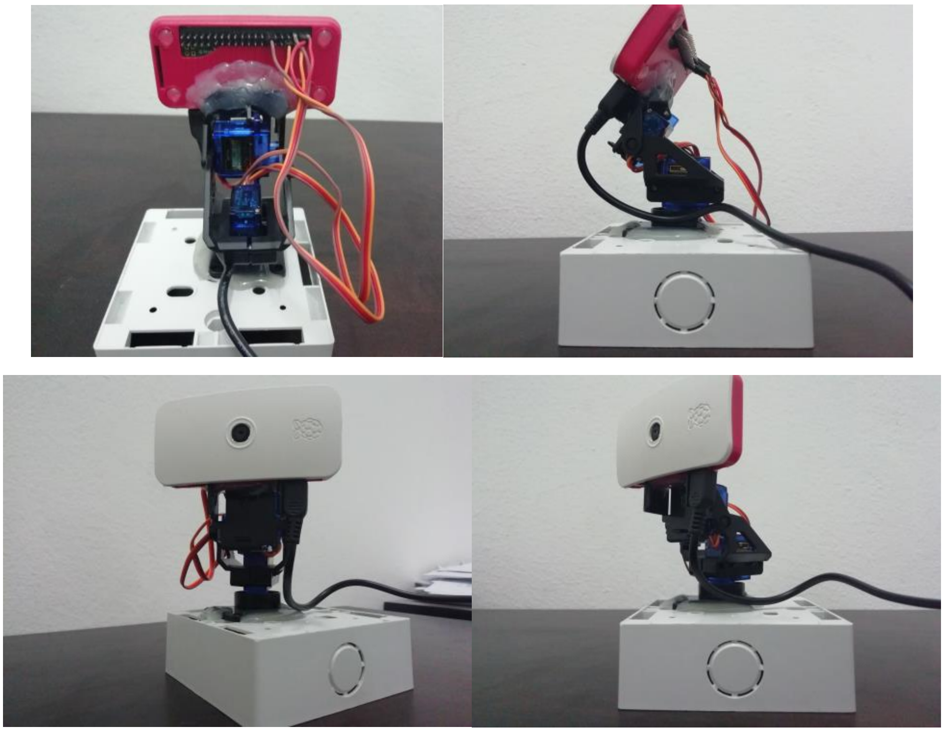


Figure 4.3: Surveillance

4.2 System Features

Various features of the system are discussed in this section.

4.2.1 Node-RED User Interface

Figure 4.4 shows the user interface created using Node-RED. The status of the ionisation smoke detector, alarm and the real time outputs from the sensors are displayed on the dashboard. The user has options to enable or disable the data storing into the Firebase Realtime Database. The values stored in the database can be retrieved and presented in graph form as shown in Figure 4.4. The user has an option to inform the fire brigade manually.

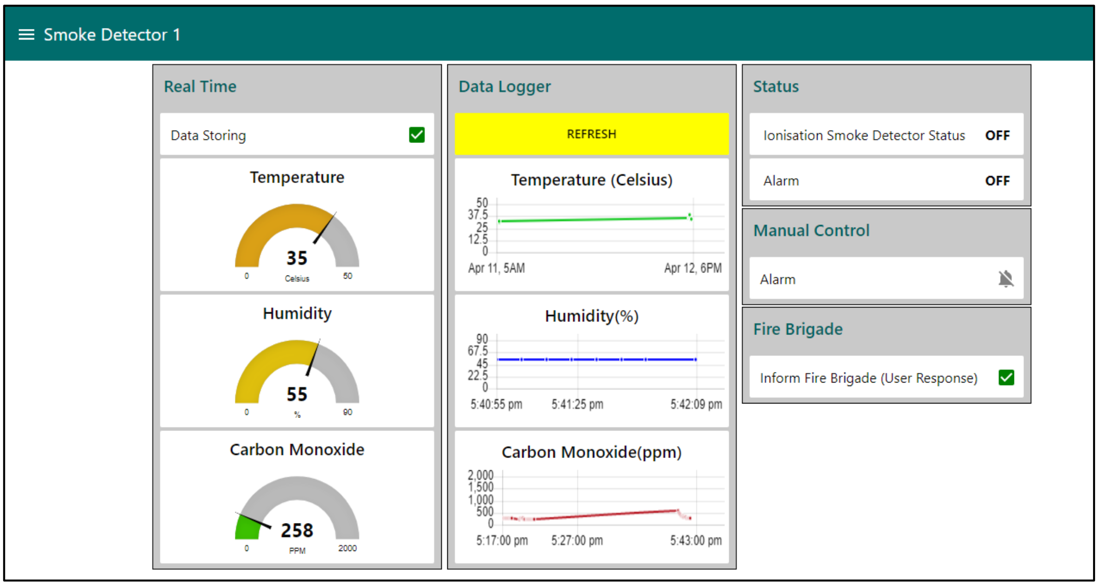


Figure 4.4: Node-RED User Interface

4.2.2 Data storing in Firebase Realtime Database

Figure 4.5 shows the output values of the sensors that are stored in the Firebase Realtime Database. The data is categorized into three types which are the humidity level, carbon monoxide concentration and the temperature level. The data format stored in Firebase is shown in Figure 4.6. The first number "214" represents the sensor output value while the second number "1553…" represents time in JavaScript Object Notation (JSON) format.

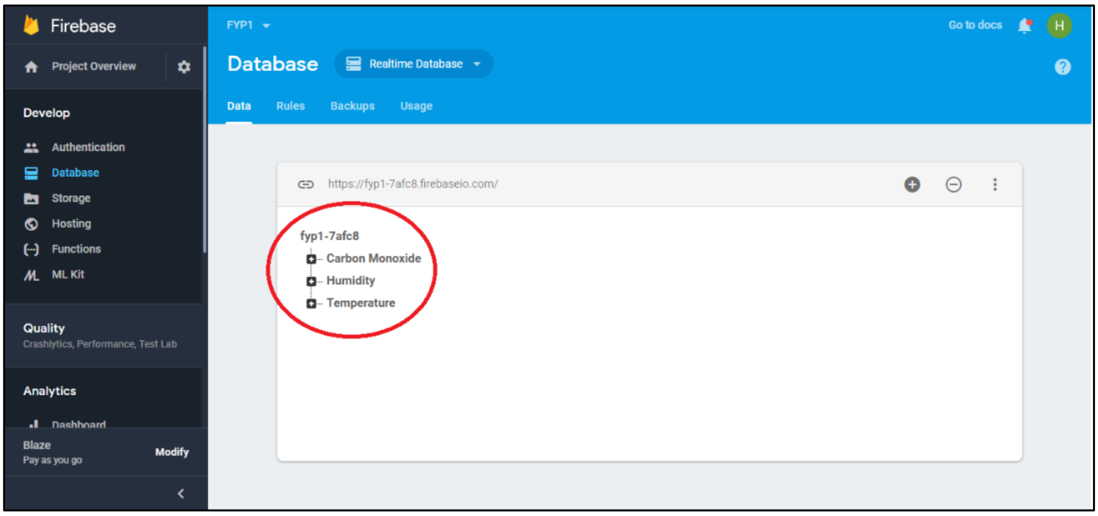


Figure 4.5: Store Data in Firebase

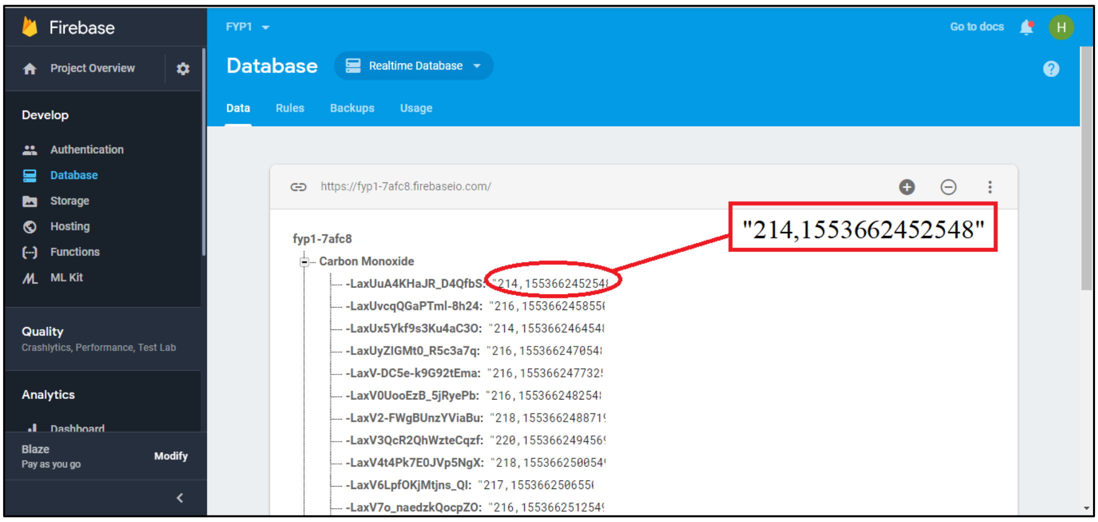


Figure 4.6: Data Format

4.2.3 Data Presentation

Figure 4.7 shows the presentation of the data obtained from the sensors. The left side shows the real time data obtained from the sensors. While on the right side, the graphs are plotted according to the data stored in the cloud.

