PCB Design for IoT Based Fire Detection and Alarm System

Norsuzila Ya'acob [1], Yasser Asrul Ahmad [2], Fadhila Nasir Al Mashhoor [3]

*[1] University Teknologi MARA, Shah Alam, College of Engineering,
School of Electrical Engineering, Malaysia. ORCID ID:0000-0002-3305-8677*
*[2] International Islamic University Malaysia, Kulliyyah of Engineering,
Department of Electrical & Computer Engineering, Malaysia.
ORCID ID:0000-0002-7700-5577*
*[3] International Islamic University Malaysia, Kulliyyah of Engineering,
Department of Electrical & Computer Engineering, Malaysia.
Email: [1] norsuzila@uitm.edu.my, [2] yasser@iium.edu.my*

Abstract

A server room is a hub for operations and IT infrastructure. As a lot of physical computers are stored, operated, and accessed from a server room, it is important to take safety measures to protect the data in this room from fire hazard. A hard-wired fire sensor is commonly placed in the server room. The system is large and when placed at proximity, the system can be damaged by the fire easily. This paper proposes an IoT-based fire alarm system that is smaller in size and operates independently from the server. This system is designed to consist of temperature and gas sensors that will take readings from its surroundings and alert the user through an IoT gateway. The Printed Circuit Board (PCB) is designed and built using KiCad software to enable integration of gas and temperature sensors together with the IoT ESP8266 on a small footprint. The system was initially tested before the PCB design and retested again for validation when integrated on the PCB. Threshold values are set for the sensors, to alert the user remotely incase the threshold values are exceeded. The system demonstrates functionality when the real-time data is retrieved by the web application Blynk. The user can be alerted and remotely monitor and use this system.

Keywords: Fire, Printed Circuit Board (PCB), sensors, server room, Blynk, IoT, temperature

I. INTRODUCTION

Fire is one of the hazardous disasters that undergoes the process of combustion, changing the composition of oxygen to carbon dioxide. It has been found in a survey that 80% of losses caused due to fire would have been avoided if the fire was detected immediately (Ghasemzadeh, Tilebon, Nasirinezhad, & Basile, 2018). It produces heat, light, and other toxic gases during this reaction process. When a fire occurs, oxygen in air reacts to produce oxides, and some nitrogen will also be converted to nitrogen oxide at high temperatures. The level of toxic gases produced in a fire is high, consisting of carbon dioxide, that can easily yield carbon monoxide (Ghasemzadeh, Tilebon, Nasirinezhad, & Basile, 2018). It can be started for multiple reasons and being highly volatile, it spreads rapidly, becoming dangerous to lives and surroundings promptly. Technology plays a huge role in the safety and security of every individual, so a fire alarm is applied to detect fire and secure places like workplaces, households, warehouses, and server rooms. A fire alarm system is a system applied in the field of security and safety, as it detects if there is an indication of a fire situation. The general components of a hardwired fire alarm circuit comprise sensors, a controlling device, and a buzzer. These components are chosen to notify the occupants immediately and commence immediate control of the fire. The system is activated automatically via smoke detectors or manual alarm activation points or pulls stations are installed, to activate the buzzer or water sprinklers in the building.

According to Artim (2018), a fire alarm system is common among every household, workplace, warehouse, and server room these days. There is a problem of the fire alarm system failing to operate when the fire is at a close range. In case of a fire, the fire detection system is also prone to the hazard which terminates the system completely. The problem with the common fire detection system is that it is hardwired, and they tear or need replacement over time. Asif, Hossain, Hasan, Rahman, & Chowdhury (2014) states that this requires maintenance of cables, which can be quite expensive for a simple fire alarm system. Fire poses a significant risk in the workplace. In a server room, there are a lot of physical computers and hence, so many cables as well. Therefore, by implementing a fire alarm system on a PCB, the fire situation in the room can easily be maintained. Also, the unavailability of real-time data to the user, makes monitoring the situation remotely difficult for fire prevention.

To overcome all of the problems stated above, the proposed system is chosen, by identifying sensors that will optimize the performance of the system, by obtaining values at a longer distance. The system will be implemented on a PCB that will provide minimal size of the system. Also, to enable the user to connect remotely to the system, the system is equipped with an IoT component. This system will locate a fire instantly remotely, and the minimal size will make the system less prone to the hazard itself.

In this paper, the main idea is to propose an IoT-based fire alarm system implemented on a PCB and improvise this system by comparing it to existing fire detection systems.

To identify and determine the components of the fire alarm on a PCB for a server room.

To design a low-cost and low-maintenance system, by implementing on a Printed Circuit Board (PCB).

To integrate and test the fire detection system on the PCB (Printed Circuit Board) with IoT platform.

II. RELATED WORK

A study by Asif, Hossain, Hasan, Rahman, & Chowdhury (2014), proposed a fire detector system, with an ATMega8L as the control unit, interfaced with an LCD to show the status of the control system. The system comprises MQ2, a temperature sensor, and an LED that will glow when the output from the sensors is high. GSM is required for quick responses to be alerted. The GSM uses Sim300CZ in this system. And the final product is implemented on a PCB using EAGLE software to design the PCB. A snapshot of the PCB is shown below in Figure 1.

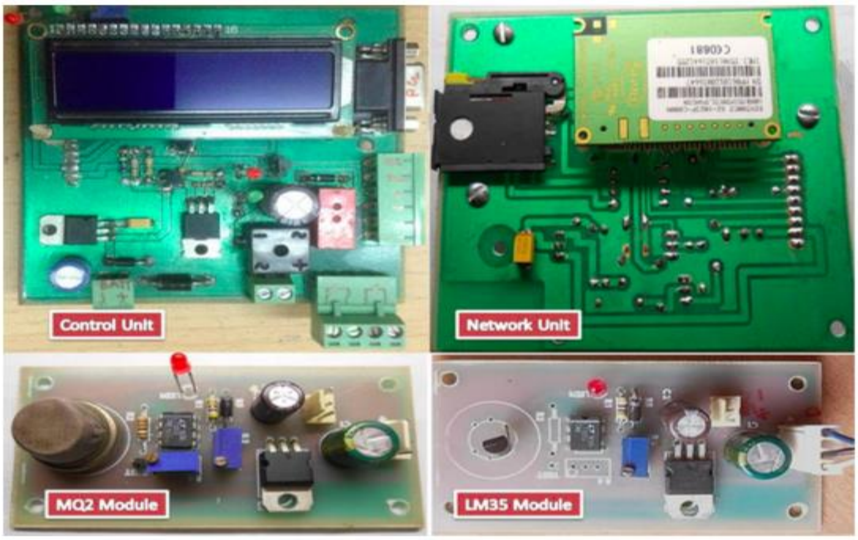

***Figure 1** PCB hardware end-product (Asif, Hossain, Hasan, Rahman, & Chowdhury, 2014)*

Table 1 below shows a summary of comparison between our proposed system and the benchmarked system by Asif, Hossain, Hasan, Rahman, & Chowdhury (2014).

