International Journal of Advance Research in Science and Engineering
Volume No. 14, Issue No. 01, January 2025
www.ijarse.com
IJARSE
ISSN 2319 - 8354

NODEMCU-BASED IOT EMISSION CONTROL SYSTEM FOR VEHICLES

Akilandeswari R[1], R Hemalatha[2], Dr S SathyaDeepa[3]

[1]Assistant Professor, [2]Associate Professor, Department of Electronics, Sindhi College, Bangalore.

[3]Assistant Professor, Department of Electronics, Hindusthan College of Arts & Science,Coimbatore

ABSTRACT

The rise in air pollution levels due to vehicular emissions has become a significant concern worldwide. In response, there is a growing need for a reliable and efficient emissions monitoring system that can be installed in vehicles to help reduce their carbon footprint. This paper proposes an IOT-based emission monitoring system that uses a Node MCU, MQ2 and MQ7 sensors, an LCD display, and a buzzer to monitor the level of harmful gases emitted by vehicles. The proposed system provides real-time monitoring of the level of carbon monoxide, methane, and other harmful gases emitted by a vehicle. The sensors used in the system are highly sensitive and can detect even the slightest change in the emission levels. The Node MCU used in the system is responsible for collecting and transmitting data to the cloud. The data can then be accessed by relevant authorities for analysis and appropriate action. The LCD display and buzzer used in the system provide immediate feedback to the driver about the emission levels of their vehicle. The display shows the real-time levels of the detected gases, while the buzzer sounds an alarm when the levels exceed safe limits. This feature helps the driver to take corrective measures and reduce their vehicle's emissions. The proposed IoT-based emission monitoring system provides a reliable and efficient solution for monitoring vehicular emissions. The system's ability to provide real-time data and immediate feedback can help reduce air pollution levels and promote a cleaner and healthier environment.

Keywords: *Node MCU, Harmful gases, LCD display.*

INTRODUCTION

The rapid increase in the number of vehicles on the road has led to a significant rise in air pollution levels. Vehicular emissions are a significant contributor to air pollution, and they pose a severe threat to the environment and human health [1]. Governments and organizations worldwide are taking steps to reduce vehicular emissions and promote cleaner transportation options. However, a reliable and efficient emissions monitoring system is crucial to effectively reduce emissions and promote a cleaner environment.

This paper proposes an IoT-based emission monitoring system that can be installed in vehicles to monitor their emissions in real-time. The system uses a Node MCU, MQ2 and MQ7 sensors, an LCD display, and a buzzer to monitor the level of harmful gases emitted by the vehicle [2]. The Node MCU is a low-cost microcontroller that is responsible for collecting and transmitting data to the cloud.

International Journal of Advance Research in Science and Engineering
Volume No. 14, Issue No. 01, January 2025
www.ijarse.com
IJARSE
ISSN 2319 - 8354