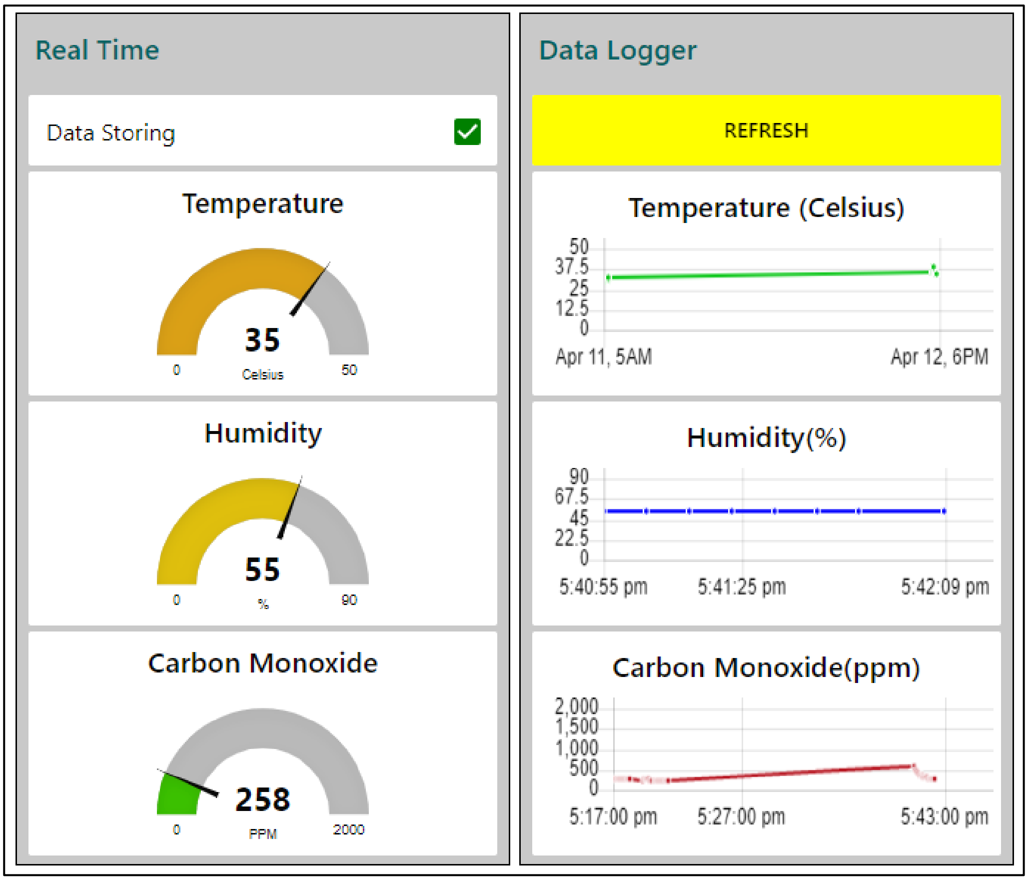


Figure 4.7: Data Presentation

4.2.4 Surveillance

Figure 4.8 shows the user interface for accessing the surveillance camera. The direction of the camera can be controlled by the user. With the function of the live streaming, the user can monitor the scene in real time.

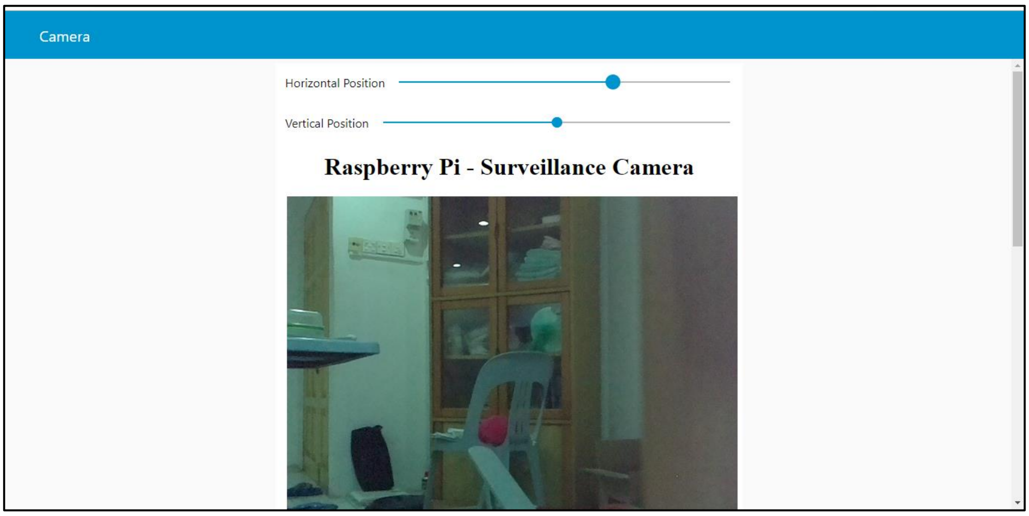


Figure 4.8: Surveillance Camera

4.2.5 Identification of fire location using floor plan

Figure 4.9 shows a sample floor plan created using Inkscape. All the circular icons represent the condition of the ionisation smoke detector. Under normal condition, the icon is in green colour. When the detector is triggered, it changes to red colour. All the detectors should be installed at their respective location according to the floor plan. The floor plan would be very useful in searching for the origin of fire during firefighting operation as well as the evacuation process in high-rise buildings.

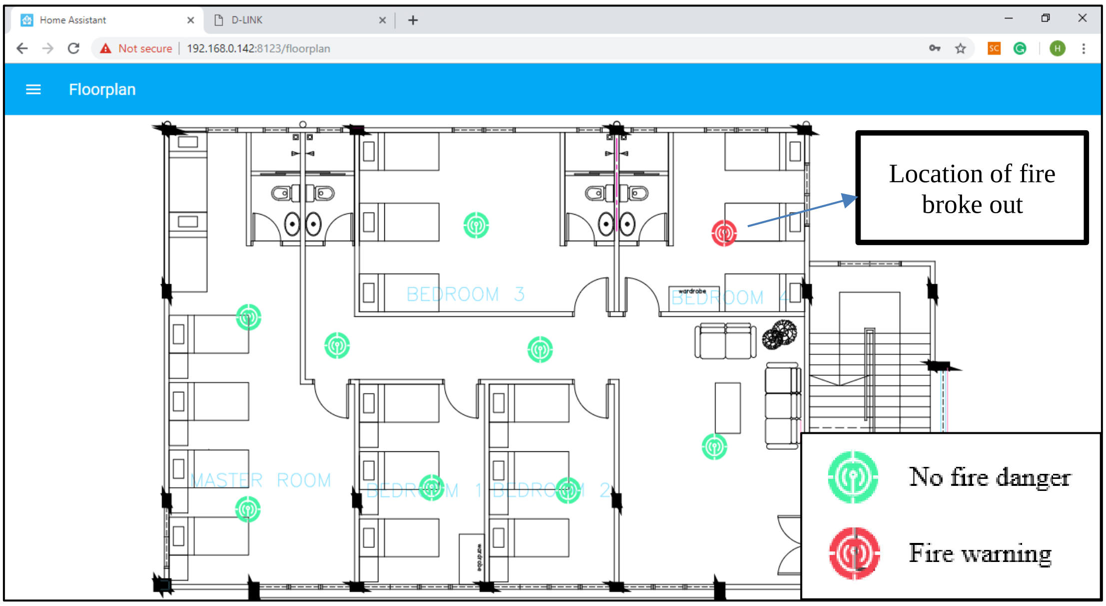


Figure 4.9: Home Assistant Floor Plan (Single Floor)

4.2.6 Blynk App

Figure 4.10 shows the user interface created using Blynk App. This app serves as an alternative for the user. Instead of using the web browser, the user will be able to access the system through the Blynk App installed in the smart phone. The function of this user interface is the same as of the Node-RED dashboard.

