ISSN 2278-3091
Volume 9, No.1.5, 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse4791.52020.pdf
https://doi.org/10.30534/ijatcse/2020/4791.52020

IoT based Vehicle Carbon Monoxide Monitoring, Alerting and Controlling System

Goh Siew Yen[1], Kohbalan Moorthy[1]*, Logenthiran Machap[2]
[1]1Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Kuantan, 26300, Malaysia, kohbalan@ump.edu.my
[2]Department of Computing and Information Technology, Tunku Abdul Rahman University College, 85000 Segamat, Johor, Malaysia, logenthiran@tarc.edu.my
*Corresponding author

ABSTRACT

Motor vehicle exhaust fumes are one of the major sources of carbon monoxide. Carbon monoxide leaking from the exhaust system can enter the vehicle through holes. Inhaling a certain concentration and duration of carbon monoxide can cause fatal carbon monoxide poisoning. It is necessary to monitor the air quality in the vehicle to make sure the level of carbon monoxide is safe. The goal of this project is to develop an IoT based Vehicle Carbon Monoxide Monitoring, Alert and Controlling System. The system detects the level of carbon monoxide in the vehicle using the MQ-7 gas sensor and Arduino. The sensor data collected will be transferred to ThingSpeak cloud over the GPRS network using the GSM module and Arduino. The system allows the user to monitor the level of carbon monoxide and turn off the vehicle remotely using the Android app. The user also can receive alert notification for the dangerous level of carbon monoxide. The system able to send text messages to the emergency contact and the user if the level of carbon monoxide is dangerous and the user does not turn off the vehicle in each time. The system also can provide effective automated countermeasures when the level of carbon monoxide is very dangerous. The system is cost-effective and can be installed in the vehicle to control the ignition system for turning off the engine.

Key words: Arduino, Carbon Monoxide, GSM module, Internet of Things, ThingSpeak, Vehicle,

1. INTRODUCTION

One of the major sources of carbon monoxide is motor vehicle exhaust fumes. The internal combustion gasoline engine will produce a high concentration of carbon monoxide. The common causes of carbon monoxide poisoning are staying or sleeping in a vehicle with the engine running, running a vehicle in closed space, running a vehicle with faulty exhaust and holes in the car body. Carbon monoxide leaking from the exhaust system can enter the vehicle through holes or windows or doors[1]. Inhaling a certain concentration and duration of carbon monoxide can cause serious health problems and even death due to carbon monoxide poisoning. Carbon monoxide affects humans by replacing oxygen to bond with haemoglobin in the blood to form carboxyhemoglobin and reduce the oxygen supply to the body [2]. PPM is the unit of the concentration of carbon monoxide. The limits of carbon monoxide in Malaysia are recommended by the Department of Environment (DOE) and the Department of Occupational Safety and Health (DOSH). It is necessary to monitor the air quality in the vehicle to make sure the level of carbon monoxide is safe to ensure the health of the driver and passenger in the vehicle. Internet of Things (IoT) is a technology that allows things to communicate by transferring data over the network to make decisions. Physical devices can be attached with sensors and assigned with IP addresses to transfer data and trigger some actions depend on the data collected. IoT enables the user to monitor or control automobiles electronic or other physical devices via the Internet. It has been implemented in various industries, such as the health care centre and manufacturing industry, to improve their existing systems[3, 4].

There are some motor vehicle-related carbon monoxide poisoning cases in Malaysia that cause serious health problems and death. A young woman found unconscious in a car with the engine running and was in a coma for three months after suffered carbon monoxide poisoning. Besides, two teenagers found dead, believed to be from carbon monoxide poisoning, after sleeping in the car with its engine running and air-conditioning on[5]. There also some cases where drivers had left their children unattended in cars with engines running and they may suffer from carbon monoxide poisoning if being trapped in the car for several hours. Most of the vehicles, especially the car with an older model year and build date does not have a carbon monoxide detection system. Existing carbon monoxide detector does not have an effective countermeasure for a dangerous level of carbon monoxide and unable to interact with the vehicle to turn off the engine to stop the source of carbon monoxide. Carbon monoxide is still being released and build up in the vehicle. People unaware of carbon monoxide leaking into the vehicle through holes and cause carbon monoxide poisoning if they inhale the gas for a certain amount and duration [6].