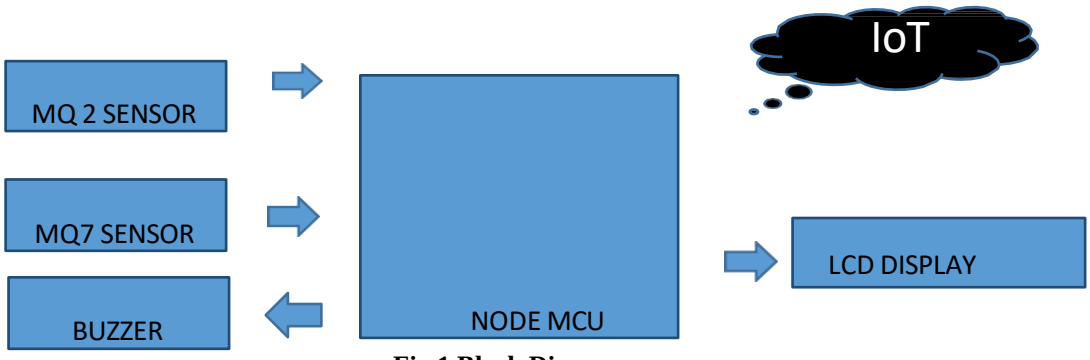

Fig.1 Block Diagram

The Node MCU is connected to the MQ2 and MQ7 sensors, which are highly sensitive and can detect even the slightest change in the emission levels[3]. The MQ2 sensor is used to detect gases such as carbon monoxide, propane, and methane, while the MQ7 sensor is used to detect carbon monoxide[4]. The data collected by the sensors is transmitted to the cloud through Wi-Fi or GSM connectivity [5]. The data can then be accessed by relevant authorities for analysis and appropriate action [6]. The cloud platform used can be customized according to the specific needs of the user [7]. The LCD display and buzzer are used to provide immediate feedback to the driver about the emission levels of their vehicle. The display shows the real-time levels of the detected gases, while the buzzer sounds an alarm when the levels exceed safe limits [8]. This feature helps the driver to take corrective measures and reduce their vehicle's emissions [9-10]. The proposed IoT-based emission monitoring system provides a reliable and efficient solution for monitoring vehicular emissions. The system's ability to provide real-time data and immediate feedback can help reduce air pollution levels and promote a cleaner and healthier environment. The system can be customized according to the specific needs of the user and can be integrated into any vehicle with ease.

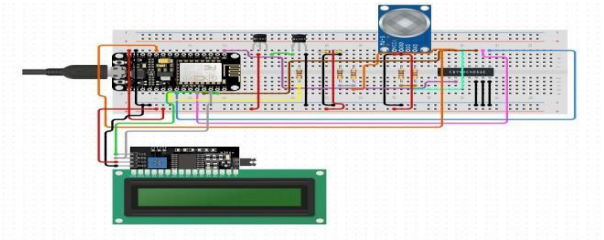

Fig.2 Circuit Diagram

HARDWARE COMPONENTS NODE MCU ESP82266:

The ESP8266 is a self- contained WiFi networking result immolation as a ground from being micro controller to WiFi and is also able of running self-contained operations. This module comes with USB connector and arrangement of pin outs. We can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino with micro USB cable. It's also breadboard friendly.

Fig.3 NODE MCU ESP82266

NodeMcu is a small board, based on ESP8266, 4ntegrates GPIO, PWM, IIC, 1- line and ADC each in one board. It's a Lua grounded firmware for WiFi- SOC(Systems On- Chop) ESP8266 WiFi module.

The NodeMCU V3 is an open-source firmware and development kit designed to help create IoT products with just a few lines of Lua script.

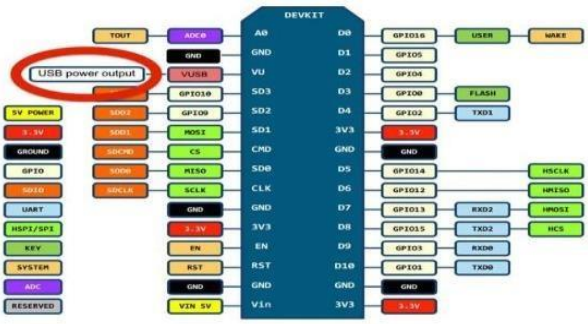

Fig.4 Block diagram of NODE MCU ESP82266

- The board features multiple GPIO pins, enabling connections with various peripherals and supporting PWM, I2C, SPI, and UART serial communications.
- The module's interface consists of two main parts: the Firmware, which operates on the ESP8266 Wi-Fi SoC, and the Hardware, which is based on the ESP-12 module.
- The firmware uses Lua, an easy-to-learn scripting language that provides a simple programming environment and access to a broad developer community.
- As open-source firmware, it offers the flexibility to edit, modify, and rebuild the module, allowing for continuous optimization to meet specific requirements.
- A USB to UART converter is integrated into the module, enabling USB data to be converted into UART data for serial communication.
- Instead of a standard USB port, the module uses a Micro USB port for both programming and powering the board.
- The board features a status LED that blinks and turns off, providing feedback on the module's status when connected to a computer.
- The module's seamless Wi-Fi connectivity between two channels makes it an excellent choice for integration with other embedded devices like Raspberry Pi.