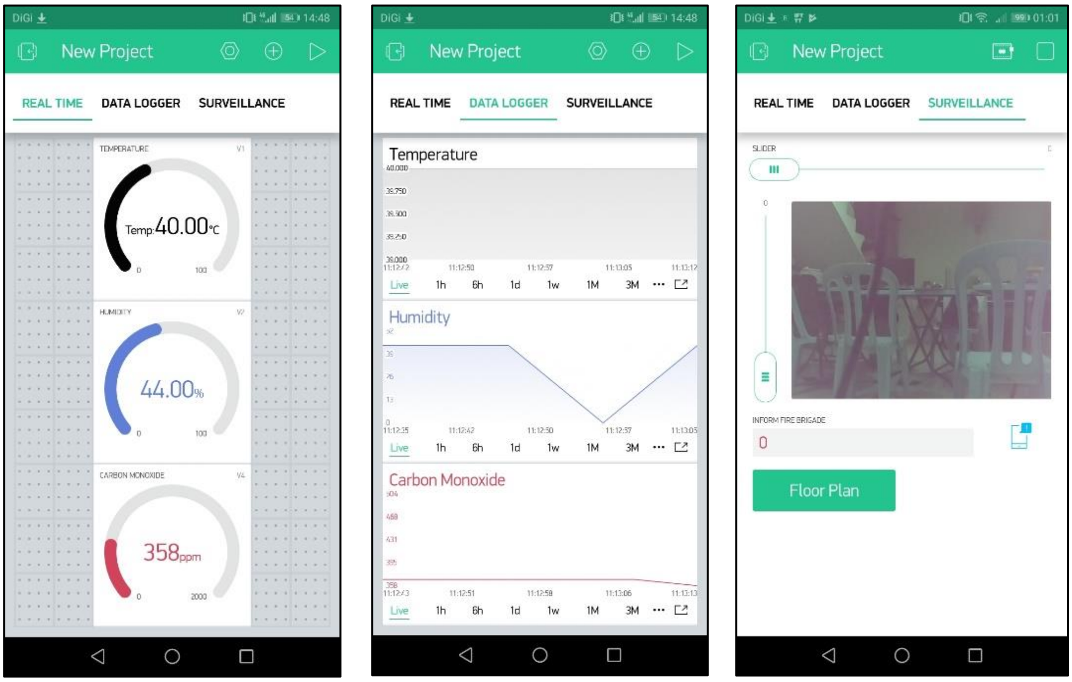


Figure 4.10: Blynk App

4.2.7 Central Fire Monitoring System

Figure 4.11 shows the central fire monitoring system which is developed mainly for the usage of the fire brigades. This system helps to monitor the conditions of the buildings within a particular area. When a fire occurred, an icon will appear on the map and information such as the Global Positioning System (GPS) coordinates of the building will be provided.

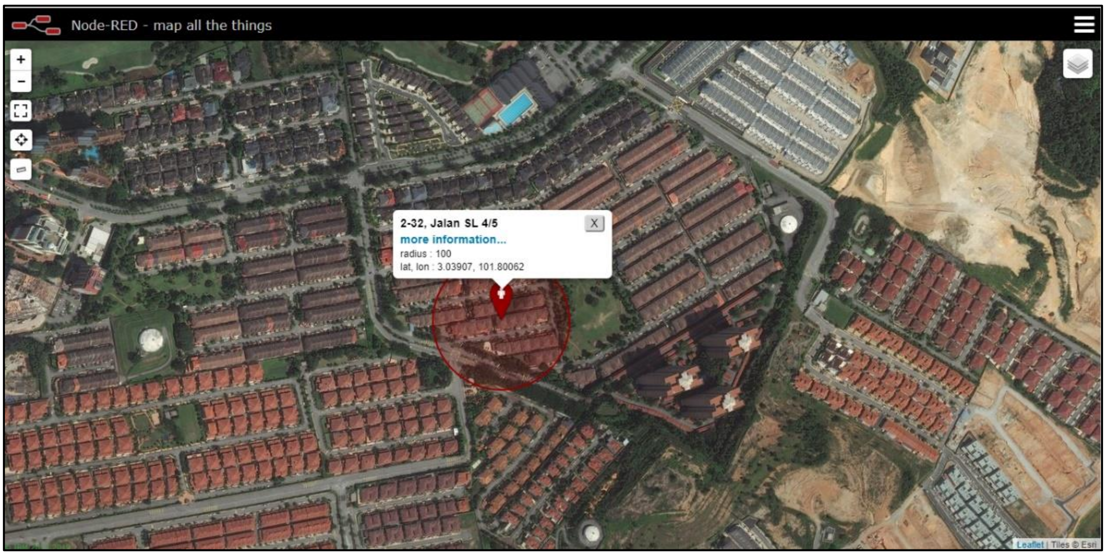


Figure 4.11: Central Fire Monitoring System

4.3 Experiment

The system response has been tested under two different types of fires. According to Appendix F, a preheat time of at least 48 hours is required to calibrate the MQ-7 (carbon monoxide sensor). Since the accuracy of the sensors is not the focus of this project, calibration will not be done prior to the experiment. Therefore, under normal condition, the CO concentration measured is around 210 ppm. Besides, the CO concentration threshold values of 35 ppm and 100 ppm shown in Section 3.6.1 are changed to 400 ppm and 800 ppm respectively for testing purpose.

4.3.1 Smouldering Fire

As shown in Figure 4.12, the burning incense is use to represent a kind of smouldering fire. The alarm went off after being exposed to the smoke for 30 s.

Figure 4.12: Testing with Burning Incense

Table 4.1 shows the sensors output. It can be observed that the temperature would not increase significantly within a short duration of time during a smouldering fire. The real time data of the sensors are shown in Figure 4.13. The ionisation smoke detector is triggered after 30 s of exposure to smoke. It was indicated by the red circular icon shown in Figure 4.14.

Table 4.1: Sensors Output in Smouldering Fire

Time (s)	Temperature (°C)	Humidity (%)	CO (ppm)
0	34	59	256
20	34	59	626
40	34	59	1245
60	34	61	1261

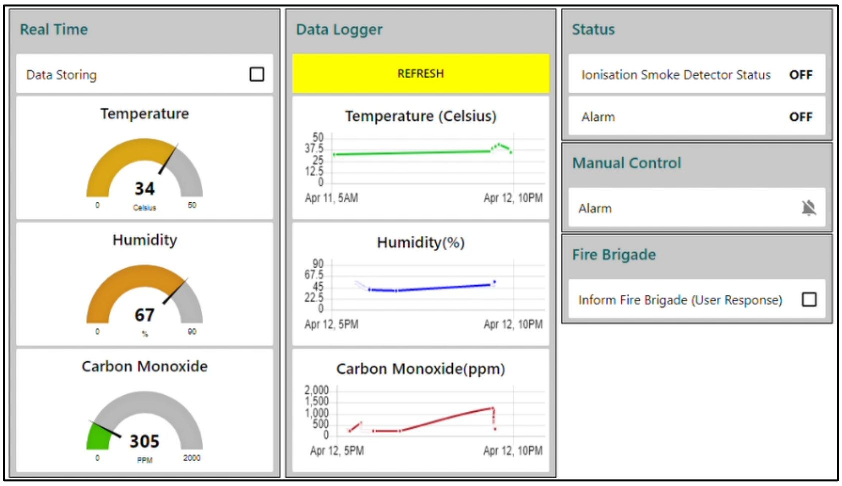


Figure 4.13: Real Time Sensors Output

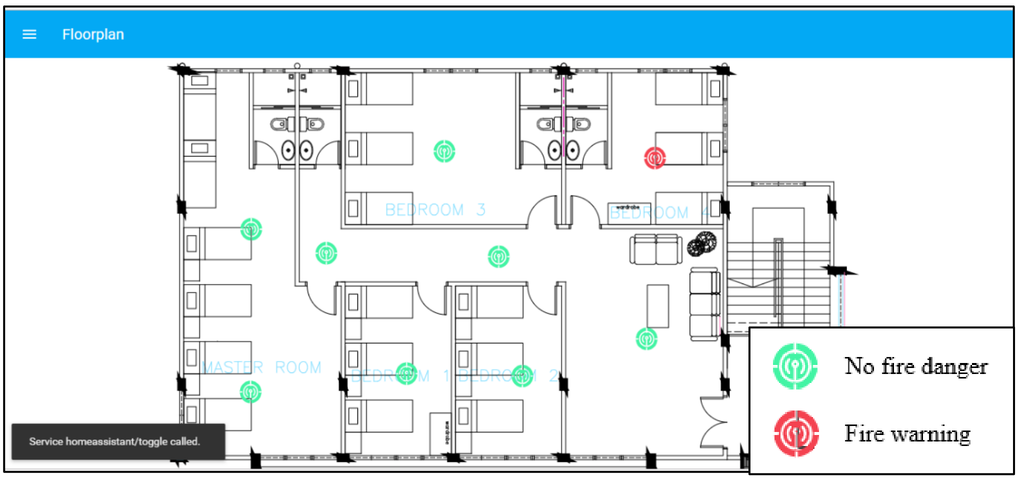


Figure 4.14: HA Floor Plan when Detector is Triggered

In the meantime, alert text message, voice message and email will be sent to the user or related party. These are shown in Figure 4.15, Figure 4.16 and Figure 4.17 respectively.

