AIR QUALITY MONITORING

TABLE OF CONTENT:

LIST OF FIGURE:

ABSTRACT:

Air pollution is both an environmental and a social problem, which affects human health, the ecosystem, and the climate. Air pollution is one of the largest environmental health risks. Quality of the air is the most important factor that directly causes diseases and decreases the quality of life. The development of air quality monitoring system will capture the amount of major pollutant and corresponding sources at appropriate level. These monitoring systems are important components for monitoring the quality of air and for controlling the concentrationofmainpollutantinthe urban areas. The internet of things is referred as the system of interrelated and internet connected objects, which is used to collect and transfer data. In order to monitor the pollution MQ135 sensor, GSM module is used for carrying out the research study in Sona College of Technology, Salem. The fundamental aspects of the proposed project are to reduce the cost of the infrastructure and enhance the data collection and dissemination to all stock

CHAPTER 1

INTRODUCTION:

IoT reveals that people are a source of data and in its second version, things or objects are creating data. It influences the computing and communication technology in the future. The vision of IoT makes the objects smarter for a better and easy world. The environmental data concentrationofmainpollutantinthe urban areas. The internet of Abstract Air quality monitoring and assessment using IoT 2 sampled and transmitted in time to the data center by air quality monitoring [1]. Internet acts as a backbone to IoT to connect with heterogeneous elements. IoT has many applications in various domains like Smart Cities, Smart Homes, IoT in transportation, IoT in Industries, Public Safety etc. This system will show the gases present in the atmospheric air and their purity there off. The system can be implemented in any kind of places like industries and houses, where the gases are mostly to be found across the threshold limit, the system gives an alert message. An alteration of air quality, which is characterized by the contents of chemical, biological or physical pollutants in the air, is called air pollution. The quality of air is indicated by air quality index[5]. Air pollution is mainly impacting the process of plant evolution by preventing photosynthesis, which leads to serious consequences in the purification of the air we are breathing in. 'Criteria air pollutants' is one of the terms used to describe air pollutants, which are the indicators of air quality. The regulations are based on criteria, which are related to health and environmental effects.

CHAPTER 2

2. SYSTEM REQUIRMENT:

2.1**Hardware requirement**

2.1.1 Arduino Uno;

Arduino UNO is developed by Arduino.cc which is an open source microcontroller board which is completely based on the microchip AT mega328P. It can detect the surroundings of the input

Features;

Its needs an operating voltage of 5v.The flash memory of the Arduino is 32 KB. It has an analog i/p pins that are 6.

2.1.2 Breadboard;

A breadboard is a construction base, which is used for the prototyping of electronics. It is easy to create temporary prototypes and experimenting with circuit design

Features

Breadboard dimensions are 6.5*4.4*0.3 inch. It has a withstanding voltage of 1,000V AC / 1 minute

2.1.3 Light-emitting diode(LED);

When current flow through the semiconductor light source, it emits light. The energy required for electrons to cross the band gap of the semiconductor determines the colour of the light

Features;

The LEDs are durable against impact and vibration. They are extremely efficient low energy light sources.

2.1.4 LCD (liquid crystal display) ;

This is a basic (16x2) 16 character by a 2-line display, which has green text on black background. It is used to indicate the Air and its quality in PPM.

Features ;

Every character can be build with a 5×8-pixel box. These are obtained in Blue & Green Backlight.

2.1.5 MQ135 gassensor ;

The MQ135 sensor is used in the air quality monitoring system. It can sense NH3, NOx, alcohol, Benzene, smoke, CO2, and some other gases. It gives the output in the form of voltage levels.

Features;

Wide detecting scope. It can be used as a digital or analog sensor.

2.1.6 Sim 900a gsmmodule;

GSM modem is otherwise said as GSM module, which gives different type of output with the help of PCB, connected to the module. The SIM900 is a complete Quad-band GSM or GRPS solution in a SMT module.