There are three objectives in this project, which are to study and analyze the existing carbon monoxide detection system for a vehicle. Then, to design and develop an IoT based prototype to detect carbon monoxide, alert the user and provide effective automated countermeasure. The final objective is to test and validate the prototype. The scope of this project is the detection is only for carbon monoxide gas. The system detects carbon monoxide in a vehicle cabin using the MQ-7 gas sensor and Arduino. Users able to monitor the level of carbon monoxide and receive alert notification using the Android app. The system aims to be tested and used by the staff and students of UMP. The significance of this project is the implementation of IoT technology in this system allows the user to monitor the level of carbon monoxide in a vehicle cabin remotely using the Android app. The system can send alert notification to the user using an Android app for a dangerous level of carbon monoxide. The system able to send text messages to the emergency contact and the user if the level of carbon monoxide is dangerous and the user does not turn off the vehicle in each time. The system can be installed in the vehicle to control the ignition system for turning off the engine.

2. LITERATURE REVIEW

In this section, three existing systems will be analyzed and compared to find more suitable solutions to be adopted into the project to solve the problems. Car Indoor Gas Detection System is a low-cost Arduino based carbon monoxide detection and alarm system that can be installed in a vehicle easily [7]. The system can measure the level of carbon monoxide in a vehicle cabin using two MQ-7 gas sensors[8]and an Arduino Mega 2560[9]. An Android app is connected to the Arduino via Bluetooth connection. The system will activate the buzzer alarm and blink the LED and show the warning message in the Android app to alert the user when there is a leak of carbon monoxide inside the vehicle. If the user does not respond to the alarm for 10 s, the system will send a warning text message to emergency contacts with the geographic coordinate of the vehicle, which is detected using the smartphone GPS [10].

Embedded System for Vehicle Cabin Toxic Gas Detection and Alerting is designed to detect the level of carbon monoxide and oxygen in the vehicle cabin by using an Atmel 89c51 microcontroller, carbon monoxide sensor, and oxygen sensor[11]. The level of the carbon monoxide and oxygen in the vehicle is read for every second and is displayed on the LCD screen. The audible alarm circuit is designed to alert the user when the level of carbon monoxide reaches 30 PPM and the level of oxygen is below 19.5 %. The system can send a warning text message to the authorized user via the GSM module and provide ventilation automatically when the toxic gas reaches the maximum threshold level[12, 13].

An Intelligent Green Gas Detector model is designed to monitor the emission of carbon dioxide and carbon monoxide in vehicles in real-time using the Cognitive Internet of Things (CIOT) [14-16]. The CIOT has been used to interface the devices to communicate without human intervention and generate integrated data that will be processed by intelligence for decision-making. The system uses Raspberry Pi, MQ7 gas sensor, MQ135 gas sensor, MySQL server, and PHPMyAdmin. The system consists of two modules, which are the owner module and the central board module. The vehicle owner needs to install the system at the vehicle exhaust. The sensors communicate with the Raspberry Pi to detect the level of gases emitted in terms of PPM. Then, the values are averaged and stored in the local database of Raspberry Pi and sent to the webserver over the Internet. The system will send an email to alert the owner to rectify the vehicle if the emissions exceed the threshold level. The owner can log-in to the website to check his emission data. If the owner does not take any action to reduce the emissions, an alert is sent to the central board. The central board module will receive the alert and the credentials from the vehicle. The system is cost-effective and can be easily produced and integrated with vehicles [17, 18]. Table 1 shows a comparison of the three existing systems.

Table 1: Comparison of the three existing systems

System	Car Indoor Gas Detection System [19]	Embedded System for Vehicle Cabin Toxic Gas Detection and Alerting[20]	Intelligent Green Gas Detector [21]
Monitoring	No	Yes (LCD screen)	Yes (Website)
Alarm/ Alert	Buzzer alarm, LED blinking and Android app	Buzzer alarm	Email
Hardware/ Software	Arduino Mega 2560, 2 MQ-7 gas sensors, HC-05 Bluetooth module, buzzer card, LED and Android app	Atmel 89c51, LCD screen, carbon monoxide sensor, an oxygen sensor, an alarm circuit, and GSM Modem	Raspberry Pi, MQ7 gas sensor, MQ135 gas sensor, MySQL server, PHPMyAdmin
Features	Wireless Bluetooth connection between the Android app and Arduino. Send a text message to emergency contact with the location of the vehicle. Alarm history.	Detect carbon monoxide and oxygen. Send a warning message to the authorized user. Provide ventilation.	Cognitive IoT. Detect carbon monoxide and carbon dioxide. Uses of local and online databases. Website monitoring. Send email to alert the owner and