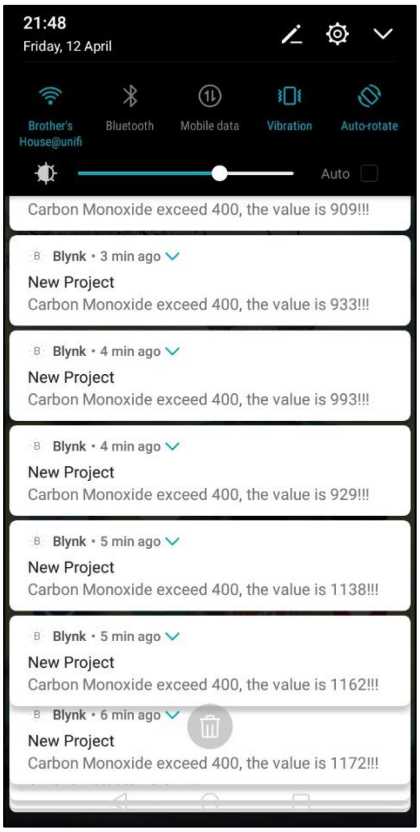


Figure 4.15: Blynk Notifications

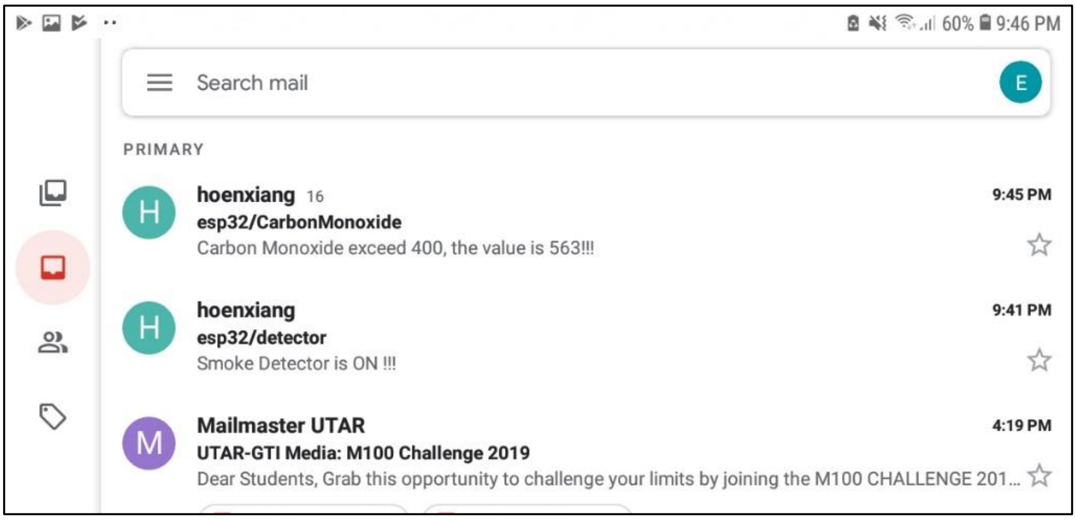


Figure 4.16: Email

Figure 4.17: Voice Message Through Internet

Since the CO concentration exceeded 800 ppm, a Voice Over Internet Protocol (VoIP) call was sent to the user.

4.3.2 Flaming Fire

Figure 4.18 shows the set-up of the experiment. The temperature threshold value required to send the location information to the fire brigade was lowered down from 49 °C to 40 °C for testing purpose. Other than that, the threshold values are the same as in the experiment shown in Section 4.3.1. A glowing candle is used to represent a flaming fire. The alarm went off after 55 s of exposure to the fire.

Figure 4.18: Testing with Glowing Candle

Table 4.2 shows the output value of the sensors every 30 s. The glowing candle produce much less smoke as compared to the burning incense. There is a significant rise in the ambient temperature within the duration of 120 s.

Table 4.2: Sensors Output in Flaming Fire

Time (s)	Temperature (°C)	Humidity (%)	CO (ppm)
0	33	60	212
30	34	64	360
60	35	69	528
90	36	70	693
120	41	70	720

As in the smouldering fire experiment in Section 4.3.1, alert text message, voice message and email will be sent to the user or related party. Besides, the circular icon on the floor plan also changed from green to red when the ionisation smoke detector is triggered. The result is the same as in Figure 4.14.