Table 1 . *Comparison with Benchmarking*

Parameters	Proposed System	Benchmark	Limitation/Gap of Benchmark System
Temperature sensor	DHT22	LM35	Has a smaller range
IoT component	ESP-01	GSM	GSM limited to SMS and requires a sim card
PCB model	Single layered, one unit	Double layered PCB, 2 units	Two units are required to monitor and control. PCB is more complex

Microprocessor

A processor is used to understand and comprehend the data obtained from the sensors. This processor incorporates the function of a central processing unit to read and write commands to be followed. The execution of a command depends on the type of data that is sent, from the sensor. The instruction to be followed is programmed into the processor and the central processing unit will execute it. Another proposed system uses Arduino UNO, which is an open-source microcontroller, easy-to-use hardware, and software, that can execute a command easily, whether to turn on an LED or activate a motor by programming the commands to the board with the programming language IDE (Sharma, Ansari, Siddiqui & Baig, 2017). In addition, Katravath, M., Prasad, Arulananth (2019) proposed a system using a different member of the Arduino family, that is; Arduino Mega. The Arduino Mega is connected serially to a Raspberry pi, along with other sensors. The only difference between Arduino Mega and Uno is that Mega has more memory space and more pins. It is a bigger version of UNO. Meanwhile, as alternatives, as stated by Vatsal, & Bhavin (2017), Raspberry pi is used as a microprocessor. It runs with Python programming that can access real-time control with a real-time operating system of electronics. An ARM processor is used in a study by Mahgoub, Tarrad, Elsherif, Al-Ali, & Ismail, (2019), to transfer the results from sensors over a long distance using a GPRS unit. ARM is an architecture with several processors built into its chip. Studies conducted by Sharma, Ansari, Siddiqui & Baig (2017) and Keshamoni & Hemanth, (2017) mention the ethernet shield, ESP8266 WiFi module device is used to connect the microcontroller to the local WiFi. It can host an application as well as offload networking functions from the applications to the processor. It also occupies a minimal PCB area.

Another alternative for the user to interact with the device is the GPRS/GSM module. Other studies have used the GPRS/GSM module for interfacing with the internet and the user. Global System for Mobile communication (GSM) establishes communication between a mobile device and GPRS system using SMS or call service. After the data is collected from the sensor and sent to the processor, the user will be alerted whether a fire has occurred and for necessary action to be taken. Although this module is outdated, it is worth mentioning due to its popularity (Katravath, M., Prasad, Arulananth, 2019; Keshamoni & Hemanth, 2017 and Piera & Salva, 2019).

Cloud Server

In an IoT system, the data is first collected, processed, and stored in a cloud server (Li, Y., 2018). A cloud server does not have any physical attributes, and hence, does not occupy space or require any wiring. All the data collected and processed is accessed through the cloud server by the user remotely, through an internet connection. One of the common examples of Cloud servers is Amazon Web Service (AWS) (Kang, Park, Kim, Kim, D. Y., Kim, S. H., Son, H. J., & Lee, S. G., 2017). It is a platform to store and compute the data, as it also supports the MQTT protocol. However, Li and Y., (2018) utilized the Zigbee protocol and interfaced it with a Zigbee Coordinator Software that read the data through the cloud server as well as sent commands to be executed.

Benchmarking

A study proposed by Sharma, Ansari, Siddiqui & Baig, (2017) is an IoT-enabled system, proposed to detect forest fire, with key components such as Arduino Uno and Ethernet shield Wiznet W5100. The Arduino UNO is implemented with the ethernet shield for WIFI connection, and the data is accessed using TCP and UDP protocol as the Wiznet provides an IP network. The data is visible through the webpage created by a PHP tool called 'Fire Security System'. To collect the data from the forest, a temperature sensor and a gas sensor (LM35 and MQ6). These sensors will monitor the CO2 levels and temperature. The Arduino is also equipped with an LCD screen as well as a buzzer. The hardware connection is as shown below in Figure 2.

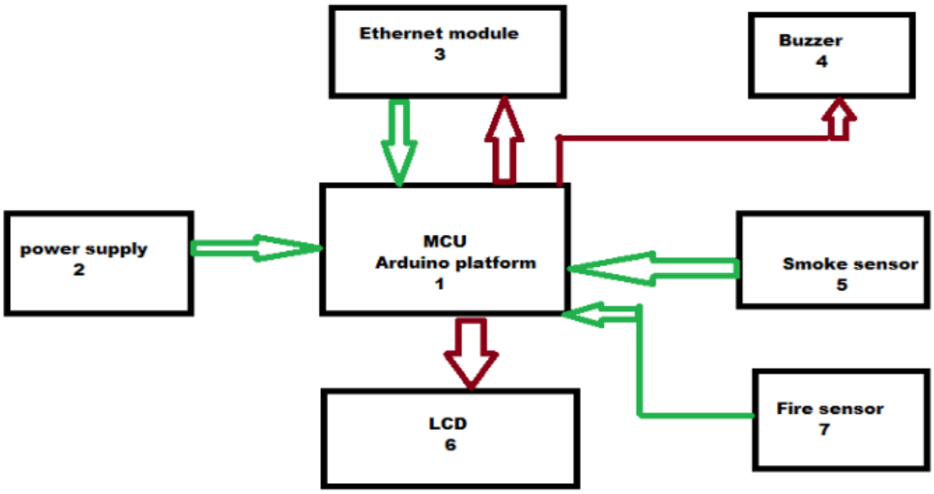

Figure 2 *Hardware connection (Sharma, Ansari, Siddiqui & Baig, 2017).*

Vatsal, S., & Bhavin, M. (2017) made a temperature and humidity sensing system using Raspberry Pi 2 Model B which is crucial in food industries and the pharmaceutical industry. This system makes use of one simple temperature sensor, i.e., DHT22. DHT22. The system uses the New Out of The Box Software (NOOBS) to prepare the Raspberry Pi and store the data. The DHT22 library is downloaded and commands using Python language are written. To observe the results, the private-eye-pi software is accessed. Although the DHT22 temperature sensor is highly accurate, it produces an analog signal and requires an analog to digital converter to retain the data.

