iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: III Month of publication: March 2026

DOI: https://doi.org/10.22214/ijraset.2026.79155

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

IoT-Based Real-Time Air Quality Monitoring System Using ESP32, MQ2, DHT11 and SH1106 OLED Display with Cloud Integration

Priyam Ashish Rolston[1], Raghav Gupta[2], Tushar Kumar[3], Yash Rastogi[4], Mrs. Meetu Mann[5]
Department of Computer Science and Engineering IoT, Meerut Institute of Engineering and Technology, Meerut, India

Abstract: Air pollution has become a key issue in today's urban settings as a result of industrial growth, car emissions, and population increase. We see that it is very important to have continuous reportage of environmental elements which in turn will improve awareness. We present in this paper our work which is the design and implementation of an IoT based real time air quality monitoring system which we have put together with ESP32. The system we have put together uses a DHT11 sensor for temperature and humidity measurement, an MQ2 gas sensor for identification of harmful gases, and an SH1006 OLED display for local data visualization. Also we see that we are able to transmit the collected data via WiFi to the ThingSpeak cloud platform for remote monitoring and analysis. Our put forth system is a low cost, efficient and real time solution for environmental monitoring. We report that in our experiments the system does very well in the capture and display of environmental parameters both locally and on the cloud.

***Index Terms*: IoT, ESP32, Air Quality Monitoring, MQ2 Sensor, DHT11, SH1106 OLED, ThingSpeak**

I. INTRODUCTION

Air pollution has become a very important issue in terms of environment and public health in urban and semi-urban areas. We see that with increase in industrialization, from car emissions to large scale construction and use of fossil fuels which have all played a role in degrading air quality. Also we see that a great deal of the time people are dealing with smog, reduced visibility, and respiratory issues. The World Health Organization reports that air pollution is a factor in millions of deaths every year which is why it is of the upmost importance we do continuous study of environmental elements.

Conventional air quality monitoring is done by government agencies which use large, stationary and high cost equipment. While they do provide accurate data, the scale of their deployment is limited. Pollution levels may very between close proximity areas such as a busy road and a residential block. Also we see a lack of local and real time data which in turn leaves gaps in our ability to react to environmental issues.

The rise of the Internet of Things (IoT) has seen the development of small scale, economic, and connected monitoring solutions. IoT based solutions report in real time, process data, and transmit it to the cloud which in turn makes environmental monitoring more accessible and scalable. Also we see the idea of distributed sensing which is supported by these systems in which many low cost devices together present a picture of air quality which is very detailed and which also extends to different locations.

In our work we present a development of an IoT based air quality monitoring system which we put together using the ESP32 microcontroller. We have included an MQ2 gas sensor for the detection of harmful gases and smoke and a DHT11 sensor for measurement of temperature and humidity. Also we used SH1006 OLED display for real time local presentation of sensor data and WiFi for transfer of data to the ThingSpeak cloud platform which in turn is used for remote monitoring and analysis. We aim in this paper to put forth the design, implementation and evaluation of a low cost, real time environmental monitoring solution which we have which includes local display and cloud based visualization.

II. PROBLEM STATEMENT

Air quality which we are still very much in the process of improving reports is a large issue brought out by what we have in terms of present infrastructure and access issues. While it is a fact that environmental monitoring is a base requirement for public and environmental health, at present our systems do not cover enough and also do not give out timely information at a local level. We note the main issues are:

1) Limited Spatial Coverage: Standard air quality monitoring networks are sparse, which also in that of inadequate reporting of local pollution levels within short distances we see much variation in air quality undetected.
2) High Implementation Cost: Classic monitoring systems that use very complex and expensive equipment making them unfit for wide scale deployment.
3) Lack of Real-Time Accessibility: Many present solutions provide delayed or averaged data which is insufficient for immediate decision-making related to health and safety.
4) Limited User Accessibility: Present day systems are often complex, requiring advanced technical knowledge or proprietary systems that also limit their use for students, researchers, and general users.

To solve these issues we see a need for a low cost, portable, user friendly system that is also able to monitor environmental conditions in real time. The proposed solution we put forth is a design of a base which is the IoT architecture around the ESP32 which also includes MQ2 and DHT11 sensors and a SH1106 OLED screen for what is to be displayed on a local scale and the access of the data into the cloud through the use of ThingSpeak also. This approach is aimed to present a practical and flexible model of air quality which can be used in large scales.