GAS SENSORS

Gas sensors, or gas detectors, are electronic devices that detect and identify various gases. They are primarily used to monitor toxic or explosive gases and measure their concentration. These sensors are commonly found in

factories and manufacturing plants to detect gas leaks, as well as in homes to monitor smoke and carbon monoxide levels. Gas sensors vary in size, ranging from portable to fixed units, and differ in their range and sensitivity. Often, they are integrated into larger systems, such as hazmat and security systems, and are usually connected to alarms or interfaces for alerting purposes. Due to their continuous exposure to air and gases, gas sensors require more frequent calibration compared to many other types of sensors.

The physical construction and sensing mechanisms of gas sensors can differ greatly depending on their intended use and environment. One of the most widely used types for detecting toxic gases and smoke is the metal oxide-based sensor. This sensor employs a chemiresistor that reacts with target gases, causing an increase in electrical resistance when exposed to gases like carbon monoxide, hydrogen, methane, and butane. Most home based smoke detection systems are oxide based sensors

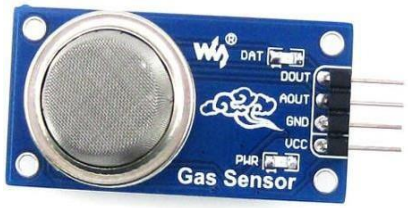

Fig.5 Gas sensor

PIN CONFIGURATION

Pin No:	Pin Name:	Description
For Module		
1	Vcc	This pin powers the module, typically the operating voltage is +5V
2	Ground	Used to connect the module to system ground
3	Digital Out	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer
4	Analog Out	This pin outputs 0-5V analog voltage based on the intensity of the gas
For Sensor		
1	H -Pins	Out of the two H pins, one pin is connected to supply and the other to ground
2	A-Pins	The A pins and B pins are interchangeable. These pins will be tied to the Supply voltage.
3	B-Pins	The A pins and B pins are interchangeable. One pin will act as output while the other will be pulled to ground.

APPLICATIONS

- Detects or measure Gases like LPG, Alcohol, Propane, Hydrogen, CO and even methane

- Air quality monitor
- Gas leak alarm
- Safety standard maintenance
- Maintaining environment standards in hospitals

Carbon-monoxide (MQ 7) Sensor

Fig.6 MQ-7 CO Carbon Monoxide Coal Gas Sensor

The MQ-7 gas sensor has SnO2 as its sensitive material, which has lower conductivity in clean air. The sensor detects gases by alternating between high and low temperatures, detecting CO at low temperatures (when heated by 1.5V). As the gas concentration increases, the sensor's conductivity rises. At high temperatures (when heated by 5.0V), the sensor cleans off any other gases that were adsorbed at lower temperatures. Using a simple electronic circuit, the change in conductivity can be converted into an output signal that corresponds to the gas concentration. The MQ-7 sensor is highly sensitive to carbon monoxide and can detect various gases containing CO. It is low-cost and suitable for a range of applications**.**