A system was proposed by Yan, D., Yang, Y., Hong, Y., Liang, T., Yao, Z., Chen, X., & Xiong, J. (2018) in an article about wireless, passive temperature sensors on a PCB. The author proposed a microwave patch antenna type sensor. This sensor is of simple structure, small size, easy processing, and low cost. It was proposed to overcome the problems caused by the thermocouple type, thermo-resistance type, and p-n junction type sensors that require wiring. Although the sensor promises accurate results for a specific range, it is still theoretical and advanced to corporate into a simple fire alarm system. The prototype of the temperature sensor is shown below in Figure 3.

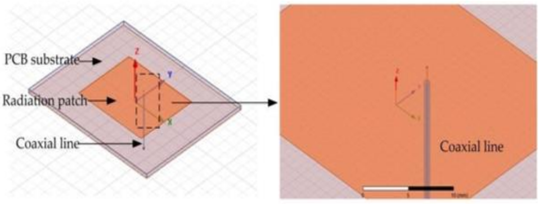

Figure 1 *Schematic of temperature sensor prototype [9]*

A study discusses the placement of a temperature sensor on a printed circuit board. The idea behind embedding the sensor into the product itself is encouraged, as it improves the product's sensor data as well as saves space. To permanently embark the sensor into the product requires precision and placement at the right place. Because placing the sensor incorrectly can cause inaccurate data from the sensor. The author proposes a method to calculate the required parameter of the sensor, based on the PCB's layout (Neiser, A., Seehase, D., Fink, A., & Nowottnick, M., 2015). Attached below is a layout of the PCB as suggested by the utility software in Figure 4.

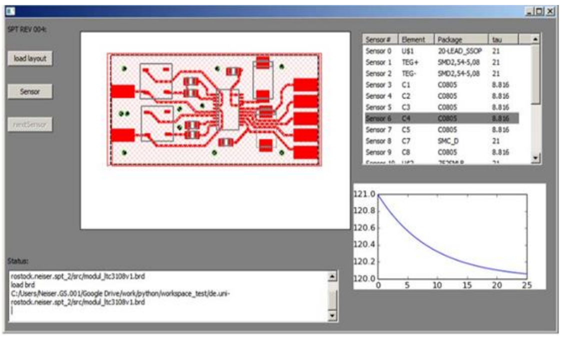

Figure 2 *Layout of PCB from Utility software (Neiser, A., Seehase, D., Fink, A., & Nowottnick, M. 2015)*

Sindhuri, B., & Haritha proposed a smart home automation system, to control the room temperature and lights away from home. An ARM processor is equipped with a fire sensor, temperature sensor, light sensor, sprinkler, and buzzer. Here, the user is alerted and able to control the automation system using 802.11 WiFi Protocol. Through the Think Speak platform, the user can access the house virtually.

A Study uses a Raspberry Pi microcontroller to communicate with an ad-hoc network that has several nodes distributed at different areas of the house. Each node can detect fire through the temperature and smoke sensors attached to it, and it can send the signal to Raspberry Pi through a 4G module via an SMS to the user's smartphone as well as the fire department. This proposed system utilizes the Message Queuing Telemetry Transport (MQTT) protocol to communicate between each node. Figure 5 below displays the prototype of the proposed system (Mahgoub, A., Tarrad, N., Elsherif, R., Al-Ali, A., & Ismail, L., 2019)

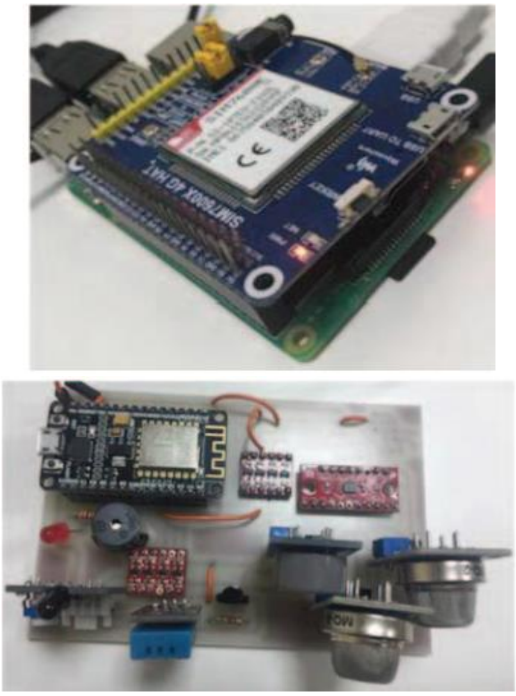

Figure 3 *Implementation of central Node & sensing node.*

However, this system has several sensing nodes that is dependent on an external power source. This is a disadvantage if a fire were to occur during a power outage. The system is based on a bridge node that is used to communicate between the central node and other sensing nodes. This causes a delay in sending an SMS to the user, with an average of 22.58 seconds' delay. The distance between each node should also not exceed a value of 21m when walls are present (Mahgoub, A., Tarrad, N., Elsherif, R., Al-Ali, A., & Ismail, L., 2019)

Another Study proposed a system, to minimize false alarms and make the system more reliable. The proposed system consisted of a GSM modem that can alert the user remotely as well as an automatic water sprinkler that sets off when the threshold values of the sensors are exceeded. The proposed system required a 12V pump to control the water movement from the tank to the sprinkler. An LED and a buzzer were also used in the system to visualize any signs of danger (Alqourabah, H., Muneer, A., & Fati, S. M. 2020).