III. OBJECTIVES

This research project has the following goals:
1) System Development: To create a cost effective and small scale IoT device that measures air quality in real time with MQ2 and DHT11 sensors.
2) Microcontroller Interface: To connect these sensors to an ESP32 microcontroller which in turn will report data out wirelessly.
3) Real-Time Tracking: To have continuous unattended collection of key environmental variables like gas content, temp, and humidity.
4) Local Display: To present live sensor data directly on an SH1106 OLED display so as to immediately see the status of the environment at that location.
5) Cloud Connectivity: To upload the collected data to ThingSpeak which in turn makes it easy to store and also view the results from a remote location.
6) Data Interpretation: To transform raw sensor data into easy to understand values which in turn help people analyze the air quality.
7) Scalability: To develop a modular platform which easy to expand by adding in more sensors or to include new features as they become available.

IV. LITERATURE REVIEW

In recent years the issue of environmental monitoring has grown to be of great interest which we attribute to the increase in demand for real time and access to pollution data. We have seen put forth various communication technologies and system architectures which include the use of Zigbee networks, GSM based systems and also WiFi enabled Internet of Things solutions.

A. *Existing Systems*

In [7] the authors put forth a context aware computing model for the IoT which they designed to be adaptive in its sensing and in which they also included intelligent data processing. But that work was mainly in the high level system design which in turn did not pay attention to the issues of low cost and practical implementation.

In [8] Kumar and Singh presented an air pollution monitoring system that they built out of Arduino and a GSM module. While the system did enable remote data transmission that was a plus, it also required a dedicated SIM card which in turn had operational cost issues. Also the use of GSM in this case put forth higher latency and is not the best for continuous real time monitoring.

In many cases we see that Arduino based platforms are used. Though easy to deploy, Arduino boards do not have in board WiFi which in turn requires addition of other modules or shields thus increasing the system’s complexity and cost.

Also we have looked at Raspberry Pi based systems which do have greater computing power. But these systems are also more expensive and power hungry which in turn makes them a less than ideal choice for simple environmental monitoring applications.

B. *Comparative Analysis*

Table I reports which is the comparison between common monitoring technologies which we put forward the case for WiFi based systems.

Table I
COMPARISON OF MONITORING TECHNOLOGIES

Feature	GSM Based	Zigbee Based	Proposed (Wi-Fi)
Range	Very High	Low (Local)	High (Internet)
Cost	High (SIM cost)	Medium	Low
Power	High	Low	Medium
Setup	Complex	Complex	Simple
Internet	Direct	Gateway needed	Direct

What we see from the comparison is that which WiFi puts forth a balanced solution in terms of price, ease of implementation and real time connectivity. Also we see that they do away with the issue of recurring costs as is the case with GSM based systems and we note that in comparison to Zigbee which they don’t require extra gateways.

C. Research Gap and Proposed Approach

Existing solutions tend to focus on high end designs or they come with a higher price tag and greater system complexity. Also many of the present systems do not have local visualization and cloud based monitoring integrated into one platform.

To improve on these issues the introduced system uses the ESP32 microcontroller which in addition to other things has built in WiFi, low power use and enough processing power. We also put in the MQ2 and DHT11 sensors for gas concentration, temp and humidity which also play a role in air quality. Also we included an SH1006 OLED display for real time local visual feedback and ThingSpeak for cloud based data storage and analysis. This we feel is a cost effective, scalable and easy to use solution for real time air quality monitoring.

V. THEORETICAL FRAMEWORK

Understanding the basics of how sensors and communication systems work is key to accurate interpretation of data. MQ2 Gas Sensor Operation Principle.

A. MQ2 Gas Sensor Working Principle

The MQ2 gas sensor uses a chemiresistive sensing mechanism which is made up of Tin Dioxide (SnO_2 as the sensitive material. In clean air the sensor reports high resistance which is a result of oxygen molecule’s adsorption on to its surface.