Features ;

Single supply voltage: 3.4V – 4.5V. It sends a Short message service (SMS

2.2 Software requirement:

2.2.1 Light-emitting diode(LED):

Arduino 1.6.13software;

It is a software which makes easy to write codes on board and upload it. It runs in windows, mac OS X, Linux

2.2.2Embedded Python language;

It is a generic termassociated with the hardware architecture. It is an extensionto python language.

SOFTWARE DESCRYPTION:

Python:

Python is a general-purpose language which means it is versatile and can be used to program many different types of functions. Because it is an interpreted language, it precludes the need for compiling code before execution and because it is a high-level programming language, Python is able to abstract details from code. In fact, Python focuses so much attention on abstraction that its code can be understood by most novice programmers.

Python code tends to be short and when compared to compiled languages like C and C++, it executes programs slower. Its user-friendliness makes it a popular language for citizen developers working with machine learning algorithms in low-code no-code (LCNC) software applications.

Python has a simply syntax and is known for having a large community that actively contributes to a growing selection of software modules and libraries. Python's initial development was spearheaded by Guido van Rossum in the late 1980s. Today, Python is managed by the Python Software Foundation**.**

Techopedia Explains Python

Python offers several frameworks for web development. A Python Web framework is a group of modules and libraries that enable programmers to re-use another developer's code. This collaborative approach can developers avoid dealing with low-level issues such as protocols, sockets and process/thread management.

Python Frameworks

Here are 10 frameworks that web developers, machine learning teams and data analytics teams should consider when using Python:

Open-source Django is a popular Python web framework that facilitates quick web design and development. Django is a free-to-use framework that enables developers to reuse code to build high-quality web apps and APIs. Django is known for:

- Helping programmers avoid security blunders.
- Supporting a data-driven architecture.
- Moving software from concept to launch quickly.

Pyramid is a compact open-source web framework that works in all supported versions of

Python. It offers the essential elements required for online applications including delivering static content and converting URLs to code. Some of Pyramid's attributes include:

- Security APIs that support authentication and authorization.
- A cookiecutter that generates sample Pyramid projects from project templates.
- Supporting the SQLAlchemy project and using its object-relational mapper (ORM) to interface with databases.

Bottle is a Web Server Gateway Interface (WSGI) micro-web framework for Python that is known for being lightweight and easy to use. Bottle is distributed as a single file module and the default Python library is the only dependency of the framework. It is is a popular framework for building mobile applications and supports:

- Python versions 2.7 and above.
- Mako, Jinja2, and Cheetah templates.
- WSGI-capable HTTP servers, including Bjoern, Google App Engine, fapws3 and CherryPy.
- URL mapping using condensed syntax.

CherryPy is an object-oriented HTTP framework that supports Apache and Microsoft IIS. Some of CherryPy's attributes include:

- A robust configuration system suitable for both developers and deployers.
- Built-in support for testing, coverage and profiling.
- Tools for authentication and caching.
- Flexible plugins.
- Robust configuration management.

Flask offers more control than its closest competitor, Django, and features support for unit testing. Along with RESTful request-dispatching and WSGI compatibility, Flask is known for:

- Providing an integrated development server with a debugger.
- Jinja2 templating (tags, filters, macros, and more).

paradigm.

- Automatically fix problems that may result in security risks.

- Support a database abstraction layer (DAL) that dynamically writes SQL is part of the framework.

Tornado is an open-source asynchronous framework for I/O operations. Tornado is known for supporting applications that require long-lived connections, real-time location services and allowing the integration of authentication and authorization methods from third parties.

BlueBream is a web application framework, server and library for Python programmers that was initially known as Zope 3. BlueBream is known for being durable, reliable and adaptive. It supports reusable software components as well as:

- WSGI (Web Server Gateway Interface) compatibility for Python.
- A template-development language that complies with XHTML.
- A program for creating forms automatically.