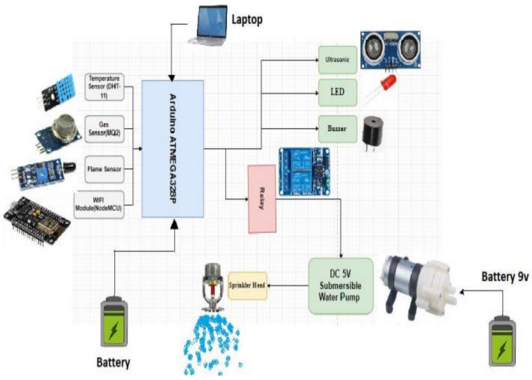

Figure 4 *Block Diagram of Smart Fire Detection System with Automatic Water Sprinkler*

The system suggested realize improvements that should be made to existing fire alarm systems, with an addition of automated water sprinklers and embodied with an IoT system. However, the system requires an external power supply to operate the water sprinklers which may not be convenient in case of an outage.

Table 2 shows a summary of comparison between our proposed system and the benchmarked system (Alqourabah, H., Muneer, A., & Fati, S. M. 2020).

Table 2 . *Comparison with Benchmarking*

Parameters	Proposed System	Benchmark	Limitation/Gap of Benchmark System
IoT component	ESP-01	GSM	GSM limited to SMS and requires a sim card
PCB model	Single layered, one unit	No PCB	Maintenance of cables

III. METHODOLOGY

This section consists of descriptions of each component used to design a Printed Circuit Board integrated Internet of Things-based fire alarm system. The chosen sensor components are; DHT22 temperature sensor and MQ2 gas sensor. The IoT node chosen is an Arduino Uno R3 and the communication gateway chosen is an ESP8266 WIFI module.

Sensors Flowchart

The sensors chosen for this fire alarm system are the DHT22 and MQ2 gas sensors. The flowcharts of these sensors are similar. The sensors only collect data when the sensors are connected to the Arduino board and booted up. The sensors will check the connection to the Arduino board to send the data from its surroundings. If the connection is faulty, the system remains in sleep mode and if the connection is established, Arduino Uno obtains the sensor readings from its surroundings.

Data Flowchart

The data received from the sensors are processed by the Arduino Uno and sent to the

ESP8266 WIFI Module for communication purposes. In Figure 7 shown below, the connections between the board and WIFI module will be checked, and if it is established, the WIFI module connected to the user's WIFI network is also checked. If the connections are faulty, the system will remain in sleep mode. Once the connections are established, the data is collected and processed. If the measured value is lower than the desired value, the processor continues to read the data from the processor. If the measured value exceeds the desired value, the data will be sent to the web application Blynk which will alert the user graphically.

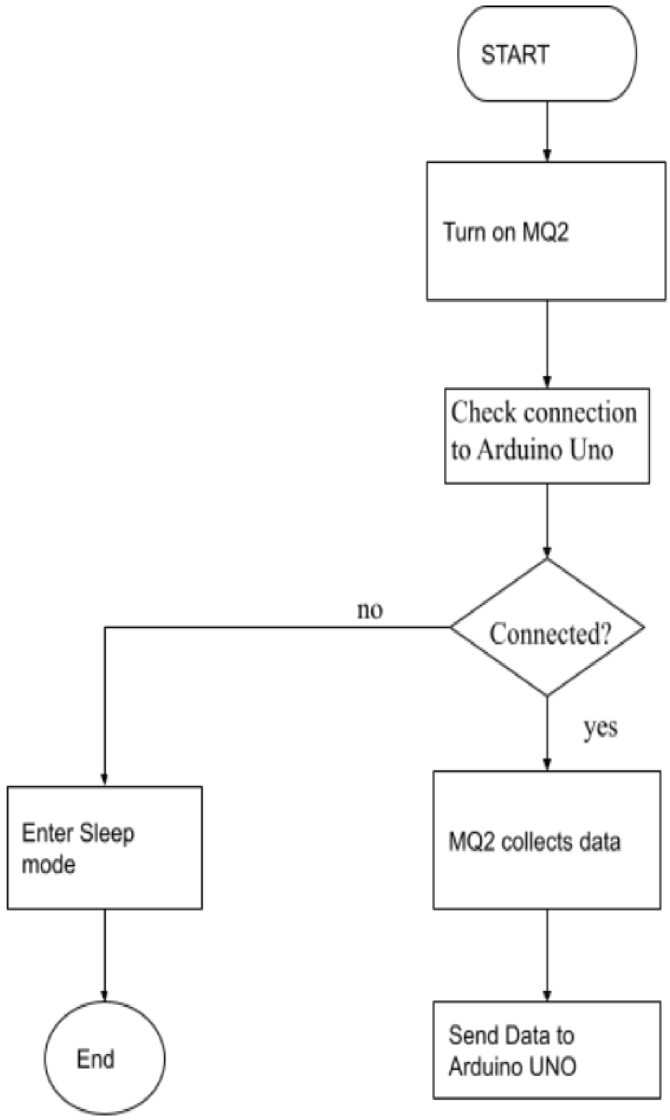

Figure 7 *Data Flowchart*

PCB Fabrication Flowchart

To integrate the components on PCB, initially, the designing stage must be implemented. This is where the components are placed at their optimum position to minimize the size of the board. The connections are made through copper tracks and all the specifications for the board size, track width, pad clearance, etc. are set at this stage. The designed board's layout is printed on a transparency sheet to fabricate the PCB. After developing the design, the process to fabricate the PCB commences. This is done through a methodological process that begins at UV exposure and is followed by developing, etching, drilling, and soldering the components. Figure 8 illustrates the process below.