As the sensor comes in contact with lower levels of gases such as LPG, methane, carbon oxide, or smoke which the sensor reports it has detected, the gas molecules react with the present oxygen ions on the sensor’s surface. This causes a shift which forces electrons back into the sensor’s conductive band thus reducing its electrical resistance.

$$R_s \propto \frac{1}{C} \qquad (1)$$

We see that there is a correlation between the sensor resistance (R_s) and the actual gas concentration (C) which in turn reports what is in the air.

To make this all make sense the ESP32 uses its Analog to Digital Converter (ADC) to track the changing voltages. Although the sensor does not give out exact numbers like a lab would report it does very well at reporting even the smallest changes in the local air quality.

B. DHT11 Sensor Principle

The DHT11 sensor reports on temperature and humidity that is determined from a thermistor and a capacitive humidity sensor element. The humidity sensor reports on changes in capacitance which in turn is a result of different moisture levels in the air, at the same time the thermistor measures temp based on changes in resistance.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

The sensor puts out calibrated digital signals which in turn are read by the ESP32 which does away with the need for in depth analog processing.

C. IoT Communication Mechanism

The system we have put in place uses HTTP (Hyper Text Transfer Protocol) for the transfer of sensor data to the ThingSpeak cloud platform. The ESP32 in this setup is a client which puts out HTTP GET requests that include sensor data which we encode into the URL./update?api_key=KEY & field1=x & field2=y& field3=z
which includes temperature, humidity, and gas sensor readings.

We find this method to be very lightweight and it works well for periodic data transfer in IoT settings. Also we used WiFi enabled ESP32 which gives us direct internet access which in turn eliminates the need for extra communication modules.

D. Display Mechanism

The Serial data (SDA) and Serial clock (SCL) makes it simple to interface with the ESP32 which in turn requires few pins. The ESP32 uses I2C bus to send out commands and data to the display which presents this info as pixels.

This feature allows for live update of sensor readings on the screen. Also having a local display is a great asset as data updates in real time without the need to check the cloud platform. Which in turn makes the system more user friendly and also very practical in situations where internet access is not present.

VI. SYSTEM ARCHITECTURE

Component Flow: The system follows a hybrid data flow model where environmental data is collected through sensors, processed by the ESP32 microcontroller, which is displayed locally and sent to the cloud for remote monitoring.

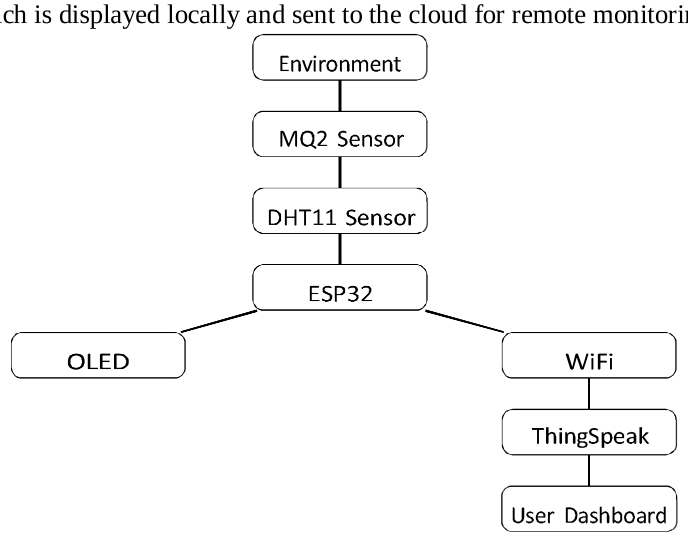

Figure 1. Compact System Architecture of the Proposed IoT-Based Monitoring System

VII. HARDWARE DESIGN AND IMPLEMENTATION

This section reports on the hardware components which also includes system and their interconnections.

A. Hardware Components

1) *ESP32 Microcontroller:* The ESP32 is at the core of this system. It is a low cost and power efficient microcontroller which also has in built WiFi which makes it very much at home in IoT.
   - Processor: Dual-core Tensilica Xtensa LX6
   - Clock Speed: Up to 240 MHz
   - Operating Voltage: 3.3V
   - WiFi: Built-in 802.11 b/g/n
   - GPIO Pins: Multiple digital and analog pins

2) *MQ2 Gas Sensor:* The MQ2 gas sensor which is used for detection of gases like LPG, methane, carbon monoxide, and smoke. It works via a heated sensing element and reports an analog voltage output that corresponds to gas concentration DHT11 Sensor.
3) *DHT11 Sensor:* The DHT11 sensor is out for temperature and humidity. It gives calibrated digital output which in turn makes it easy to work with the ESP32 for this application and also no extra signal conditioning is required. SH1106 OLED Display.
4) *SH1106 OLED Display:* The SH1106 OLED display is for real time visualisation of sensor data. It's used in conjunction with the ESP32 via the I2C protocol which it uses the SDA and SCL lines which also which reports to note the minimal wiring required.

