International Research Journal of Engineering Science, Technology and Innovation Vol. 11(1)
pp.1-7, January, 2025
Available online http://www.interesjournals.org/IRJESTI

Short Communication

Environment Pollution Monitoring System Using IoT

T Sravanthi, B Raja Vamshi, B Thanvi Sushrutha*, B Hitham Raj, D Anitha

Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Hyderabad, India

*Corresponding Author's E-mail: thanvib6@gmail.com

Received: 09-October-2023; Manuscript No: irjesti-24-116176; **Editor assigned:** 12-October-2023; Pre-QC No: irjesti-24-116176 (PQ); **Reviewed:** 26-October-2023; QC No: irjesti-24-116176; **Revised:** 10-January-2025; Manuscript No: irjesti-24-116176 (R); **Published:** 17-January-2025, DOI: 10.14303/2315-5663.2025.120

Abstract

The Internet of Things (IoT) represents a global network of intelligent devices capable of sensing their environment and engaging with users and other interconnected systems. In our current era, one of the paramount concerns is the escalating issue of worldwide air pollution. This pollution has surged over time due to various factors, including population growth, increased vehicular activity, industrial expansion, and urbanization, culminating in detrimental consequences for human well-being. These detrimental effects are particularly pronounced among populations directly exposed to the pollution. Air quality deteriorates as the concentration of harmful gases such as carbon dioxide, smoke, benzene, Ammonia (NH_3), and Nitrogen Dioxide (NO_2) in the atmosphere reaches critical levels.

Keywords: Internet of Things (IoT), Population growth, Carbon dioxide, Global network, Intelligent devices

INTRODUCTION

Air, an indispensable element in our environment, comprises a mixture of gases, including Nitrogen, Oxygen, Carbon Monoxide, and trace elements (Matthews VO et al., 2018). The presence of clean, uncontaminated air is vital for human and life. Any alteration in the natural composition of the atmosphere can have severe consequences for all living organisms on Earth. Air pollution is characterized by the presence of various contaminants in the air, typically in quantities that can be detrimental to humans, animals, and plants (Arun Raj V et al., 2017). These contaminants are quantified using units such as Parts per Million (ppm). Primary pollutants are substances released directly into the atmosphere, while secondary pollutants are generated through chemical reactions involving primary pollutants and other atmospheric compounds (Wei-Ying Yi et al., 2016). The quality of air has impact on public health. The impacts of air pollution encompass a wide spectrum of health issues, including difficulties in breathing, persistent coughing, and the exacerbation of conditions like asthma and emphysema. Additionally, polluted air can severely limit visibility (Sani AB et al., 2019). Shockingly, air pollution is responsible for an alarming 7 million premature deaths globally each year, equating to one in every eight premature fatalities annually. Particularly alarming is the fact that nearly 570,000 children under the age of five succumb to respiratory infections associated with indoor and outdoor pollution as well as exposure to second-hand smoke. Children exposed to air pollution face an elevated risk of developing chronic respiratory ailments like asthma (Ullo SL et al., 2020). In response to the pressing issue of air pollution, researchers worldwide have developed models to monitor various polluting gases, such as Sulphur Dioxide (SO_2), Carbon Monoxide (CO_2), Nitrogen Oxides (NO), and more (Asha P et al., 2022). This paper primarily centers on the creation and implementation of an intelligent air pollutant monitoring system. It delves into the methodology of utilizing a gas sensor, ESP-32 and DHT11 to gauge pollutant levels in air (Cynthia J et al., 2019). The primary objective of this research is to devise a smart air pollution monitoring system

 ISSN: 2315-5663

capable of continuously observing, analyzing, and transmitting real-time air quality data to a remote server *via* the internet, thus ensuring the data remains current and accessible for informed decision-making (Idrees Z et al., 2018).

DESCRIPTION

System design

An IoT-based environmental pollution monitoring system utilizes interconnected components to gather and process data for real-time pollution tracking. In this system, the ESP32 microcontroller serves as the central control unit. It interfaces with the MQ-2 gas sensor and the DHT11 temperature and humidity sensor. These components are represented as blocks in the block diagram. The ESP32 reads data from the MQ-2 sensor to detect various gases like carbon monoxide, methane, and smoke, while the DHT11 measures temperature and humidity levels. These readings are transmitted *via* the internet to a central server for data analysis and visualization. Arrows in the diagram show the flow of data from the sensors to the ESP32 and then to the server, depicting the functional relationships and information flow within the system (**Figure 1**).