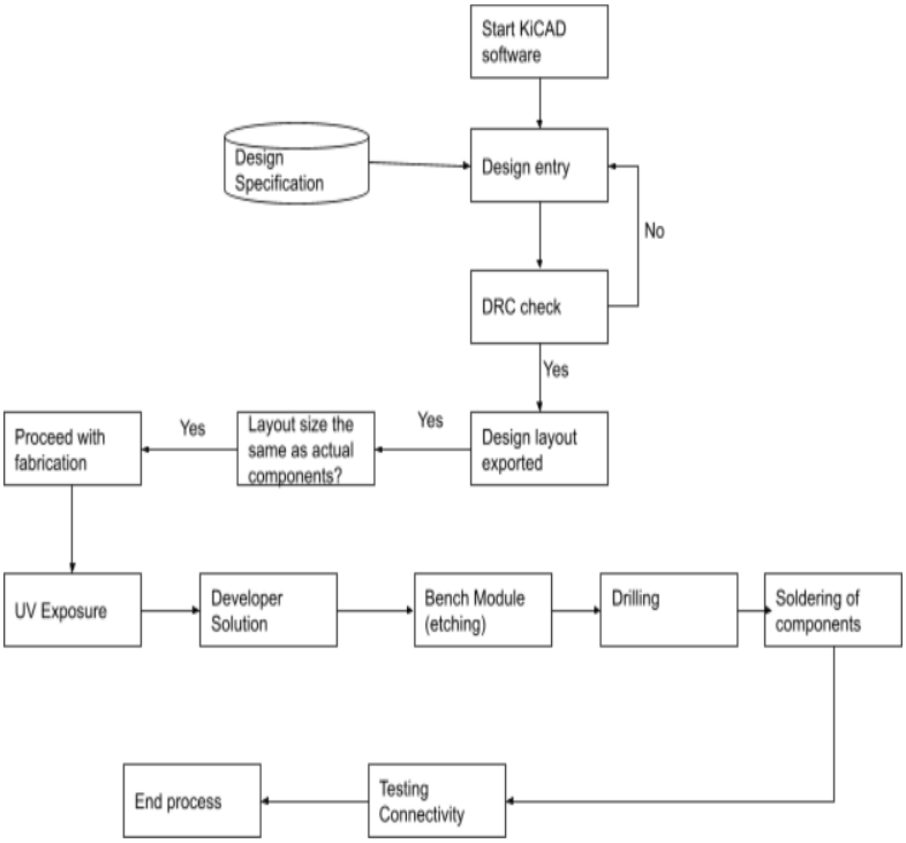

Figure 8 *PCB Fabrication Flowchart*

Parameters and Board Setup for KiCAD

KiCAD is online design software used to design a printed circuit board per the specification that has to be met. It comprises a schematic view, PCB new layout view, and GerbView to design the board at different stages. The following parameters were chosen to achieve the best possible outcome. The components chosen in the schematic view are assigned to their footprints with the following dimensions in Table 3 and Table 4.

Table 3 . *Dimensions of components*

Components	Dimensions (Inches)
PCB board	2.54 x 1.98
MQ2	1.433 x 1.047
DHT22	0.4985 x 0.2362
Bi-directional logic converter	0.63 x 0.52
ESP-01	0.96 x 0.57
Mounting holes	0.1377
Resistor 10K	0.274 x 0.082

These sizes were chosen following the actual size of the components, by measuring the size of the whole component as well as the size of the holes connected to its pins with digital calipers.

Table 4. *Components mounting holes specifications*

Components Drill Size	Pad Type, Shape	Size X (mm)	Hole Size (mm)
Logic level converter	Through-hole, Circular	2.032	1.524
ESP-01	Through-hole, Oval	1.72	1.016
DHT22	Through-hole, Circular	2.032	1.524

IV. FINDINGS AND RESULTS

Parameters of components were slightly varied to achieve the best possible results.

Breadboard Setup

To validate the circuit connection prior to integrating the components on the PCB, a breadboard connection was tested. The DHT22 temperature and humidity sensor were connected in series with a resistor from the Data pin to the digital pin 2 of Arduino Uno board, the MQ2 gas sensor was connected to the Analog pin of Arduino (A3) and the ESP-01 was connected with a voltage regulator to draw the desired 3.3V from Arduino Uno. Arduino Uno serial pins 10 and 11 were used to transmit and receive a signal from the ESP-01. Finally, the Arduino Uno was powered through its USB port whilst connecting to the computer, in order to program the system. An online software, circuit.io was used to design the breadboard. Figure 9 shows the final setup of the breadboard.

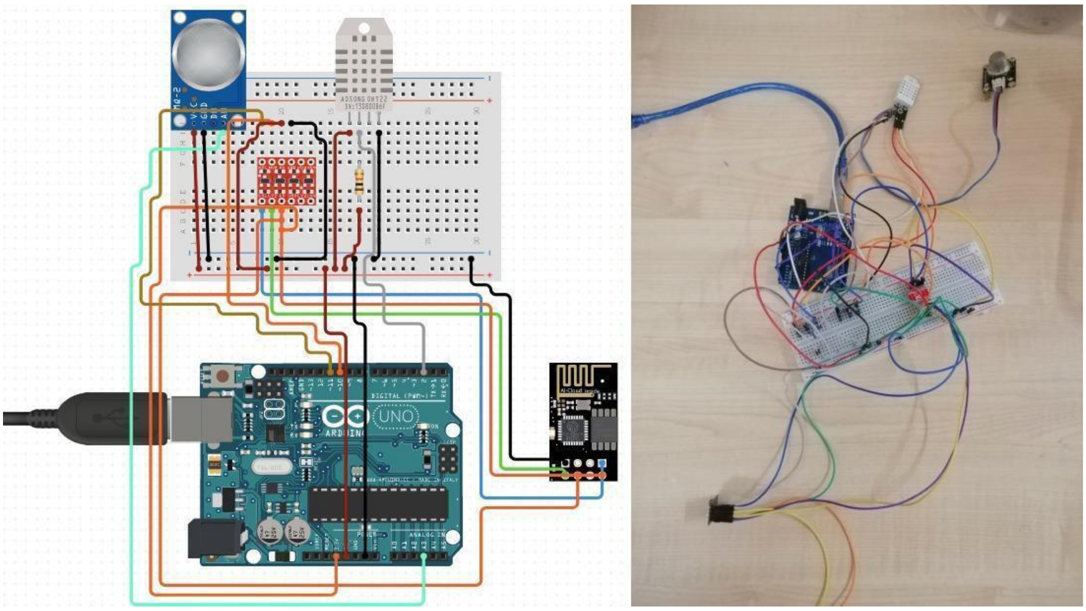

Figure 9 *Breadboard connection for testing*

Results From Sensors on Breadboard

Under normal room temperature, the temperature and humidity were between 30-35

Celsius and 50-60% humidity respectively. Figure 10 below demonstrates the values received by the sensors:

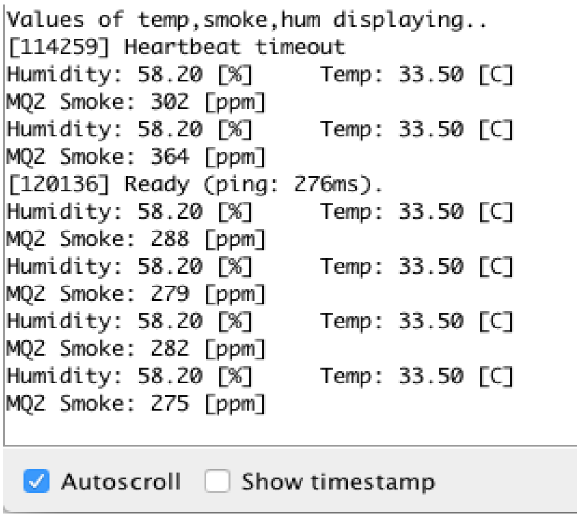

Figure 10 *Output Readings from DHT22 and MQ2*

Under the hazardous condition, where a fire is about to erupt, the command to notify the user through email will be given and the following message will appear.

Figure 11 *Email Notification from Blynk*

Results from Arduino Serial Monitor

The condition was set, so that when the temperature and humidity values reach the threshold temperature and humidity values, the system will print "Fire Alert" in the serial port, as well as send an email notification to the user. The threshold value set in this scenario to imitate a fire eruption was 42 Celsius and 90% humidity. And the MQ2 threshold was set to 440 ppm. If the threshold values are exceeded, the alerts will be given. Hence, under hazardous conditions, the following alert will be displayed as mentioned in Figures 12 below.

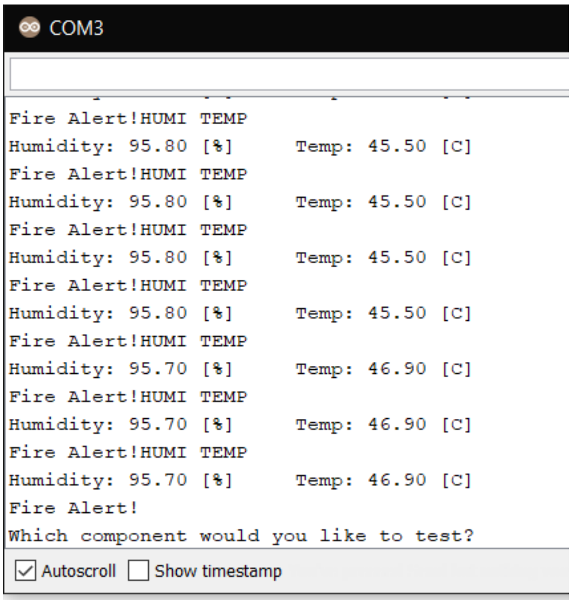

Figure 12 *MQ2 Readings when Threshold Value Exceeded*

Results from the IoT System- ESP-01 and Blynk

The IoT gateway component ESP8266 operates at a voltage of 3.3V. Hence, a bi-directional logic converter was used to regulate the Arduino Uno voltage from 5V to 3.3V. Lastly, a web application Blynk was used to display the readings from the sensors continuously and can be accessed at any time, provided an internet connection is established.

Figure 13 *Blynk App with Widgets*

Figure 13 displays the Blynk Application used and shows the different widgets used to display the readings. The Gauge widget is used to display the individual readings for each variable, where each gauge is assigned to a virtual pin. The Super chart widget shows a comparison of all the readings from both the sensors and gives continuous data within the period chosen. Lastly, the Email notification widget notifies the user in case of a fire, where the command is established in the Arduino coding. Figure 14 below shows the Blynk Application under normal and hazardous conditions. The super chart widget allows the user to visualize the values and even export them into a spreadsheet as shown below.

***Figure 14** Readings from Super Chart under conditioned-fire*

Design and Integration of PCB

At this stage, the testing process is completed, and the fabrication process of the PCB commenced. A single-sided PCB is chosen with appropriate specifications of board size, pad size, and copper track width. The copper trace is routed at the bottom side of the board, whereas the components are placed on the top of the board.

PCB Design using KiCAD software

KiCAD is online design software used to design a printed circuit board per the specification that has to be met. It comprises a schematic view, PCB new layout view, and GerbView to design the board at different stages.

Schematic

The schematic layout in KiCAD is used to add the components in the Eeschema sketch. The components DHT22, ESP-01, and Bi-directional logic converter were chosen and connected accordingly. The sensor MQ2, however, was drilled onto the board through mounting holes. Connectors for the Arduino pins were also added. Figures 15 exhibit the components connected and an Eeschema sketch preview of the system respectively.

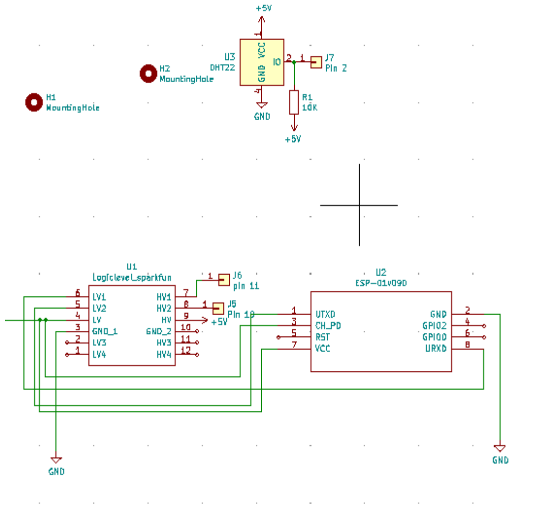

***Figure 15** Schematic Diagram*

PCB Layout

After annotating the components in running the Design Rules Check (DRC) check, the suitable footprints were associated with each component. A preview of the 2D and 3D were generated to verify the design. Then, a netlist is generated and saved in the file location. The PCB layout is updated based on the schematic and viewed on the PCB layout page. Figure 16 below shows the initial stage after generating a netlist and before routing the connections.