B. Circuit Diagram and Connections

The circuit we put together on a breadboard has proper power and signal connections. The ESP32 we use to get analog input from the MQ2 sensor and digital from the DHT11 sensor, we process the values received and display them on the OLED screen. At the same time the data goes out to the ThingSpeak cloud platform over WiFi.

Table II
PIN CONNECTION TABLE

Component	ESP32 Pin	Description
MQ2 VCC	5V	Power Supply
MQ2 GND	GND	Ground
MQ2 A0	GPIO34	Analog Input
DHT11 VCC	3.3V	Power Supply
DHT11 GND	GND	Ground
DHT11 Data	GPIO4	Digital Signal
OLED VCC	3.3V	Power Supply
OLED GND	GND	Ground
OLED SDA	GPIO21	I2C Data
OLED SCL	GPIO22	I2C Clock

Also we power the ESP32 via a USB connection (5V) which at the same time powers the MQ2 sensor. We have proper grounding of all components for stable performance.

VIII. SOFTWARE DESIGN AND METHODOLOGY

The software controls data acquisition, processing, display, and cloud communication using the ESP32 microcontroller programmed in Arduino IDE.

A. Algorithm

1) Initialize sensors, OLED (I2C), and WiFi.
2) Connect to WiFi; retry until successful.
3) Read MQ2 (gas) and DHT11 (temperature, humidity).
4) Display values on SH1106 OLED.
5) Send data to ThingSpeak via HTTP.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

6) Delay 20 seconds before next cycle.

B. Software Flowchart

The flowchart in Fig. 2 represents the control logic of the system.

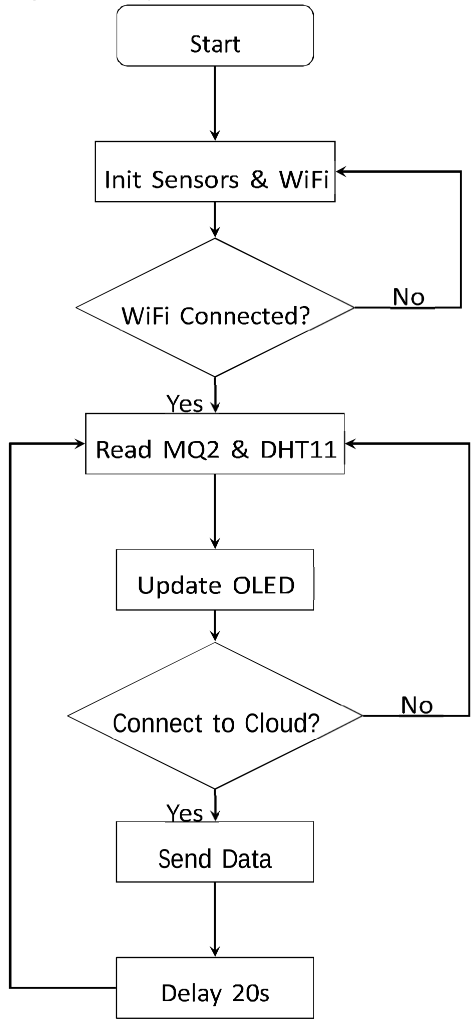

Figure 2. Flowchart of the system firmware.

IX. MATHEMATICAL MODELING AND CALIBRATION

The MQ2 gas sensor reports out in analog voltage which corresponds to the level of gas in the environment. But for that output to be converted into Parts Per Million (PPM) very precisely it requires a detailed calibration in controlled settings. In this work we took a simplified relative measurement approach.