Grok

Grok is a robust framework for creating dependable and adaptable web applications. It supports DRY (Don't Repeat Yourself) software development and has a quick learning curve. Like other full-stack Python web frameworks, Grok features an intuitive UI (user interface).

Quixote

Quixote allows Python programmers to quickly create Web-based apps. This framework's objective is to offer web developers exceptional performance and flexibility for producing HTML with Python code

JSON

JSON, or JavaScript Object Notation, is a format used to represent data. It was introduced in the early 2000s as part of JavaScript and gradually expanded to become the most common medium for describing and exchanging text-based data. Today, JSON is the universal standard of data exchange. It is found in every area of programming, including front-end and server-side development, systems, middleware, and databases.

This article introduces you to JSON. You'll get an overview of the technology, find out how it compares to similar standards like XML, YAML, and CSV, and see examples of JSON in a variety of programs and use cases.

TABLE OF CONTENTS

A little bit of history

JSON was initially developed as a format for communicating between JavaScript clients and back-end servers. It quickly gained popularity as a human-readable format that front-end programmers could use to communicate with the back end using a terse, standardized format. Developers also discovered that JSON was very flexible: you could add, remove, and update fields ad hoc. (That flexibility came at the cost of safety, which was later addressed with the JSON schema.)

[Why Wasm is the future of cloud computing | The rise of WebAssembly]

In a curious turn, JSON was popularized by the AJAX revolution. Strange, given the emphasis on XML, but it was JSON that made AJAX really shine. Using REST as the

convention for APIs and JSON as the medium for exchange proved a potent combination for balancing simplicity, flexibility, and consistence.

Next, JSON spread from front-end JavaScript to client-server communication, and from there to system config files, back-end languages, and all the way to databases. JSON even helped spur the NoSQL movement that revolutionized data storage. It turned out that database administrators also enjoyed JSON's flexibility and ease of programming.

Today, document-oriented data stores like MongoDB provide an API that works with JSON-like data structures. In an interview in early 2022, MongoDB CTO Mark Porter noted that, from his perspective, JSON is still pushing the frontier of data. Not bad for a data format that started with a humble curly brace and a colon.

Why developers use JSON

No matter what type of program or use case they're working on, software developers need a way to describe and exchange data. This need is found in databases, business logic, user interfaces, and in all systems communication. There are many approaches to structuring data for exchange. The two broad camps are binary and text-based data. JSON is a text-based format, so it is readable by both people and machines.

JSON is a wildly successful way of formatting data for several reasons. First, it's native to JavaScript, and it's used inside of JavaScript programs as JSON literals. You can also use JSON with other programming languages, so it's useful for data exchange between heterogeneous systems. Finally, it is human readable. For a language data structure, JSON is an incredibly versatile tool. It is also fairly painless to use, especially when compared to other formats.

How JSON works

When you enter your username and password into a form on a web page, you are interacting with an object with two fields: username and passward.

CHPTER 3

3.SYSTEM DESIGN:

ALGORITHM;

The below algorithm is followed to collect data from the sensors.

1. Define the Blynk credentials, WiFi credentials, and other variables requiredfor the code.

2. Setup the serial communication and the Blynk connection using Blynk.begin(). 3. Set up the timer to run a function to send data to ThingSpeak every second.

4. Connect to the WiFi network using WiFi.begin() and wait until the connectionis established.

5. Define the changeMUX function and set the MUX_A pin as output.

6. In the loop, run the Blynk and timer functions, and read the sensor datafrom the analog pin A0.

7. Calculate the sensor value 1 (ppm (parts per million)) value for the sensordata using a formula.

8. Read the sensor data from A0 for a total of six times, and take the average ofthese readings to get the sensor value 0.

9. Change the MUX_A pin to HIGH, and read the sensor data from A0 foranother six times, and take the average of these readings to get the sensor value

1.