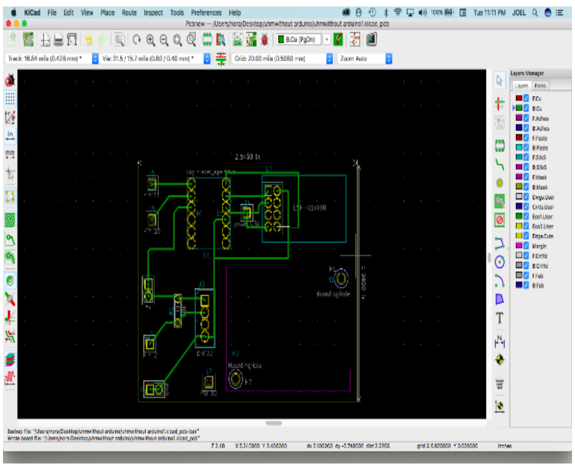

***Figure 16** PCB layout*

GerbView

After measuring the dimensions and placing the components, the ratsnest is untangled by moving and rotating the components that produce the optimum design and the least amount of track routes. After placing the components, the routing is completed on the Bottom layer of the boards (B.u.Cu). Following the completion of routing tracks, the Design Rule Check (DRC) is run to ensure the components pads are not too close to each other and that all routes are connected. Succeeding the completion of the PCB layout, the diagram is plotted on GerbView through KiCad. It is used to display drill files from the PCB layout. Hence, this file prints the track routes and outlines of areas to be drilled. Figure 17 below illustrates the PCB layout as well as the GerbView plot. The layer polarity chosen is the positive display.

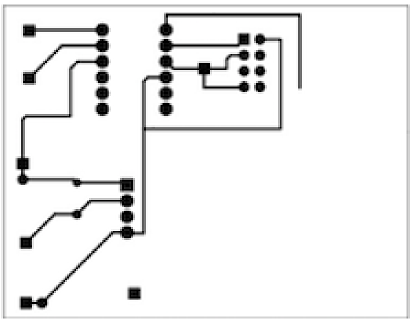

Figure 17 *GerbView of Bottom Copper Layer*

PCB Board Fabrication

At this stage, the design simulation is complete and prepared for fabrication. The components are physically integrated on a copper-routed circuit board through the following steps. It begins with UV exposure followed by the developer, bench module, photoresist stripper, drilling, and lastly, the soldering of components onto the PCB. Soldering is done to establish a connection between the component and the copper traces. Without soldering, a connection cannot be made.

Jumper wires, connectors, and resistors were soldered in the beginning, followed by the components MQ2, DHT22, Logic level, and ESP-01. Figure 18 below shows the final product after completing all the processes.

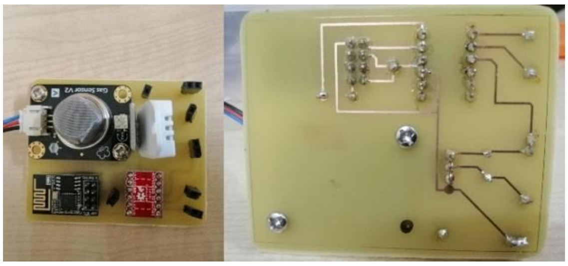

Figure 18 *PCB end-product fabrication*

Functional Verification

The readings are taken after the printed circuit board is fabricated. We can conclude that the sensors worked accurately and showed continuous uniform values under room temperature and non-hazardous conditions.

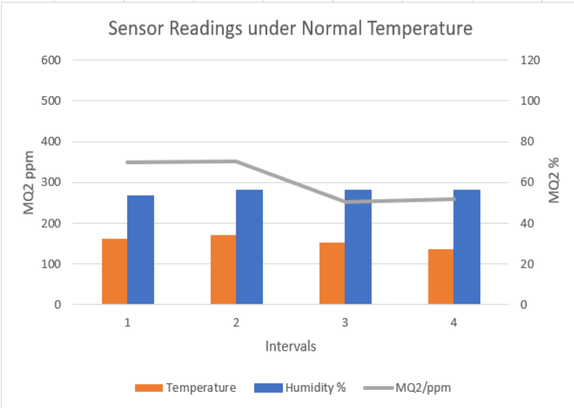

Figure 19 *Readings from MQ2 and DHT22 under Normal Temperature*

Under circumstances of a fire eruption, the readings taken from DHT22 and MQ2 sensors are as shown below through Blynk in Figure 20

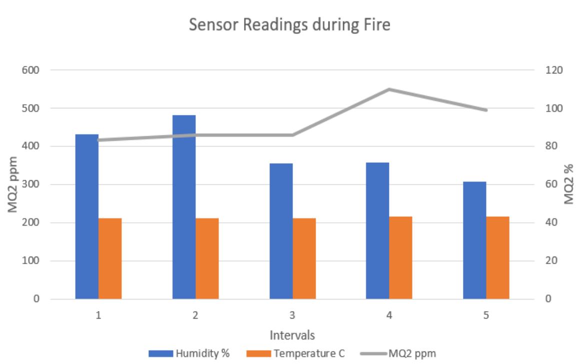

Figure 20 *Readings from DHT22 and MQ2 when Fire has erupted*

Figures 19 and 20 below illustrate the readings from DHT22 and MQ2 over a few minutes. The time axis can be altered to get data continuously for 15 minutes or more than 3 months. Figure

20 suggests that the temperature crossed the threshold value of 40 Celsius and the smoke detected by the DHT22 sensor was above 75% and came close to 100%. The MQ2 values displayed have reached a maximum of 552 ppm, meanwhile, DHT22 detected humidity at a maximum of 90%. The threshold was set to 400 ppm and 60% for the DHT22 sensor. These graphs suggest that the DHT22 sensor operated with more accuracy than MQ2. Although, it is worth noting that the circumstance of fire was set on a small-scale and that the scarce amount of fire could have affected the readings from sensors. The final prototype is displayed below in Figure 21.