The sensor resistance R_s can be estimated using:

$$R_s = \left(\frac{V_c}{V_{out}} - 1\right) \times R_L \quad (2)$$

where:

- V_c is the supply voltage,
- V_{out} is the analog output voltage from the sensor,
- R_L is the load resistance.

The value of $\frac{R_s}{R_0}$, which is what we see as the sensor resistance in clean air (R_0) is usually used to determine gas concentration from the log relationships presented in datasheets. But we note that for very accurate determination of R_0 which in turn gives us precise gas concentrations we require a controlled calibration which is out of the question for our present use.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Thus instead of coming up with exact PPM values we have the system report raw analog readings from the MQ2 sensor out which we in turn use as relative indicators of air quality. Also we map these to what we have as low, moderate, and high pollution levels. This approach we find to be a practical and at the same time very reliable for real time monitoring which also has the benefit of getting away from the issues related to precise gas concentration calibration.

X. COST ANALYSIS

One of the goals of this project was to develop a low cost air quality monitoring system. In table III we present the approximate price of the components used (as of to typical market prices in INR).

Table III
PROJECT COST ESTIMATION

Component	Approx. Cost (INR)
ESP32 Microcontroller	400
MQ2 Gas Sensor	150
DHT11 Sensor	80
SH1106 OLED Display	200
Breadboard	80
Jumper Wires	50
USB Cable	50
Total	1,010

While the total cost is a little over what basic microcontroller only sets do, it is still very much lower than that of commercial air quality monitoring devices which tend to cost out at over 5,000. We included both local display (OLED) and cloud connectivity features which at the same time we kept the price down.

XI. RESULTS AND DISCUSSION

The system was tested in three different environments to validate its sensitivity and accuracy.

A. Test Scenarios

1) Scenario A (Indoor/Clean): Tested inside a ventilated room.
2) Scenario B (Indoor/Cooking): Tested in a kitchen while cooking.
3) Scenario C (Outdoor/Traffic): Tested near a busy road intersection.

B. Observed Data

The ThingSpeak dashboard successfully logged data points every 15 seconds. The gathered data is summarized below:

Table IV
AVERAGE SENSOR READINGS

Scenario	Analog Reading	Air Quality Status
Scenario A	120 - 180	Good / Fresh
Scenario B	350 - 500	Moderate / Poor
Scenario	600 - 850	Hazardous

C		

C. Data Visualization

The data we collected from the sensors was uploaded on the ThingSpeak cloud platform and we used graphs to present it. We saw that there is great variation in sensor readings across different test scenarios.

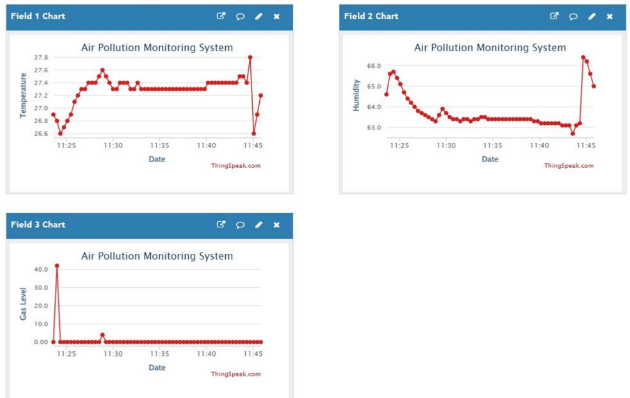

Figure 3. ThingSpeak visualization of sensor data showing variation in air quality across different conditions

We noted we had very large responses in the graphs which we associated with the sensor's exposure to smoke (e.g. incense stick) which proved the MQ2 sensor's high sensitivity. Also we noted that the response time of the system was around 2–3 seconds which we present as proof of near real time monitoring.

Also we noted the SH1006 OLED display which was very useful in giving out immediate local feedback by which also we update sensor values which in turn made the system very usable even when not connected to the cloud. Also we included a graphical dashboard which had gauge indicators (Green/Yellow/Red) which we felt made the system very easy to use.