			central control board.
Disadvantages	Unable to monitor the level of carbon monoxide Range of data transmission via Bluetooth is limited Unable to interact with the vehicle	Local device	Installed at the exhaust system, gases emitted from other vehicles may affect the accuracy of the data Unable to interact with the vehicle

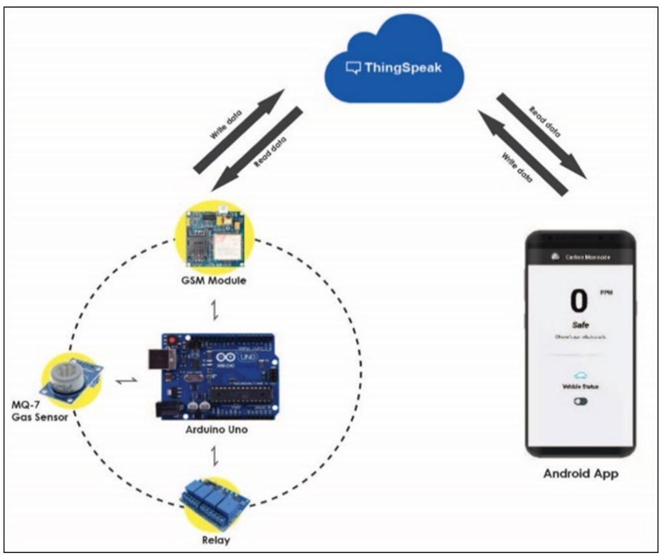

Figure1: System overview

3. METHODOLOGY

The overview of the IoT based Vehicle Carbon Monoxide Monitoring, Alert and Controlling System is shown in Figure 1. Figure 2 (a) and Figure 2 (b) show the flowcharts of the system.

Arduino Uno R3 is an open-source microcontroller which its board is based on ATmega328P. It has 14 digital pins for input or output, which six of them can be used as PWM outputs, and six analogue inputs. It can be powered with a USB connection or external power supply 8 – 12 V.

GSM SIM900A module (Mini V3.8.2) is built with dual-band GSM/GPRS. It is connected to Arduino Uno and communicate using AT commands. It enables the user to send or read the text message, make, or receive a voice call, and connect to the internet via GPRS with an active SIM card. The module is powered with a 5 V external power supply. The GSM module enables data to be transferred between Arduino, ThingSpeak and Android app over the GPRS network, and it is used to send text messages to an emergency contact and the user

MQ-7 is a long-life gas sensor that has a high sensitivity to carbon monoxide as its sensitive components is made of tin dioxide with stability. It can be used to detect carbon monoxide in the car or industry. The sensor needs to be powered with 5 V. The sensor must run through high-heating and low-heating cycles to get proper measurements. The sensor requires a heater voltage that cycles between 5 V (high-heating) for 60 s and 1.4 V (low-heating) for 90 s. During the low-heating phase, CO is absorbed on the plate for measurement. During the high-heating phase, absorbed CO and other compounds evaporate and cleared from the sensor plate for the next measurement. The detection ranges from 10 to 1,000 PPM carbon monoxide. The output of the sensor is an analogue value, which needs to be converted to the level of carbon monoxide in PPM value. The higher the concentration of the carbon monoxide, the higher the voltage output. The sensor is recommended to do calibration to adjust the sensitivity.