10. Connect to ThingSpeak using the WiFiClient object.

11. Build the request string with the ThingSpeak API key and field values (sensorValue0 and sensorValue1) and send the GET request using the HTTPClient object.

12. Delay for a second before running the loop again.

13. Define the function to be called by the timer to send data to ThinkSpeak.

14. Change the MUX_A pin to LOW and read the sensor data from A0.

15. Calculate the ppm value for the sensor data using a formula.

16. Change the MUX_A pin to HIGH and read the sensor data from A0 for atotal of six times, and take the average of these readings to get the sensor value

2.

17. Write the sensor value 1 and sensor value 2 to virtual pins V 1 and V 2 respectively using Blynk.virtualWrite().

3.2 SYSTEM ARCHITECTURE:

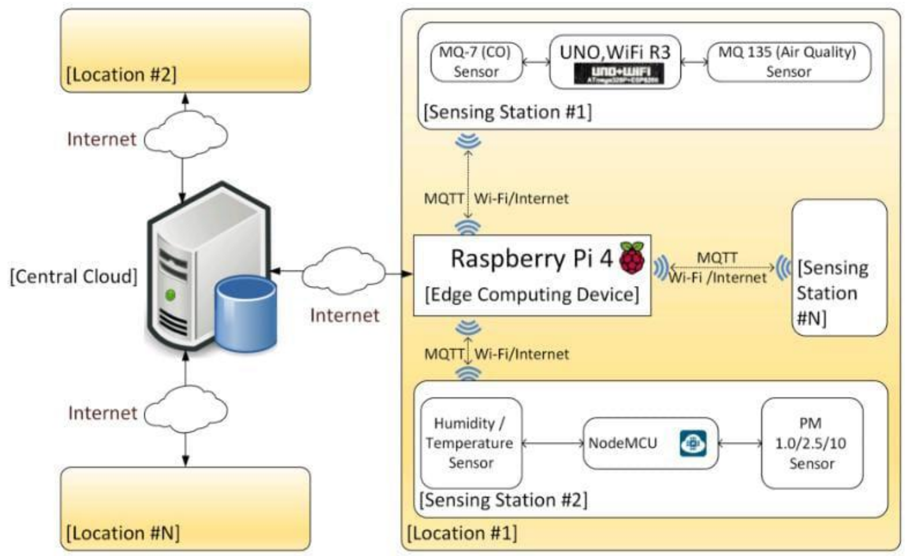
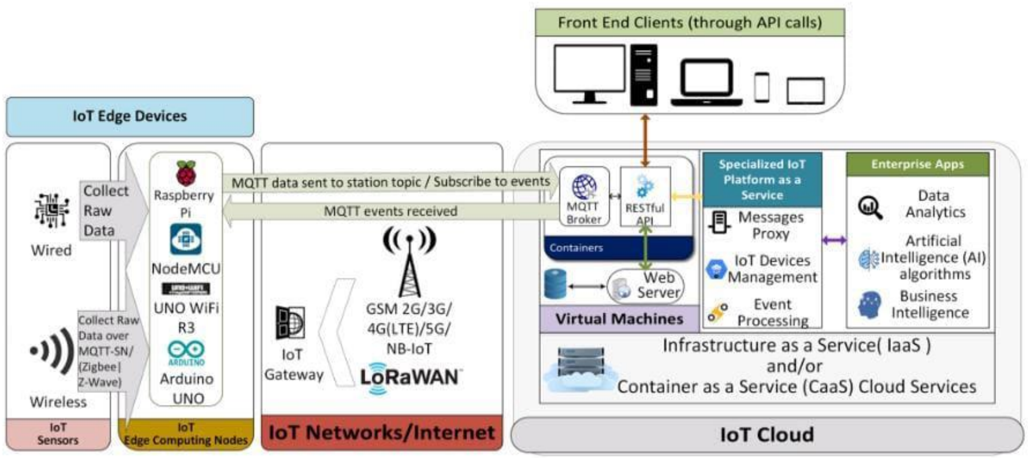