XII. CHALLENGES FACED

During the development of the system, we faced several practical challenges which are:

1) Sensor Heating: The MQ2 gas sensor has a built in heating element which causes the sensor to heat up. This is normal for proper function but it require care in handling and proper placement to avoid issues of heat interference.
2) WiFi Performance: We had issues with the ESP32's connection in networks which had weak or unstable WiFi signals. To improve performance a reconnection feature was put into the software which will auto restore the connection.
3) Sensor Calibration: Determine out of what we thought to be clean air we had great difficulty in determining a base resistance value R0 for the MQ2 sensor. We had to go with relative base values which in the end were enough for us to use in the detection of air quality changes.
4) Sensor Accuracy Limitations: Also we found that in the case of low cost sensors like MQ2 and DHT11 which give out approximate results and are very much at the mercy of the environment we had to interpret the data as relative indicators and not as absolute values.

XIII. FUTURE ENHANCEMENTS

Presently the present system is that it does effective real time air quality monitoring but we may see to it that we improve and scale up its performance:

1) Advanced Sensor Integration: The system might be extended which includes the addition of different sensors like PMS5003 for particulate matter (PM2.5/PM10) and also more accurate gas sensors (e.g., MQ135 or MQ7) which in turn will present a very complete environmental picture.
2) Mobile Application Development: Also we see that a dedicated mobile app may be put together using platforms like Flutter to present real time monitoring, historical data visualization, and push notifications for when pollution levels go beyond pre determined thresholds.
3) Power Optimization: Implement in sleep mode features in ESP32 which along with use of renewable energy sources like solar panels may enable long term energy efficient operation in remote sites.

4) Data Analytics and Prediction: We can use machine learning models which will use historical data uploaded to the cloud. we might use models such as Linear Regression or LSTM is a way to predict future air quality trends.
5) Enclosure Design: We can design a custom enclosure which is 3D printed to improve on durability, protect against environmental factors, and also to make outdoor deployment easy.

XIV. CONCLUSION

This paper reports on the development and roll out of a low cost IoT enabled air quality system that uses ESP32 as the microcontroller. We present MQ2 and DHT11 sensors which we used to monitor gas concentration, temperature, and humidity in real time. Also we used SH1106 OLED display for immediate local results and ThingSpeak for cloud based data storage and remote access.

The study reports that we were able to see the system's performance in terms of it's ability to detect environmental changes which it does very well and report back in near real time. Also the sensor data is relative which we had put that into play, thus we present a reliable and practical solution for air quality trends.

We also looked at the large scale picture which is that the system is low cost and easy to implement which makes it a great fit for use in schools, residential areas, and for small scale monitoring. Also the system is designed with future growth in mind which makes it a great base to develop more complex environmental monitoring tools.

XV. ACKNOWLEDGMENT

The present authors wish to thank Mrs. Meetu Mann for her very useful guidance, support and encouragement which she gave us through out the project development. Also we are grateful to her for the insights she gave us into IoT architectures. We also acknowledge the Department of Computer Science and Engineering at Meerut Institute of Engineering and Technology for providing the required infrastructure and laboratory facilities which made this work possible.

REFERENCES

[1] World Health Organization (WHO), "Ambient (outdoor) air quality and health," [Online]. Available: https://www.who.int/news-room/ fact-sheets/detail/ambient-(outdoor)-air-quality-and-health
[2] Hanwei Electronics, "MQ-2 Gas Sensor Datasheet," [Online]. Available: https://www.handsontec.com/dataspecs/MQ-2.pdf
[3] ThingSpeak, "IoT Analytics Platform," [Online]. Available: https:// thingspeak.com/
[4] Espressif Systems, "ESP32 Datasheet," [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32 datasheet en.pdf
[5] Aosong Electronics, "DHT11 Temperature and Humidity Sensor Datasheet," [Online]. Available.
[6] Adafruit Industries, "SH1106 OLED Display Documentation," [Online]. Available.
[7] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," IEEE Communications Surveys and Tutorials, vol. 16, no. 1, pp. 414–454, 2014.
[8] K. A. Kumar and P. Singh, "Low Cost Air Pollution Monitoring System using Arduino and Gas Sensors," International Journal of Engineering Research and Technology, vol. 7, no. 11, 2018.
[9] O. Postolache, J. Pereira and P. Girao, "Smart Sensors Network for Air˜ Quality Monitoring Applications," IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 9, 2009.
[10] S. Devarakonda et al., "Real-time Air Quality Monitoring Through Mobile Sensing in Metropolitan Areas," in Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing, 2013.