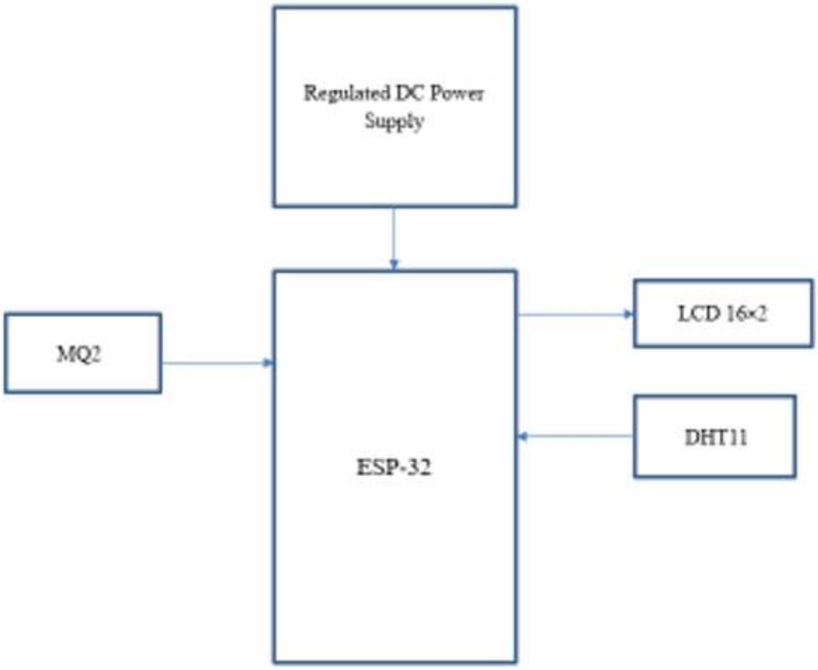

Figure 1. Block diagram.

Power supply

ESP-32: Esp-32 is known as system on chip microcontroller. It is a powerful and flexible microtroller which has gained more popularity recently in makers and IoT communities due to its feature and affordability. It features a dual core processor which allows it to handle tasks efficiently. It has built in Wi-Fi and bluetooth capabilities, making it easy to connect to the internet and communicate with the other devices wirelessly. It consists of 30 pins, which include GPIO pins, Power pins, analog pins, UART, SPI, I2C, PWM and control pins.

Gas sensor: The gas sensor used is of variant MQ-2. It is sensitive to gases like carbon monoxide, methane, propane, sulphur dioxide and other combustible gases including smoke. It is a vital component in the field of environmental monitoring and safety applications. MQ-2 relies on a heated Metal Oxide Semiconductor (MOS) to measure changes in resistance when exposed to target gases. MQ-2 sensor features a 4 pin configuration including VCC (Power Supply), Analog Out (AO), Digital Out (DO), Ground (GND).

DHT11: The DHT11 sensor, a digital temperature and humidity sensor, has emerged as a corner stone in the domain of IoT enabled environmental monitoring. It operates based on a capacitive humidity sensing element and a calibrated digital signal processing chip. When exposed to environment, the sensor’s humidity-sensitive capacitor undergoes changes in capacitance due to variations in humidity levels. The sensor’s internal processor converts these changes into digital temperature and humidity readings, which are interfaced with microcontrollers. DHT-11 sensor features a 4 pin configuration including VCC (Power Supply), Data Out, Not Connected (NC), Ground (GND).

LCD: Ground 16×2 LCD displays are based on liquid crystal technology. Each character position consists of a matrix of

 ISSN: 2315-5663

pixels, and liquid crystal cells control the passage of light through these pixels. By selectively activating and deactivating these cells, characters and symbols are displayed on the screen. Microcontrollers send data and commands to the LCD to control what is displayed (**Figure 2**).