IoT Working

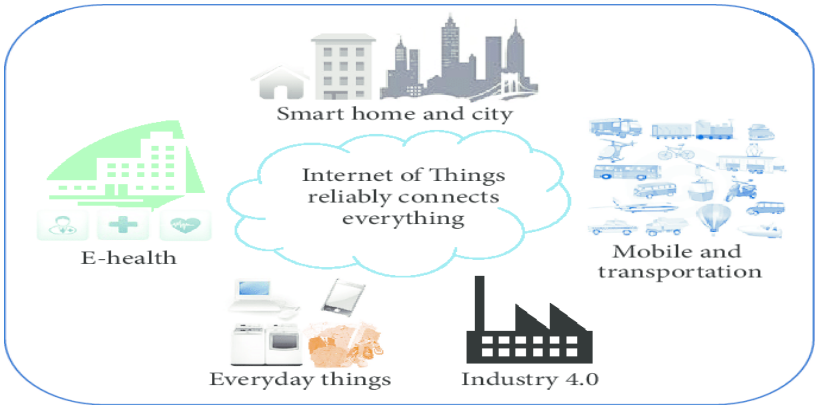

Fig.7 Functional diagram of IoT

An IoT ecosystem is made up of smart devices that are web-enabled and utilize embedded systems, including processors, sensors, and communication hardware, to collect, transmit, and act upon the data they obtain from their environment. These devices share the data they collect by connecting to an IoT gateway or another edge device, where the data is either sent to the cloud for analysis or processed locally. In some instances, devices communicate with other related devices and take action based on the information they receive. While these devices usually operate autonomously, users can interact with them for tasks such as setup, instruction input, or data retrieval.

ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is an official software developed by Arduino.cc, mainly used for writing, compiling, and uploading code to Arduino devices. Most Arduino modules are compatible with this open-source software, which is simple to install and allows users to quickly compile their code. This article will introduce the software, explain how to install it, and show how to set it up for developing applications using Arduino modules.

- Arduino IDE is an open-source software designed for writing and compiling code for Arduino modules.
- As the official Arduino software, it simplifies the code compilation process, making it user-friendly even for beginners without technical expertise.
- The IDE is available for operating systems like macOS, Windows, and Linux, and it runs on the Java platform. It comes with built-in functions and commands that are useful for debugging, editing, and compiling code.
- Several Arduino modules are available, such as Arduino Uno, Arduino Mega, Arduino Leonardo, and Arduino Micro.
- Each module contains a microcontroller that is programmed and receives information in the form of code.
- The primary code, called a "sketch," is created in the IDE and generates a Hex File, which is then uploaded to the microcontroller on the board.
- The IDE consists of two main parts: the Editor, used to write the code, and the Compiler, which compiles and uploads the code to the Arduino module.
- The environment supports both C and C++ programming languages.

HOW TO GET ARDUINO IDE

The software can be downloaded from the official Arduino website. As previously mentioned, it is available for popular operating systems like Linux, Windows, and macOS. You should choose the version that best matches your operating system for easy compatibility.

IDE OVERVIEW:

The IDE environment is mainly divided into three sections:

- **Menu Bar**
- **Text Editor**
- **Output Pane**

Text Editor - Output pane

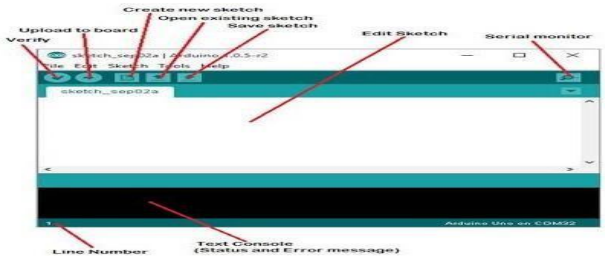

Fig.8 Functional diagram of Text editor

The bar appearing on the top is called Menu Bar that comes with five different options as follow:

- **File** – This option allows you to open a new window for writing code or open an existing file. The table below shows the various subsections under the File option:

1. Menu Bar
2. Text Editor
3. Output Pane

- **Edit** – This section is used for copying, pasting, and modifying code, as well as changing the font.
- **Sketch** – This option is used for compiling and programming your code.
- **Tools** – Mainly used for testing projects. The Programmer section in this menu is where you can burn a bootloader onto a new microcontroller.
- **Help** – If you have any doubts or need assistance with the software, a full range of help options is available, from getting started to troubleshooting. The six buttons under the Menu tab are linked to the running program and function as follows:
- The six buttons under the Menu tab are linked to the running program. A separate pop-up window will appear as an independent terminal, which plays a key role in sending and receiving serial data. You can also access the Serial Monitor by going to the Tools menu or by pressing **Ctrl+Shift+M** to open it directly.
- The Serial Monitor is useful for debugging your written sketches, as it helps you understand how your program is running. To activate the Serial Monitor, ensure that your Arduino module is connected to your computer via USB cable.
- You need to select the baud rate that matches your Arduino board in use.