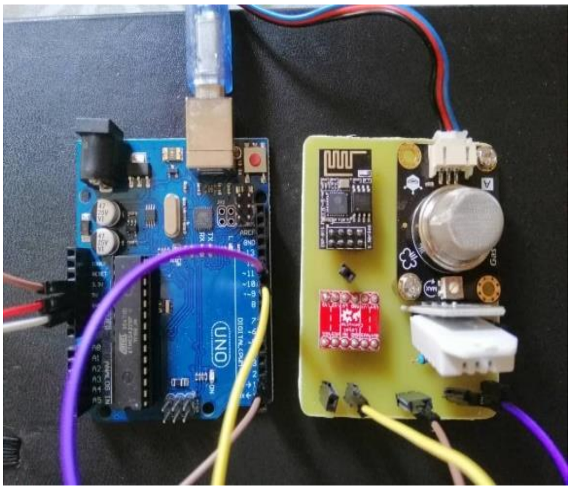

Figure 21 *The Proposed System with Arduino*

V. CONCLUSION

This proposed PCB design for an IoT-based fire alarm system is discussed throughout the chapters from the beginning until the completion of this report. The proposed fire alarm system has advantages over existing fire alarms in terms of size, IoT gateway used, and the use of one-layered PCB. The main components are chosen based on expense, reliability, and availability. Although, due to the small-scale fire that was made, to test the fire alarm system, the threshold values had to be altered to be lesser than the actual threshold value. Therefore, the MQ2 sensor detected less smoke and provided an alert to the user. Hence, we can say that the DHT22 sensor and the MQ2 sensor are the optimum components for the system.

The PCB integration of components was done through designing software followed by fabricating a single-sided PCB. The purpose of the system is to choose components that give proximity results and to minimize the size of the whole system by integrating the system on a PCB. The proximity of the sensors used in the proposed system far exceeds the reliability and proximity of other sensors used in previous models. The methods taken to complete the system are shown and discussed in this report. Hence, all the objectives have been met such as,

- Choosing optimum components and testing on a breadboard before manufacturing
- Designing the PCB on software to produce a print out the layout
- Manufacturing a single-sided PCB and soldering components on the board
- To analyze the performance of the circuit after PCB manufacturing and achieving data successfully.

ACKNOWLEDGMENT

The authors would like to thank School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA (UiTM), Shah Alam and Department of Electrical & Computer Engineering, Kulliyyah of Engineering, International Islamic University Malaysia (IIUM) for their valuable support. This research is partly funded by the government through UiTM under 600-IRMI/5/3/BESTARI (122/2018).

REFERENCES

1. Artim, N. 3.2 An Introduction to Fire Detection, Alarm, and Automatic Fire Sprinklers. Retrieved February 08, 2018.
2. Alqourabah, H., Muneer, A., & Fati, S. M. (2020). A Smart Fire Detection System using IoT Technology With Automatic Water Sprinkler. International Journal of Electrical and Computer Engineering (IJECE), 9(4).
3. 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. Int'l J. of Communications, Network and System Sciences, 7(09), 386.

4. Fire Detection and Warning, © 2020 Health & Safety Authority Retrieved from https://www.hsa.ie/eng/Topics/Fire/Fire_Detection_and_Warning/
5. Ghasemzadeh, K., Tilebon, S. M. S., Nasirinezhad, M., & Basile, A. (2018). Economic assessment of methanol production. In Methanol (pp. 613-632). Elsevier.
6. Katravath Ravi, M.Raju naik, S.V.S Prasad, Arulananth T S (2019). Fire Alarm Robot and Authentication System Using Raspberry Pi and Cloud. International Journal of Innovative Technology and Exploring Engineering (IJITEE) 08, 4S2,256-259.
7. Keshamoni, K., & Hemanth, S. (2017, January). Smart gas level monitoring, booking & gas leakage detector over IoT. In 2017 IEEE 7th International Advance Computing Conference (IACC) (pp. 330-332). IEEE.
8. Kang, D. H., Park, M. S., Kim, H. S., Kim, D. Y., Kim, S. H., Son, H. J., & Lee, S. G. (2017, February). Room temperature control and fire alarm/suppression IoT service using MQTT on AWS. In 2017 International Conference on Platform Technology and Service (PlatCon) (pp. 1-5). IEEE.
9. Li, Y. (2018, April). Design of smart home gateway based on Wi-Fi and ZigBee. In AIP Conference Proceedings (Vol. 1955, No. 1, p. 040112). AIP Publishing LLC.
10. Mahgoub, A., Tarrad, N., Elsherif, R., Al-Ali, A., & Ismail, L. (2019, July). IoT-based fire alarm system. In 2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4) (pp. 162-166). IEEE.
11. Neiser, A., Seehase, D., Fink, A., & Nowottnick, M. (2015, May). Placement of embedded temperature sensors in a printed circuit board for a manufacturing process. In 2015 38th International Spring Seminar on Electronics Technology (ISSE) (pp. 213-217). IEEE.
12. Piera, P. J. Y., & Salva, J. K. G. (2019, July). A Wireless Sensor Network for Fire Detection and Alarm System. In 2019 7th International Conference on Information and Communication Technology (ICoICT) (pp. 1-5). IEEE.
13. Sharma, A. K., Ansari, M. F. R., Siddiqui, M. F., & Baig, M. A. (2017). IoT enabled forest fire detection and an online monitoring system. Int. J. Curr. Trends Eng. Res. (IJCTER), 3, 50-54.
14. Sindhuri, B., & Haritha, G. Room temperature controlling fire alarm suppression and lightning conditions using 802.11 protocols.
15. Vatsal, S., & Bhavin, M. (2017). Using Raspberry Pi To Sense Temperature And Relative Humidity. Int. Res. J. Eng. Technol, 4(2), 380-385.
16. Yan, D., Yang, Y., Hong, Y., Liang, T., Yao, Z., Chen, X., & Xiong, J. (2018). Low-cost wireless temperature measurement: design, manufacture, and testing of a PCB-based wireless passive temperature sensor. Sensors, 18(2), 532.