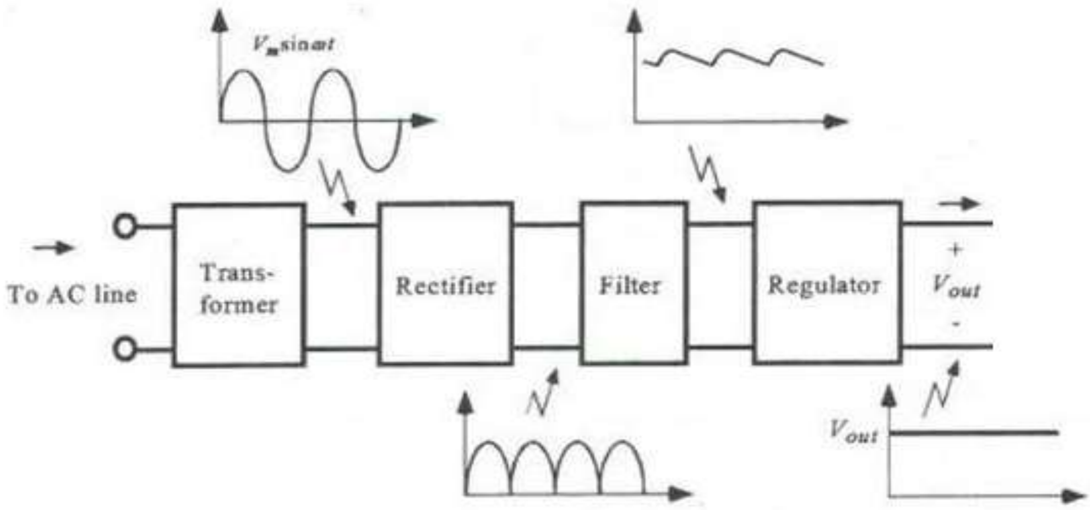

Figure 2. Power supply.

Design steps and process

Pin connections

- Connect AO of MQ2 to D22 of ESP-32.
- Connect DO of MQ2 to DO of ESP-32.
- Connect VCC of MQ2 to 5V of PCB board.
- Connect GND of MQ2 to Ground.
- Connect VCC of DHT11 to 5V of PCB board.
- Connect DO of DHT11 to D23 of ESP-32.
- Connectt GND of DHT11 to Ground (**Figure 3**).

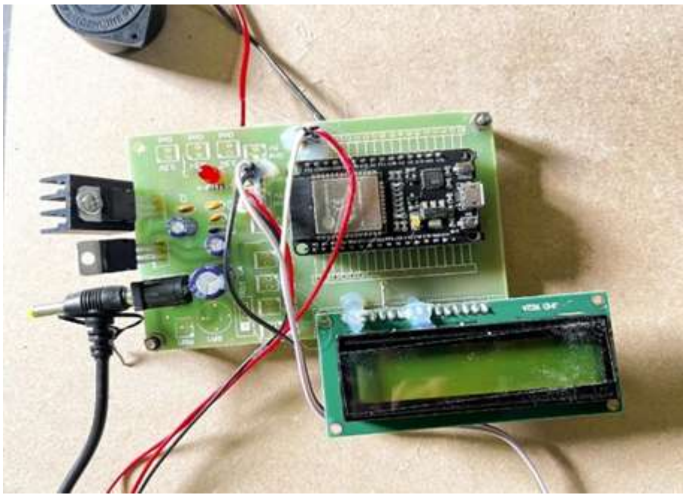

Figure 3. System design.

Steps to install Arduino IDE

- Visit Arduino website at https://www.arduino.cc/e n/software.
- Download the Arduino ide for your operating system(windows,linux,MacOs).
- Install the downloaded software by following on-screen instructions.
- Once installed open the Arduino Ide.

Steps to install ESP-32 board in Arduino IDE

- Open Arduino IDE.
- Go to “file”>“preferences”.

ISSN: 2315-5663

- In the “additional boards Manager” URLs field, enter the following URL:https://dl.espressif.com/dl/package_esp32_index.json.
- Click “Ok” to close preference window.
- Go to “tools”> “board”> “board manager”.
- In the boards Manager type ESP-32 in the search bar.
- Click on “install” for “esp-32” platform by Espressif systems.
- Once installation is completed, close the boards Manager.
- Now you can select esp32 from “Tools”> “board” menu.
- Choose specific variant module you are using from the same menu.
- After successfully installing esp32 board to Arduino IDE we can start developing and uploading our code.