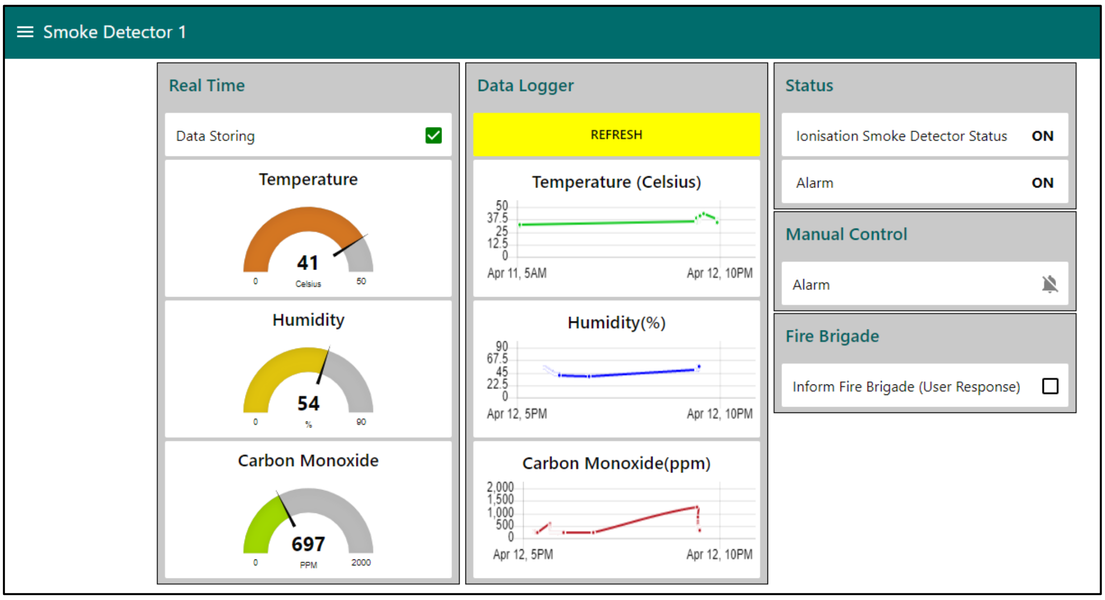


Figure 4.19: Real Time Sensors Output

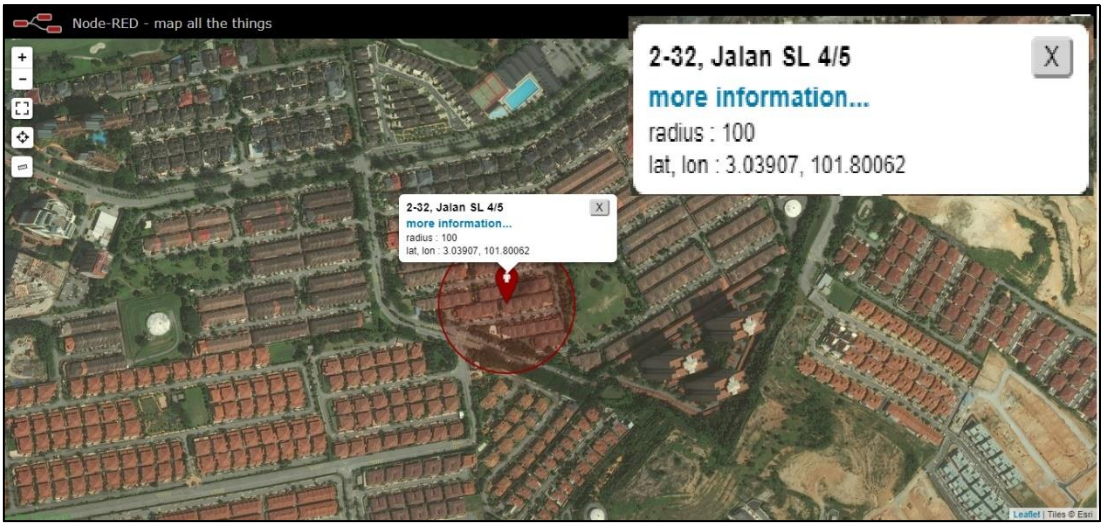


Figure 4.20: Send Location to Fire Station Monitoring System

Since the temperature value exceeded 40 °C after 120 s, the GPS coordinates of the building was sent to the fire brigade and the information was displayed on the map as shown in Figure 4.20.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

The primary purpose of this project is to develop an Internet of Things (IoT) based Smoke Alarm System. The proposed system was able to provide solutions to the problem related to the conventional system. To reduce false alarms, multi-sensors were used instead of one. Besides, user will be able to monitor the condition of the surrounding through the webcam. The user can control the direction of the camera manually with the usage of servo motors. The fire source can be located with the help of the floor plan. This information would be very useful during the firefighting operation as well as the evacuation process. However, with the requirement that the actual installation arrangement of the smoke detector must be the same as what is shown in the floorplan. Experiments were also conducted to prove the functionality of the system. The system can be applied in residential and commercial buildings.

5.2 Recommendations for future work

The communication network of the system should be widened so that when a building is on fire, the people in the nearby buildings will also be informed. The quality of the live streaming can also be improved by using camera that comes with fisheye lens and able to support night vision. Instead of monitoring the fire condition, it can also be used as security camera.

Besides, people who are deaf or having hearing problem may not be able to depend on the alarm to alert them when fire occurred. Therefore, strobe light should be integrated into the detector unit. The light intensity should be strong enough to wake a sleeping person.

REFERENCES

Asif, O., Hossain, M. B., Hasan, M., Rahman, M. T. & Chowdhury, M. E. 2014. Fire-Detectors Review and design of an automated, quick responsive fire-alarm system based on SMS. *International Journal of Communications, Network and System Sciences,* 7**,** 386.

Chakkor, S., Cheikh, E. A., Baghouri, M. & Hajraoui, A. 2014. Comparative performance analysis of wireless communication protocols for intelligent sensors and their applications. *arXiv preprint arXiv:1409.6884*.

Chen, T.-H., Yin, Y.-H., Huang, S.-F. & Ye, Y.-T. The smoke detection for early fire-alarming system base on video processing. *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP'06. International Conference on, 2006*. IEEE, 427-430.

Espressif Systems, 2019. *ESP32 datasheet*. [online] Available at: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf> [Accessed 10 February 2019].

Fire and Rescue Department of Malaysia, 2019. *Annual Report.* [online] Available at: < http://www.bomba.gov.my/index.php/pages/view/746?mid=251> [Accessed 1 February 2019]

Fuzi, M. F. M., Ibrahim, A. F., Ismail, M. H. & Ab Halim, N. S. HOME FADS: A dedicated fire alert detection system using ZigBee wireless network. *Control and System Graduate Research Colloquium (ICSGRC), 2014 IEEE 5th, 2014*. IEEE, 53-58.

Gaikwad, K., Quadri, A., Akshada, S. & Reshma, Z. 2016. Fire Monitoring and Control System. *Proceedings of International Research Journal of Engineering and Technology (IRJET).*

Goldstein, M. 2008. Carbon monoxide poisoning. *Journal of Emergency Nursing,* 34**,** 538-542.

Gottuk, D. T., Peatross, M. J., Roby, R. J. & Beyler, C. L. 2002. Advanced fire detection using multi-signature alarm algorithms. *Fire Safety Journal,* 37**,** 381-394.

Heđi, I., Špeh, I. & Šarabok, A. IoT network protocols comparison for the purpose of IoT constrained networks. *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017 40th International Convention on, 2017*. IEEE, 501-505.

Home Assistant, 2019. *Home Assistant dev docs*. [online] Available at: <https://developers.home-assistant.io/docs/en/architecture_hassio.html> [Accessed 11 February 2019].

Hunkeler, U., Truong, H. L. & Stanford-Clark, A. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on, 2008*. IEEE, 791-798.

Imteaj, A., Rahman, T., Hossain, M. K., Alam, M. S. & Rahat, S. A. An IoT based fire alarming and authentication system for workhouse using Raspberry Pi 3. *Electrical, Computer and Communication Engineering (ECCE), International Conference on, 2017*. IEEE, 899-904.

Islam, T., Rahman, H. A. & Syrus, M. A. Fire detection system with indoor localization using ZigBee based wireless sensor network. *Informatics, Electronics & Vision (ICIEV), 2015 International Conference on, 2015*. IEEE, 1-6.

Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F. & Alonso-Zarate, J. 2015. A survey on application layer protocols for the internet of things. *Transaction on IoT and Cloud Computing,* 3**,** 11-17.

Khan, R., Khan, S. U., Zaheer, R. & Khan, S. Future internet: the internet of things architecture, possible applications and key challenges. *Frontiers of Information Technology (FIT), 2012 10th International Conference on, 2012*. IEEE, 257-260.

Kong, S. G., Jin, D., Li, S. & Kim, H. 2016. Fast fire flame detection in surveillance video using logistic regression and temporal smoothing. *Fire Safety Journal,* 79**,** 37-43.

Kwon, O.-H., Cho, S.-M. & Hwang, S.-M. Design and implementation of fire detection system. Advanced Software Engineering and Its Applications, 2008. ASEA 2008, 2008. IEEE, 233-236.

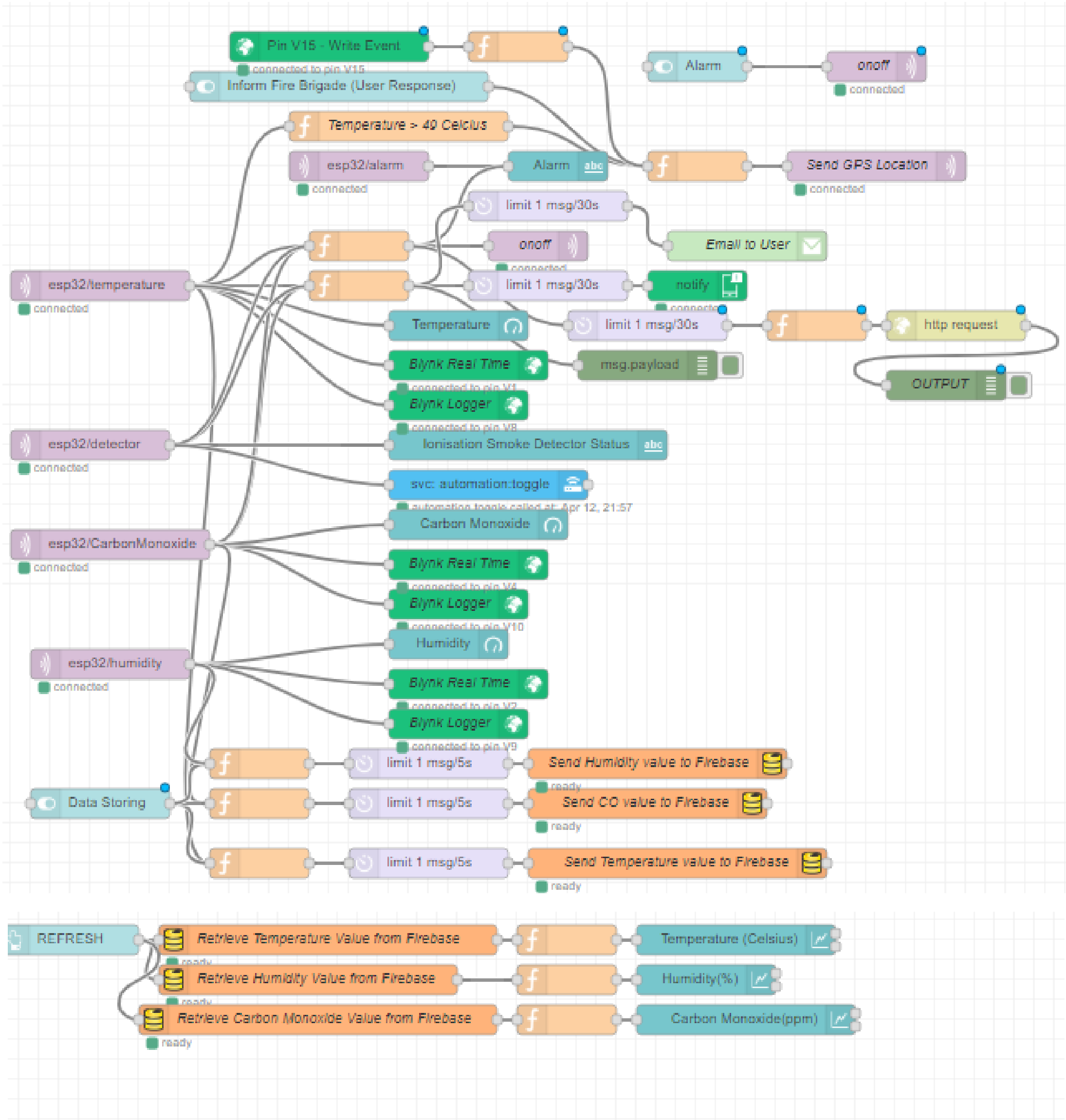
National Fire Protection Association, 2018. *Reporter's Guide: The consequences of fire.* [online] Available at: <https://www.nfpa.org/News-and-Research/Publications-and-media/Press-Room/Reporters-Guide-to-Fire-and-NFPA/Consequences-of-fire#b> [Accessed 12 December 2018].