Program Structure

- **Declarations**
- **Variables**

When using Arduino, you need to declare global variables and instances that will be used later. Essentially, a variable allows you to store and name a value for future use. For example, data from a sensor can be stored in a variable to be accessed later. To declare a variable, you define its type, name, and initial value. While declaring global variables is not strictly necessary, it is highly recommended as it makes it easier to use the stored values later in your program.

INSTANCES

In software programming, a class is a collection of functions and variables that are grouped together. Each class contains a special function called a constructor, which is used to create an instance of the class. To use the functions of the class, an instance must be declared.

Setup()

Every Arduino sketch must include a setup function. This function sets the initial state of the Arduino when it boots up and runs only once. In this function, you typically define the following:

1. Pin functionality using the **pinMode** function
2. The initial state of pins
3. Initialization of classes

4. Initialization of variables
5. Code logic

Loop()

The **loop()** function is also essential in every Arduino sketch. After the **setup()** function completes, the **loop()** function runs repeatedly in a continuous cycle. It contains the main logic of the circuit and is where the core operations are carried out.

CONCLUSION

In conclusion, the proposed IoT-based emission monitoring system offers an efficient and effective solution for monitoring vehicular emissions in real-time. The system uses a Node MCU, MQ2 and MQ7 sensors, an LCD display, and a buzzer to detect and track harmful gases emitted by vehicles. The system provides immediate feedback to the driver via the LCD display and buzzer, encouraging responsible driving and helping reduce vehicular emissions. The data collected can be accessed by relevant authorities for analysis and action, supporting the development of strategies to mitigate air pollution. This system is cost-effective, easy to install, and customizable to meet specific user needs. With IoT technology, the system can connect to the cloud, allowing real-time data access from anywhere at any time. Overall, the proposed IoT-based emission monitoring system can play a significant role in reducing vehicular emissions and promoting a cleaner, healthier environment by providing continuous monitoring of air pollution levels.

REFERENCES

[1]. Chouhan, A., Gahlaut, R., & Sharma, V. (2019). IoT Based Air Pollution Monitoring System. International Journal of Innovative Technology and Exploring Engineering, 8(8S2), 6-10.

[2]. Kumar, N., & Yadav, A. (2020). Design of IoT Based Air Pollution Monitoring System. International Journal of Control and Automation, 13(1), 100-110.

[3]. Senthil Kumar, S., Gokulakrishnan, R., & Nithiyanandam, P. (2018). IoT Based Air Pollution Monitoring System Using Node MCU. International Journal of Innovative Research in Science, Engineering and Technology, 7(11), 16090-16095.

[4]. Nishanth, N., Ravi, P., Santhosh Kumar, S., & Sathish Kumar, R. (2020). IoT Based Air Pollution Monitoring System Using Raspberry Pi. In 2020 IEEE 7th International Conference on Computing Communication and Automation (ICCCA) (pp. 319-324). IEEE.

[5]. Singh, A., Kumar, A., & Verma, M. (2020). A Review on IoT Based Air Pollution Monitoring System. In 2020 2nd International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1090-1095). IEEE.

[6]. Jangamshetti, S. R., Dikonda, D. D., & Bhat, S. S. (2019). IoT Based Air Pollution Monitoring System. In 2019 IEEE 3rd International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-5). IEEE.

[7]. Majhi, S., & Rout, B. K. (2019). An IoT-Based Air Pollution Monitoring System. In 2019 International Conference on Smart Electronics and Communication (ICOSEC) (pp. 68-71). IEEE.

[8]. Guntur, D., R, M. P., & Chandra Sekhar, G. (2019). An IoT Based Air Pollution Monitoring System. In 2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 544-547). IEEE.

[9]. Srivastava, S., Singh, S., Singh, A., & Sinha, A. (2019). IoT Based Air Pollution Monitoring System. In 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 234-238). IEEE.

[10]. Zhang, X., Dong, J., & Jin, M. (2020). Air Pollution Monitoring and Management System Based on IoT Technology. In Proceedings of the 2nd International Conference on Big Data and Internet of Things (pp. 277-285). Springer.