Code logic

The code begins by including various libraries for components such as the liquid crystal display, Wi-Fi, HTTP Client, DHT11 temperature and humidity sensor, and more.

It sets up the Wi-Fi connection using the provided SSID and password to connect to a local network.

Variable declarations: It declares variables for HTTP communication, sensor data, pins for various components (like LCD, buzzer, and smoke sensor), and strings to build the HTTP request payload.

Setup Function (setup):

- Initializes serial communication.
- Sets up pins for the smoke sensor and buzzer.
- Initializes the LCD display and shows an initial message.
- Connects to the Wi-Fi network.

Main loop (loop):

- Reads temperature and humidity data from a DHT11 sensor.
- Checks if the temperature or humidity values are above certain thresholds. If so, it triggers a buzzer alarm and sends data to a remote server using an HTTP GET request.
- Checks the state of a smoke sensor. If smoke is detected, it triggers the buzzer alarm and sends data to the server.
- It sends data to the server periodically.

Custom functions (iot_send, beep, okcheck, serialEvent, gpsEvent, get_gps, coordinate2dec, gps_convert, convlat, convlong, converts, convertl):

- **Iot_send:** Sends sensor data to a remote server using HTTP GET requests.
- **Beep:** Generates a beep sound using the buzzer.
- **Okcheck:** Waits for the character ‘K’ from the serial input.
- **Serialevent:** Reads data from the serial port.
- **Gpsevent:** Reads data from a GPS module.
- **Get_gps:** Retrieves GPS data from the GPS module.
- **Coordinate2dec:** Converts GPS coordinates from degrees, minutes, and seconds to decimal format.
- **Gps_convert:** Parses and extracts GPS data from a GPS string.
- **Convlat and convlong:** Converts latitude and longitude values to a specific format.
- **Converts and convertl:** Converts unsigned integer values to character format for display.

OBSERVATIONS

The image below shows the experimental setup of our project which displays output on LCD (**Figure 4**).

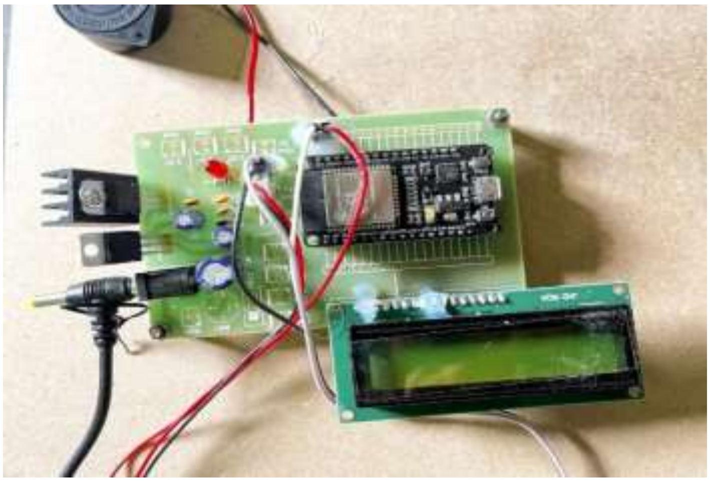

Figure 4. Output on LCD.

 ISSN: 2315-5663

Temperature is displayed on LCD. It produces Buzzer indication when reached to 40 (**Figure 5**).

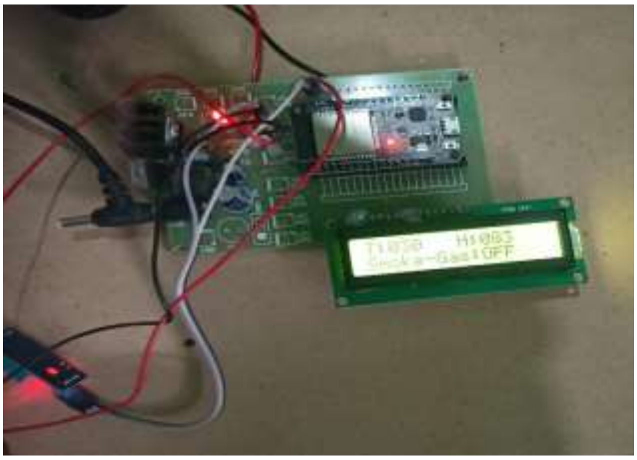

Figure 5. Temperature is displayed on LCD.