World Weather and Climate Information, 2019.*Climate and Average Monthly Weather in Kuala Lumpur (Kuala Lumpur Federal Territory), Malaysia*. [online] Available at: <https://weather-and-climate.com/average-monthly-Rainfall-Temperature-Sunshine,Kuala-Lumpur,Malaysia > [Accessed 1 February 2019].

Yassein, M. B. & Shatnawi, M. Q. Application layer protocols for the Internet of Things: A survey. *Engineering & MIS (ICEMIS), International Conference on, 2016*. IEEE, 1-4.

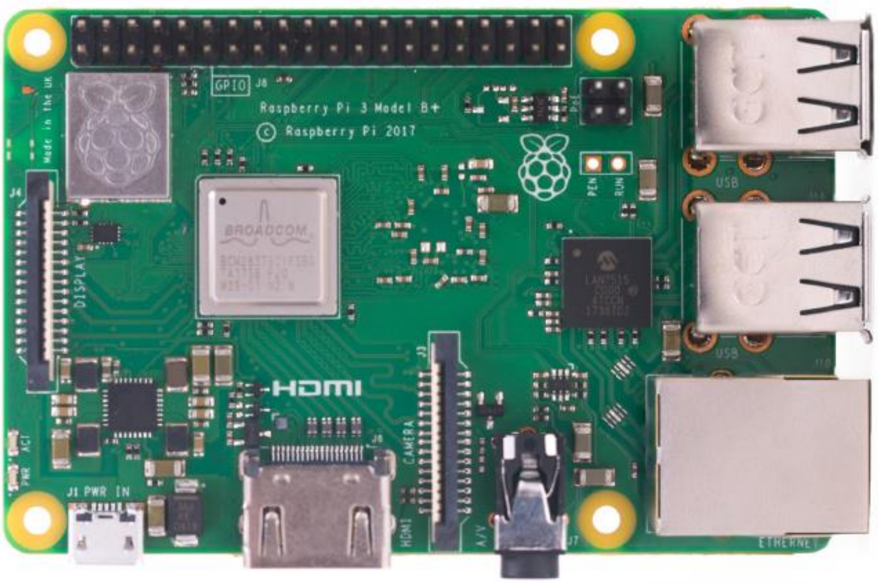
APPENDICES

APPENDIX A: Overall System Node-RED Flow



APPENDIX B: Raspberry Pi 3 Model B+ Datasheet

Overview



The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4 GHz, dual-band 2.4 GHz and 5 GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT

The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market.

The Raspberry Pi 3 Model B+ maintains the same mechanical footprint as both the Raspberry Pi 2 Model B and the Raspberry Pi 3 Model B.

Specifications

Processor: Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz

Memory: 1GB LPDDR2 SDRAM

Connectivity:

- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)
- 4 × USB 2.0 ports

Access: Extended 40-pin GPIO header

Video & sound:

- 1 × full size HDMI
- MIPI DSI display port
- MIPI CSI camera port
- 4 pole stereo output and composite video port

Multimedia: H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics

SD card support: Micro SD format for loading operating system and data storage

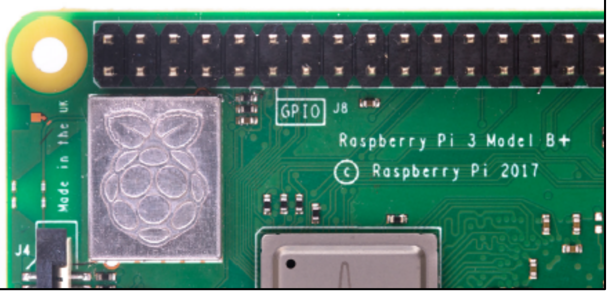
Input power:

- 5V/2.5A DC via micro USB connector
- 5V DC via GPIO header
- Power over Ethernet (PoE)–enabled (requires separate PoE HAT)

Environment: Operating temperature, 0–50°C

Compliance: For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry-pi-3-model-b+

Production lifetime: The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.



APPENDIX C: ESP32 Datasheet

1. Overview

ESP-WROOM-32 is a powerful, generic Wi-Fi+BT+BLE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding.

At the core of this module is the ESP32-D0WDQ6 chip*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the clock frequency is adjustable from 80 MHz to 240 MHz. The user may also power off the CPU and make use of the low-power co-processor to constantly monitor the peripherals for changes or crossing of thresholds. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, Hall sensors, SD card interface, Ethernet, high-speed SPI, UART, I2S and I2C.

Note:
* For details on the part number of the ESP32 series, please refer to the document ESP32 Datasheet.

The integration of Bluetooth, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is future proof: using Wi-Fi allows a large physical range and direct connection to the internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep current of the ESP32 chip is less than 5 μA, making it suitable for battery powered and wearable electronics applications. ESP32 supports a data rate of up to 150 Mbps, and 20.5 dBm output power at the antenna to ensure the widest physical range. As such the chip does offer industry-leading specifications and the best performance for electronic integration, range, power consumption, and connectivity.

The operating system chosen for ESP32 is freeRTOS with LwIP; TLS 1.2 with hardware acceleration is built in as well. Secure (encrypted) over the air (OTA) upgrade is also supported, so that developers can continually upgrade their products even after their release.

Table 1 provides the specifications of ESP-WROOM-32.

Table 1: ESP-WROOM-32 Specifications

Categories	Items	Specifications
Wi-Fi	RF certification	FCC/CE/IC/TELEC/KCC/SRRC/NCC
	Protocols	802.11 b/g/n (802.11n up to 150 Mbps)
		A-MPDU and A-MSDU aggregation and 0.4 μs guard interval support
	Frequency range	2.4 ~ 2.5 GHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZIF receiver with -97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH
	Audio	CVSD and SBC

Categories	Items	Specifications
Hardware	Module interface	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR
		GPIO, capacitive touch sensor, ADC, DAC
	On-chip sensor	Hall sensor, temperature sensor
	On-board clock	40 MHz crystal
	Operating voltage/Power supply	2.7 ~ 3.6V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500 mA
	Operating temperature range	-40°C ~ +85°C
	Ambient temperature range	Normal temperature
	Package size	18±0.2 mm x 25.5±0.2 mm x 3.1±0.15 mm
Software	Wi-Fi mode	Station/SoftAP/SoftAP+Station/P2P
	Wi-Fi Security	WPA/WPA2/WPA2-Enterprise/WPS
	Encryption	AES/RSA/ECC/SHA
	Firmware upgrade	UART Download / OTA (download and write firmware via network or host)
	Software development	Supports Cloud Server Development / SDK for custom firmware development
	Network protocols	IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
	User configuration	AT instruction set, cloud server, Android/iOS app

2. Pin Definitions

2.1 Pin Layout

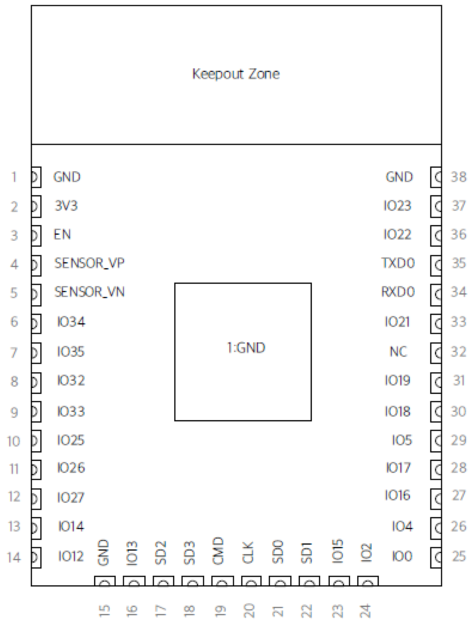


Figure 1: ESP-WROOM-32 Pin layout

2.2 Pin Description

ESP-WROOM-32 has 38 pins. See pin definitions in Table 2.