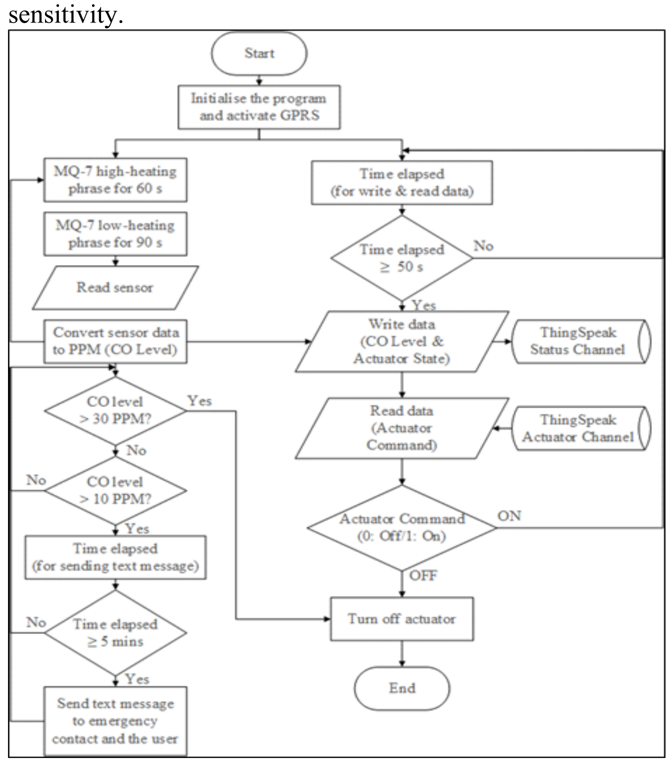

Figure 2 (a): Flowchart of the Arduino

A relay is an electrically operated switch that operates with a relatively low voltage to turn on or turn off a high voltage circuit. It can be used to disconnect the connector at the vehicle ignition system and turn off the engine. The relay has Normally Open (NO) contact, Normally Closed (NC) contact and Common (COM). COM is the centre terminal. For NO, the relay will be in open contact state and no current flows through when the relay is not triggered. On the other hand, NC allows current flow when the relay is not triggered.

As the feature of turning off the vehicle engine will require some wiring to the vehicle ignition system, hence an LED is used to represent the vehicle in this project and prototype development. If the LED is on, it indicates that the vehicle engine is running. The LED will be connected to the relay and

Arduino Uno. The relay will act as a switch to turn off the LED to indicate the engine is turned off. The vehicle or the LED and relay are known as the actuator of the system.

MIT App Inventor is an open-source web application that is used to create a fully functional Android app for smartphones or tablets. It is simple and easy to use. The application is written in Java and Kawa Scheme. In this project, the Android app is programmed with the MIT App Inventor.

Arduino IDE is an open-source software to write code and program the Arduino board. It can operate on Windows, Linux, and Mac OS X. The program is written in C or C++. The sketch can be uploaded to the board by using the USB B type cable. The process running in the Arduino board can be monitored using the serial monitor.

ThingSpeak is an open-source IoT application platform that allows real-time sensor data collection and stores the data in the cloud privately or publicly over the network. Channel must be created to send and store data. It provides API keys for sending data to the channel and reading the data from the channel [22]. In this project, two ThingSpeak channel, which is the ThingSpeak Status Channel and ThingSpeak Actuator Channel is created to store the data collected.