Humidity is displayed on LCD. It produces Buzzer indication when reached to 86 (**Figure 6**).

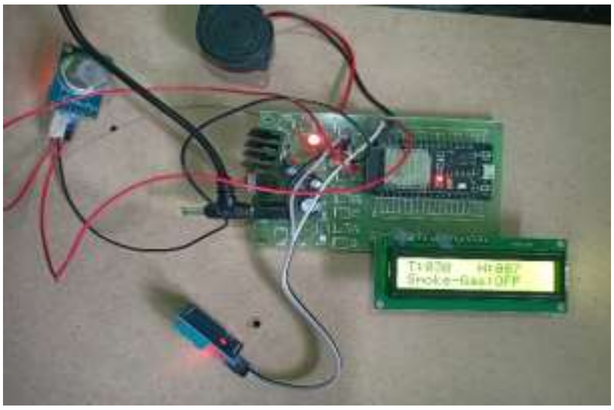

Figure 6. Humidity is displayed on LCD.

Initially Smoke-Gas is OFF, which indicates there is no detected gas. When MQ-2 is exposed to smoke of a burning paper the produced carbon dioxide is detected by sensor and displays ON (**Figures 7 and 8**).

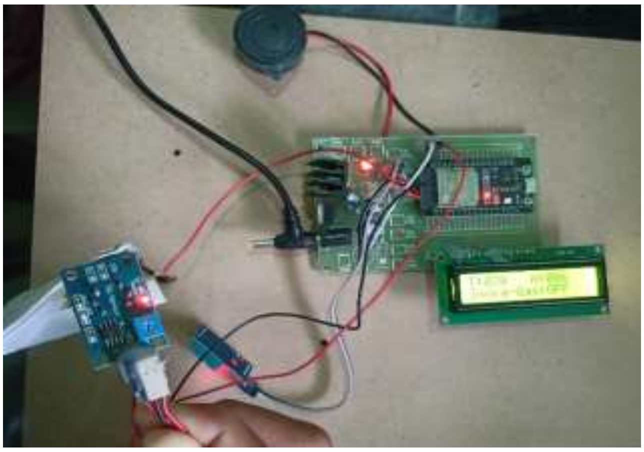

Figure 7. Smoke gas is OFF.

 ISSN: 2315-5663

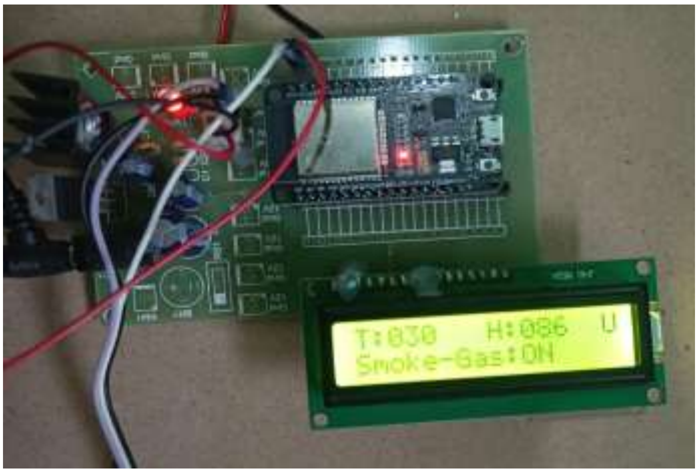

Figure 8. Smoke-Gas is On.

Go to http://projectsfactoryserver.in/index.php and login to see observations (**Figures 9 and 10**).

Figure 9. Login page.