Table 2: Pin Definitions

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply.
EN	3	I	Chip-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, SENSOR_VP, ADC_H, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, SENSOR_VN, ADC1_CH3, ADC_H, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV

Name	No.	Type	Function
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

Note:
* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

APPENDIX D: MQ-7 Datasheet

TECHNICAL DATA MQ-7 GAS SENSOR

FEATURES

* High sensitivity to carbon monoxide
* Stable and long life

APPLICATION

They are used in gas detecting equipment for carbon monoxide(CO) in family and industry or car.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remark
Vc	circuit voltage	5V±0.1	Ac or Dc
V_H (H)	Heating voltage (high)	5V±0.1	Ac or Dc
V_H (L)	Heating voltage (low)	1.4V±0.1	Ac or Dc
R_L	Load resistance	Can adjust	
R_H	Heating resistance	33 Ω ±5%	Room temperature
T_H (H)	Heating time (high)	60±1 seconds	
T_H (L)	Heating time (low)	90±1 seconds	
P_H	Heating consumption	About 350mW	

b. Environment conditions

Symbol	Parameters	Technical conditions	Remark
Tao	Using temperature	-20℃-50℃	
Tas	Storage temperature	-20℃-50℃	Advice using scope
RH	Relative humidity	Less than 95%RH	
O_2	Oxygen concentration	21%(stand condition) the oxygen concentration can affect the sensitivity characteristic	Minimum value is over 2%

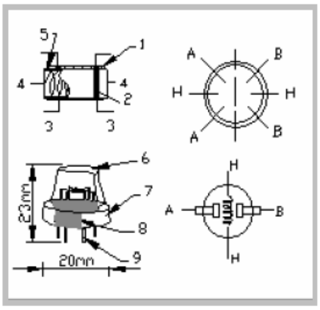
c. Sensitivity characteristic

symbol	Parameters	Technical parameters	Remark
Rs	Surface resistance Of sensitive body	2-20k	In 100ppm Carbon Monoxide
a (300/100ppm)	Concentration slope rate	Less than 0.5	Rs (300ppm)/Rs(100ppm)
Standard working condition	Temperature -20℃±2℃ relative humidity 65%±5% RL:10K Ω ±5%		
	Vc:5V±0.1V VH:5V±0.1V VH:1.4V±0.1V		
Preheat time	No less than 48 hours	Detecting range: 20ppm-2000ppm carbon monoxide	

D. Structure and configuration, basic measuring circuit

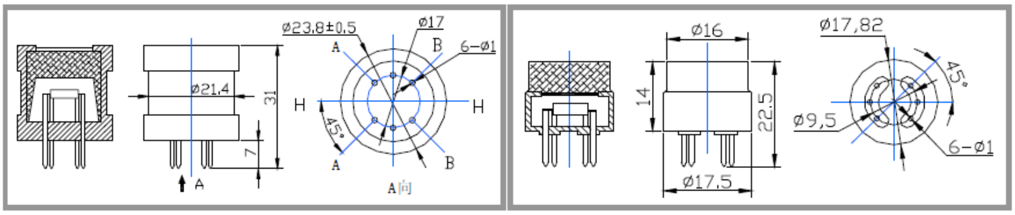
Structure and configuration of MQ-7 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro AL_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-7 have

6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.



	Parts	Materials
1	Gas sensing layer	SnO_2
2	Electrode	Au
3	Electrode line	Pt
4	Heater coil	Ni-Cr alloy
5	Tubular ceramic	Al_2O_3
6	Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7	Clamp ring	Copper plating Ni
8	Resin base	Bakelite
9	Tube Pin	Copper plating Ni

Fig.1



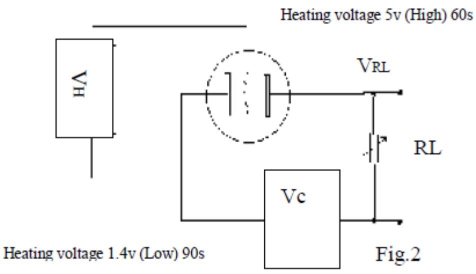


Fig.2

Standard circuit:

As shown in Fig 2, standard measuring circuit of MQ-7 sensitive components consists of 2 parts. one is heating circuit having time control function (the high voltage and the low voltage work circularly). The second is the signal output circuit, it can accurately respond changes of surface resistance of the sensor.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

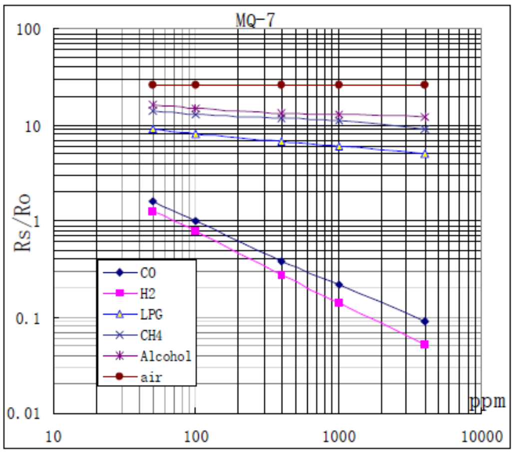


Fig.3 sensitivity characteristics of the MQ-7

Fig.3 is shows the typical sensitivity characteristics of the MQ-7 for several gases.

in their: Temp: 20℃、

Humidity: 65%、

O_2 concentration 21%

RL=10kΩ

Ro: sensor resistance at 100ppm CO in the clean air.

Rs: sensor resistance at various concentrations of gases.

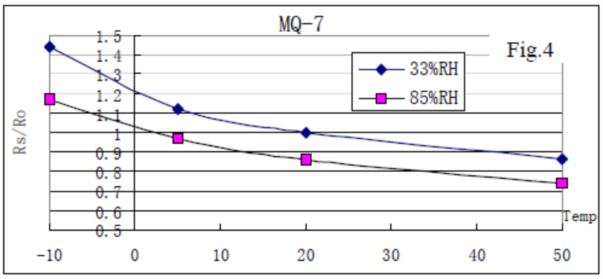


Fig.4

Fig.4 is shows the typical dependence of the MQ-7 on temperature and humidity.

Ro: sensor resistance at 100ppm CO in air at 33%RH and 20degree.

Rs: sensor resistance at 100ppm CO at different temperatures and humidities.

OPERATION PRINCIPLE

. The surface resistance of the sensor Rs is obtained through effected voltage signal output of the load resistance RL which series-wound. The relationship between them is described:

$$Rs\backslash RL = (Vc-VRL) / VRL$$

Fig. 5 shows alterable situation of RL signal output measured by using Fig. 2 circuit output

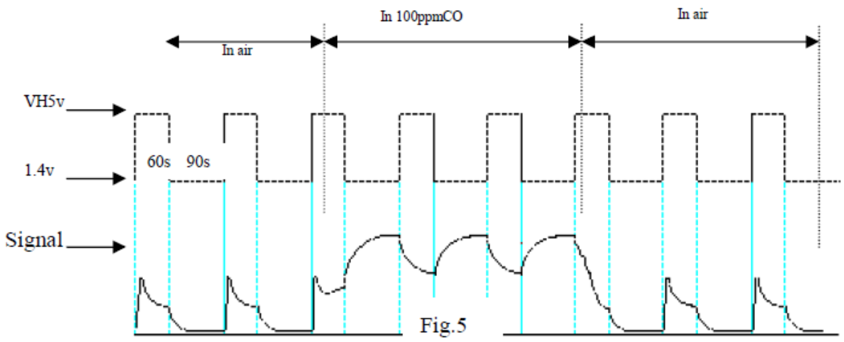


Fig.5

signal when the sensor is shifted from clean air to carbon monoxide (CO) , output signal measurement is made within one or two complete heating period (2.5 minute from high voltage to low voltage).