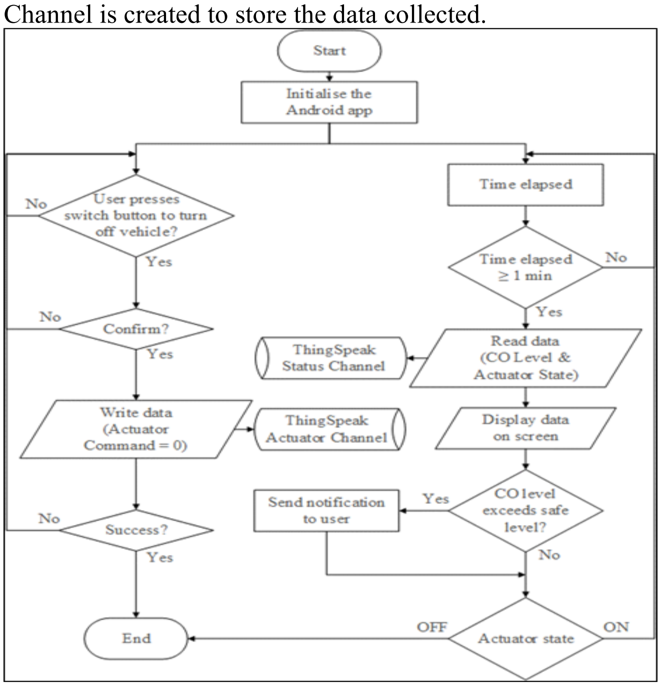

Figure 2 (b): Flowchart of the Android app

4. IMPLEMENTATION AND TESTING

All the hardware components of the system need to be attached. The Arduino is programmed with Arduino IDE. Figure 3 shows the hardware installation of the system. First, the MQ-7 gas sensor is installed to Arduino Uno and configured. The sensor will be supplied with a high heating voltage for 60 s and a low heating voltage for 90 s. The Arduino will read the sensor at the end of the low-heating phrase to get the measurement. The system will obtain new sensor data every 2 mins and 30 s.

Next, the GSM module is attached to Arduino Uno and it needs to be powered on with a 5 V external power supply. It is controlled using AT commands. Besides, an active SIM card must be inserted in the SIM cardholder of the GSM module to be able to get GPRS connection and send text messages. If the level of carbon monoxide is dangerous, and the user does not turn off the vehicle, the system will send text messages to emergency contact and the user after 5 mins.

Private ThingSpeak Status Channel is created to store the sensor data and actuator state, and the public ThingSpeak Actuator Channel is created to store the actuator command, which is the user input that will be sent by the Android app to turn off the actuator. The sensor data and the actuator state are written to ThingSpeak Status Channel approximately on the interval of 50 s using the Arduino and GSM module over the GPRS network. The Status Channel Write API key is configured to allow the Arduino to write data to the channel. After writing the data to ThingSpeak Status Channel, the Arduino will read the last stored actuator command from ThingSpeak Actuator Channel. The channel ID of the Actuator Channel is used to allow the Arduino to read the data from the channel.

Then, the relay is attached to the Arduino to act as a switch to turn off the LED. NC and COM of the relay will be used and connected to the LED. The current can flow through the relay when the relay is not triggered. When the relay is triggered, NO will connect to the COM by the electromagnet of the relay to cut the current flow and then turn off the LED.

The actuator command read from the ThingSpeak Actuator Channel should be either “0” or “1”, and anything different from the value will be ignored. If the actuator command received is “1”, the actuator state is still on. If the actuator command received is “0”, the relay will be triggered to turn off the LED. The actuator state will be changed to off and the system will stop running. Besides, the Arduino will turn off the actuator automatically when the level of carbon monoxide exceeds 24 PPM.

Lastly, the MIT App Inventor is used to program and develop the Android app using the block-based coding method. First, design the user interface. A few elements have been used to build the app. Next, the program by designing the blocks. The Android app will read data from ThingSpeak Status Channel approximately every 1 min via the smartphone internet connection and display them on the screen so that the user can monitor the data remotely. The app will send an alert notification if the level of carbon monoxide is dangerous. Besides, the user can turn off the vehicle remotely by pressing the switch button in the Android app. Then, the app will write the actuator command to the ThingSpeak Actuator Channel, and the Arduino will read the actuator command and act on the actuator accordingly.

During the implementation process, it is important to undergo the testing process to ensure that the system able to work correctly and successfully according to the requirements. Any error and problem detected will be fixed and solved immediately. The system will go through the four main stages of the testing process, which are unit testing, integration testing, system testing, and acceptance testing. Acceptance testing is carried out by the selected users from UMP students to get the expected results. The User Acceptance Test (UAT) forms have been given to the users to be filled by them after the test is conducted.

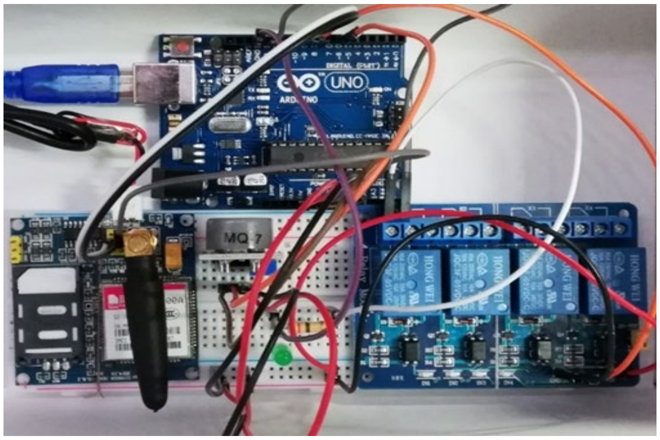

Figure 3: Hardware setup of the system

5. RESULTS AND DISCUSSIONS

After the implementation and testing, the prototype will be set up to work in a real environment. First, set up and powered the hardware devices.