S.No	Temperature	Humidity	Smoke-Gas	Date
1	28.90	87.00	OFF	2023-09-08 21:12:54
2	28.90	88.00	OFF	2023-09-08 21:12:46
3	28.90	88.00	OFF	2023-09-08 21:12:39
4	29.10	88.00	OFF	2023-09-08 21:12:31
5	29.30	88.00	OFF	2023-09-08 21:12:24
6	29.30	88.00	OFF	2023-09-08 21:12:16
7	29.80	88.00	OFF	2023-09-08 21:12:09
8	29.80	89.00	OFF	2023-09-08 21:12:01
9	30.20	88.00	OFF	2023-09-08 21:11:54
10	30.20	88.00	OFF	2023-09-08 21:11:46
11	30.80	88.00	OFF	2023-09-08 21:11:39
12	31.30	88.00	OFF	2023-09-08 21:11:31
13	31.80	89.00	OFF	2023-09-08 21:11:24
14	29.80	88.00	OFF	2023-09-08 21:11:16
15	29.30	89.00	OFF	2023-09-08 21:00:13
16	29.80	89.00	OFF	2023-09-08 21:00:05
17	29.80	90.00	OFF	2023-09-08 20:59:58
18	29.80	90.00	OFF	2023-09-08 20:59:50
19	29.80	90.00	OFF	2023-09-08 20:59:43
20	29.80	90.00	OFF	2023-09-08 20:59:35
21	29.80	91.00	OFF	2023-09-08 20:59:28
22	30.20	90.00	OFF	2023-09-08 20:59:20
23	30.20	90.00	OFF	2023-09-08 20:59:12
24	30.20	90.00	OFF	2023-09-08 20:59:05
25	30.80	90.00	OFF	2023-09-08 20:58:58
26	31.10	90.00	OFF	2023-09-08 20:58:50
27	31.30	90.00	OFF	2023-09-08 20:58:43
28	31.80	90.00	OFF	2023-09-08 20:58:35
29	31.40	90.00	OFF	2023-09-08 20:58:28
30	30.80	90.00	OFF	2023-09-08 20:58:20
31	30.80	90.00	OFF	2023-09-08 20:58:13
32	31.30	90.00	OFF	2023-09-08 20:58:05
33	32.30	90.00	OFF	2023-09-08 20:57:57
34	32.80	90.00	OFF	2023-09-08 20:57:50
35	33.40	89.00	OFF	2023-09-08 20:57:43
36	29.50	89.00	OFF	2023-09-08 20:57:35
37	28.90	88.00	OFF	2023-09-08 20:57:27
38	35.20	63.00	ON	2023-09-02 11:55:22
39	40.10	62.00	OFF	2023-09-02 11:52:44
40	34.70	81.00	OFF	2023-09-02 11:51:51

Figure 10. Observation values of temperature, humidity, smoke gas with date.

 ISSN: 2315-5663

CONCLUSION

We have successfully designed and implemented an IoT-based environment pollution monitoring system using our experimental kit. Through a series of tests and data collection, we have obtained valuable insights into the environmental conditions of our target area. This not only enhances our understanding of environmental conditions but also enables us to take proactive measures in addressing pollution issues.

REFERENCES

1. Matthews VO, Uzairue SI, Noma-Osaghae E, Nwukor F (2018). Design and simulation of a smart automated traffic system in a campus community. Int J Emerg Technol Innov. 5: 492-497.
2. Wei-Ying Yi, Kwong-Sak Leung, Yee Leung, Mei-Ling Meng, Terrence Mak (2016). Modular Sensor System (MSS) for Urban Air Pollution Monitoring. IEEE.
3. Sani AB, Beauty PE (2019). Smart framework for environmental pollution monitoring and control system using IoT-based technology. Sens Transducers. 229: 84-93.
4. Ullo SL, Sinha GR (2020). Advances in smart environment monitoring systems using IoT and sensors. Sens. 20: 3113.
5. Asha P, Natrayan LB, Geetha BT, Beulah JR, Sumathy R, et al (2022). IoT enabled environmental toxicology for air pollution monitoring using AI techniques. Environ Res. 205: 112574.
6. Cynthia J, Saroja MN, Sultana P, Senthil J (2019). IoT-based real time air pollution monitoring system. Int J Grid High Perform Comput. 11: 28-41.
7. Idrees Z, Zou Z, Zheng L (2018). Edge computing based IoT architecture for low cost air pollution monitoring systems: a comprehensive system analysis, design considerations and development. Sens. 18: 3021.