Sensitive layer of MQ-7 gas sensitive components is made of SnO_2 with stability, So, it has excellent long term stability. Its service life can reach 5 years under using condition.

SENSITVITY ADJUSTMENT

Resistance value of MQ-7 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 200ppm CO in air and use value of Load resistance that(R_L) about 10 KΩ (5KΩ to 47 KΩ).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence. The sensitivity adjusting program:

a. Connect the sensor to the application circuit.

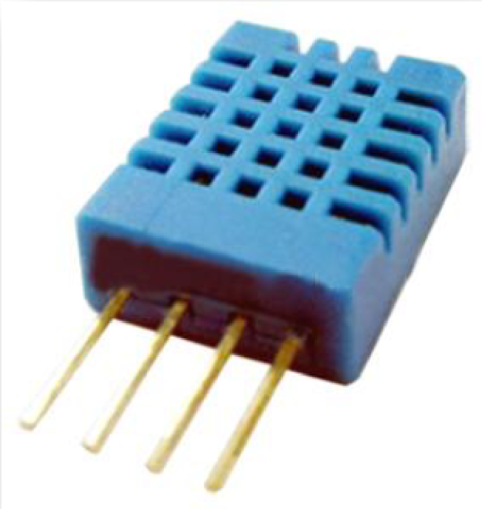
b. Turn on the power, keep preheating through electricity over 48 hours.

c. Adjust the load resistance RL until you get a signal value which is respond to a certain carbon monoxide concentration at the end point of 90 seconds.

d. Adjust the another load resistance RL until you get a signal value which is respond to a CO concentration at the end point of 60 seconds .

Supplying special IC solutions, More detailed technical information, please contact us.

APPENDIX E: DHT11 Datasheet



Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request.

2. Technical Specifications:

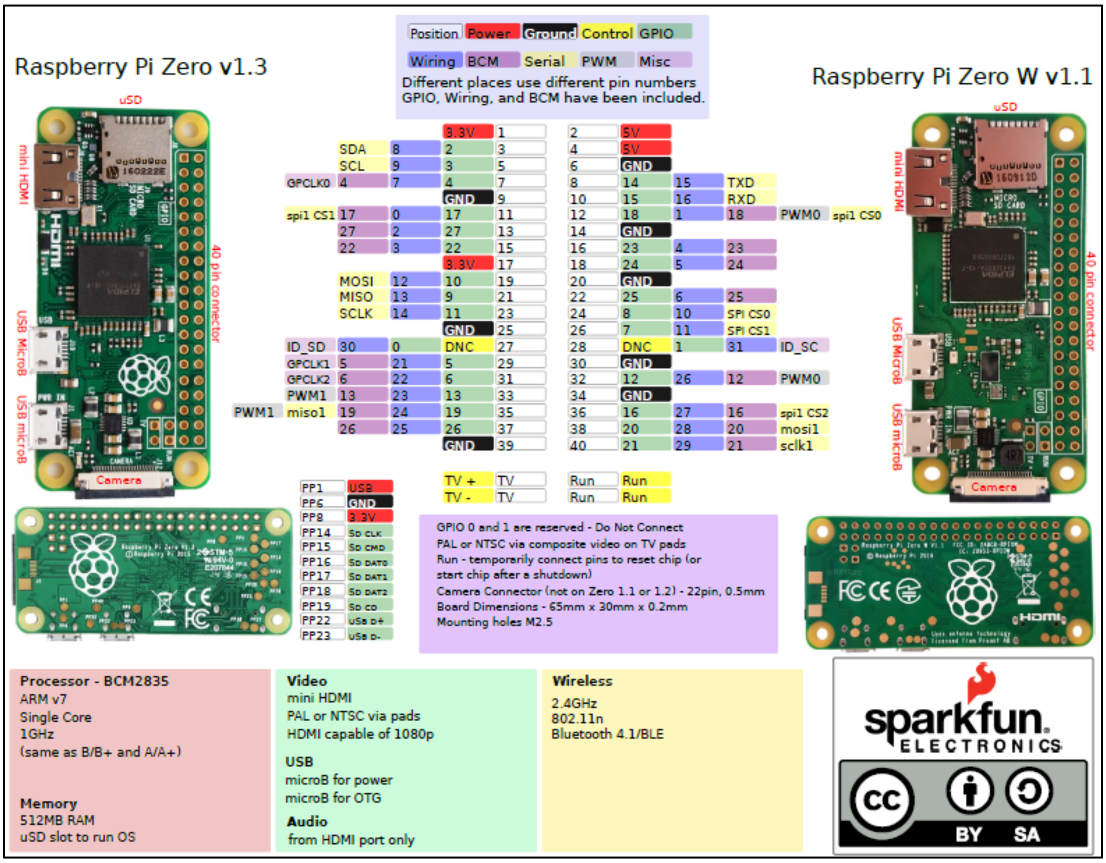
Overview:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5%RH	±2℃	1	4 Pin Single Row

Detailed Specifications:

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃，1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

APPENDIX F: Raspberry Pi Zero W Datasheet



APPENDIX G: Raspberry PI Camera v1.3

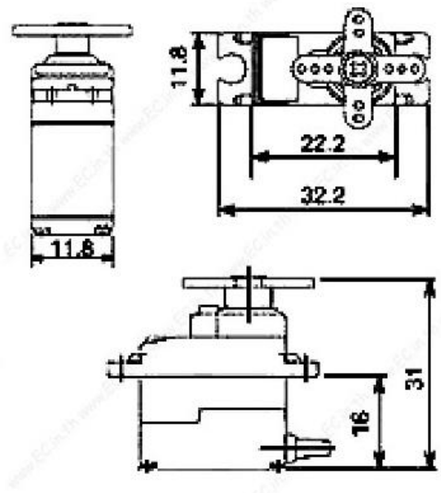
Raspberry Pi

CAMERA MODULE

Product Name	**Raspberry Pi Camera Module**
Product Description	High definition camera module compatible with the Raspberry Pi model A and model B. Provides high sensitivity, low crosstalk and low noise image capture in an ultra small and lightweight design. The camera module connects to the Raspberry Pi board via the CSI connector designed specifically for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the BCM2835 processor.
RS Part Number	**775-7731**
Specifications	
Image Sensor	Omnivision 5647 CMOS image sensor in a fixed-focus module with integral IR filter
Resolution	5-megapixel
Still picture resolution	2592 x 1944
Max image transfer rate	1080p: 30fps (encode and decode) 720p: 60fps
Connection to Raspberry Pi	15 Pin ribbon cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI-2)
Image control functions	Automatic exposure control Automatic white balance Automatic band filter Automatic 50/60 Hz luminance detection Automatic black level calibration
Temp range	Operating: -30° to 70° Stable image: 0° to 50°
Lens size	1/4"
Dimensions	20 x 25 x 10mm
Weight	3g

APPENDIX H: Servo Motor SG90 Datasheet

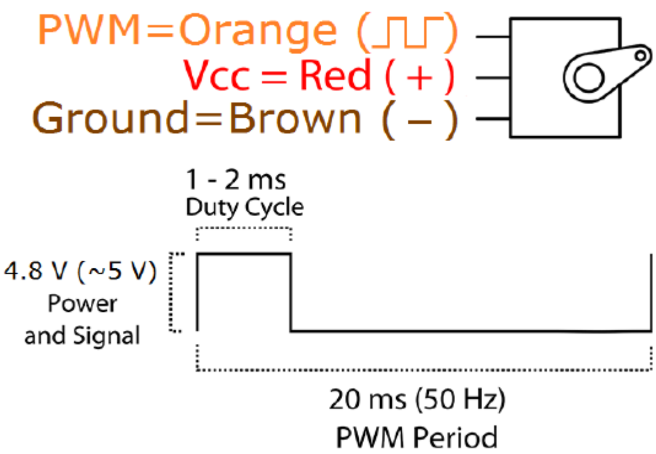
SG90 9 g Micro Servo



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

Specifications

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 µs
- Temperature range: 0 ºC – 55 ºC



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.