The Arduino initializes and established the GPRS connection. After that, Arduino starts to heat the sensor with high-heating voltage for 60 s and low-heating voltage for 90 s and get the sensor data at the end of the low-heating phrase. At the same time, Arduino sends data to ThingSpeak Status Channel and then read the last actuator command stored in ThingSpeak Actuator Channel on the interval of 50 s.

Meanwhile, the Android app is opened to monitor the status, receive the alert notification, and turn off the actuator when needed. The state of the carbon monoxide and an alert message is displayed based on the level of carbon monoxide obtained. "Safe" for the level of carbon monoxide of 10 PPM and below, "Dangerous" for the level of carbon monoxide from 11 PPM to 30 PPM, and "Very Dangerous" for the level of carbon monoxide for 31 PPM and above. Figure 4 (a) to Figure 4 (c) shows the example of the output for the Android app.

The system sent text messages to the configured emergency contact phone number and user phone number when the level of carbon monoxide is dangerous, and the user does not turn off the actuator after 5 mins. The actuator is turned off automatically if the level of carbon monoxide is very dangerous. The app stops receiving the data update after the actuator is turned off as the system has stopped running.

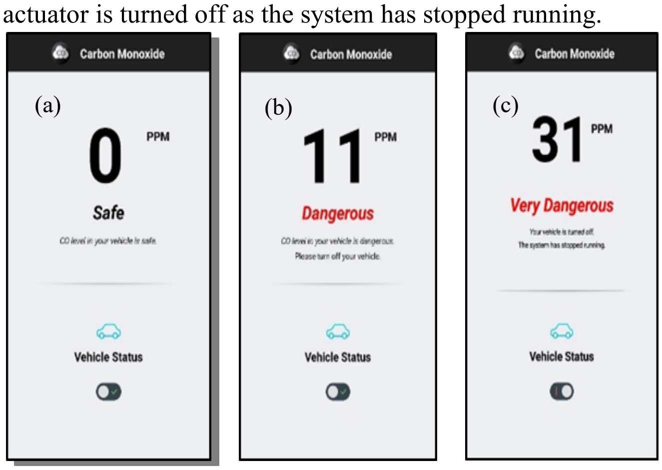

Figure 4:(a) Example of output for a safe level of Carbon Monoxide; (b) Example of output for Dangerous Level of Carbon Monoxide; (c) Example of Output for very Dangerous Level of Carbon Monoxide

6. CONCLUSION

The IoT based Vehicle Carbon Monoxide Monitoring, Alert and Controlling System is cost-effective and can be installed in the vehicle to control the ignition system for turning off the engine. The IoT technology is implemented to allow the user to monitor the level of carbon monoxide in the vehicle, receive alert notification and turn off the vehicle remotely. The range of data transmitting is unlimited as the data is stored in the cloud and transferred over

The benefits of the system are the user able to monitor the level of carbon monoxide, view the status of the vehicle and turn off the vehicle remotely using the Android app. The app can send an alert notification when the level of carbon monoxide is dangerous. Besides, the system able to send text messages to an emergency contact and the user using the GSM module if the level of carbon monoxide is dangerous and the user does not turn off the vehicle in a given time. The system can turn off the vehicle automatically using the relay when the level of carbon monoxide is very dangerous. The system stops running after the vehicle is turned off to avoid draining the battery.

The constraints of the project are the time for developing this project, knowledge of the related information and experience in this system development is limited. Poor network coverage or network congestion can cause some of the data transfer to fail as the system is network dependent. The system also depends on the specific ThingSpeak channel and API keys to store data collected for writing and reading. The data should be stored securely in a private cloud to protect the data and system from unauthorized use.

For future work, the features of the current system should be enhanced, and the system should have more advanced functionality and features. The user should be able to log in using their account to use the application privately. Artificial intelligence can be incorporated into the IoT system to analyze the data collected, identify the patterns, and detect anomalies in the data to make greater accuracy predictions and better results. The system also should be able to track and show the location of the vehicle so that help can be provided as soon as possible. The system should be more user-friendly and have lower power consumption.

ACKNOWLEDGEMENT

We would like to thank Universiti Malaysia Pahang for supporting this work under the RDU Grant, Grant number: RDU190373. We also appreciate the Ministry of Education Malaysia for supporting this work under the FRGS-RACER Grant, Grant Number: RACER/1/2019/ICT02/UMP//4 (RDU192620).

REFERENCES

1. Nandi, C., R. Debnath, and P. Debroy, Intelligent Control Systems for Carbon Monoxide Detection in

IoT Environments, in Guide to Ambient Intelligence in the IoT Environment. 2019, Springer. p. 153-176.
2. Mohamed, N. Carbon monoxide poisoning. 2020; Available from: http://www.myhealth.gov.my/en/carbon-monoxide-poisoning/.
3. Raub, J.A., et al., Carbon monoxide poisoning—a public health perspective. Toxicology, 2000. **145**(1): p. 1-14.
4. A Guide To Air Pollutant lndex ln Malaysia. [cited 2020; Available from: https://www.doe.gov.my/portalv1/wp-content/uploads/2015/09/API-Guideline.pdf.
5. Indoor Air Quality. [cited 2020; Available from: http://www.dosh.gov.my/index.php/en/chemical-management/indoor-air-quality.
6. Atzori, L., A. Iera, and G. Morabito, The internet of things: A survey. Computer networks, 2010. **54**(15): p. 2787-2805.
7. Tan, L. and N. Wang. Future internet: The internet of things. in 2010 3rd international conference on advanced computer theory and engineering (ICACTE). 2010. IEEE.
8. Kumazawa, T., et al., A curious autopsy case of accidental carbon monoxide poisoning in a motor vehicle. Legal Medicine, 2000. **2**(3): p. 181-185.
9. Board, T.R. and N.R. Council, The Ongoing Challenge of Managing Carbon Monoxide Pollution in Fairbanks, Alaska: Interim Report. 2002, Washington, DC: The National Academies Press. 154.
10. Panahi, P. and C. Bayılmış. Car indoor gas detection system. in 2017 International Conference on Computer Science and Engineering (UBMK). 2017. IEEE.
11. Sudila, N., et al. A IoT Proactive Disaster Management System for Mines. in International Symposium on Advanced Intelligent Systems. 2017.
12. Jaladi, A.R., et al., Environmental monitoring using wireless sensor networks (WSN) based on IOT. Int. Res. J. Eng. Technol, 2017. **4**(1): p. 1371-1378.
13. Abhimane, A., et al. IOT based vehicle traffic congestion control and monitoring system. in 2017 2nd International Conference for Convergence in Technology (I2CT). 2017. IEEE.
14. Ramya, V. and B. Palaniappan, Embedded Technology for vehicle cabin safety Monitoring and Alerting System. International Journal of Computer Science, Engineering and Applications, 2012. **2**(2): p. 83.
15. Ramya, V., et al., Embedded system for vehicle cabin toxic gas detection and alerting. Procedia Engineering, 2012. **30**: p. 869-873.
16. Tragos, E.Z., et al. Enabling reliable and secure IoT-based smart city applications. in 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS). 2014. IEEE.
17. Gebreslassie, B., A. Zayegh, and A. Kalam. Design, modeling of an intelligent green building using, actuator sensor interface network protocol. in 2017 Australasian Universities Power Engineering Conference (AUPEC). 2017. IEEE.
18. Wu, Q., et al., Cognitive internet of things: a new paradigm beyond connection. IEEE Internet of Things Journal, 2014. **1**(2): p. 129-143.
19. Khan, J.Y. and M.R. Yuce, Internet of Things (IoT): Systems and Applications. 2019: CRC Press.
20. Paruchuri, V.L. and P. Rajesh, IoT for monitoring carbon monoxide (CO) emissions using wireless sensor networks in smart cities. Int. J. Eng. Technol, 2018. **7**(2): p. 1045-1050.
21. Gupta, K. and N. Rakesh. IoT based automobile air pollution monitoring system. in 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence). 2018. IEEE.
22. Kodali, R.K., S.C. Rajanarayanan, and L. Boppana. IoT Based Vehicular Air Quality Monitoring System. in 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129). 2019. IEEE.