Fig. 1 Proposed IoT System architecture



3.3 UML DIAGRAM:

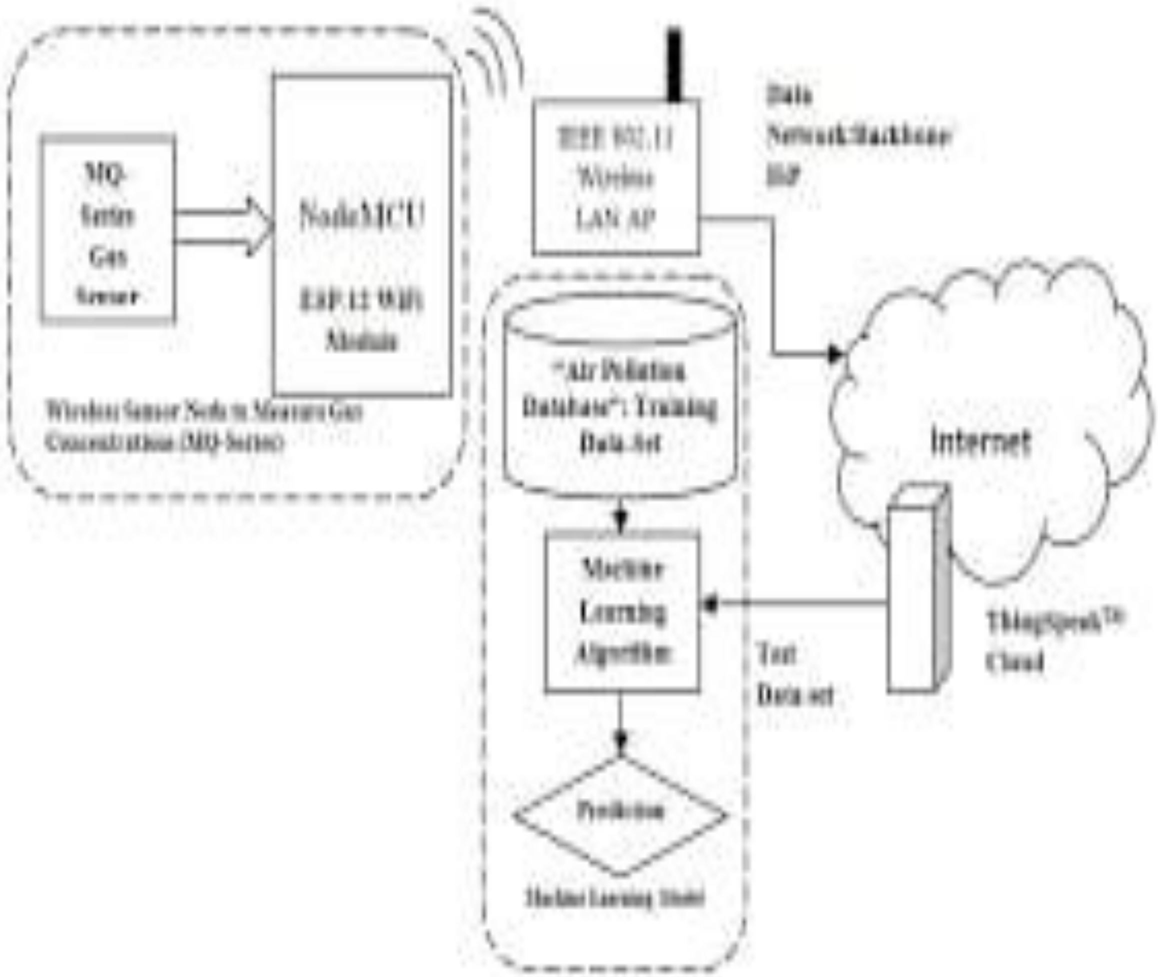


Fig 3.3.1 use case diagram

Fig 3.3.2 activity diagram

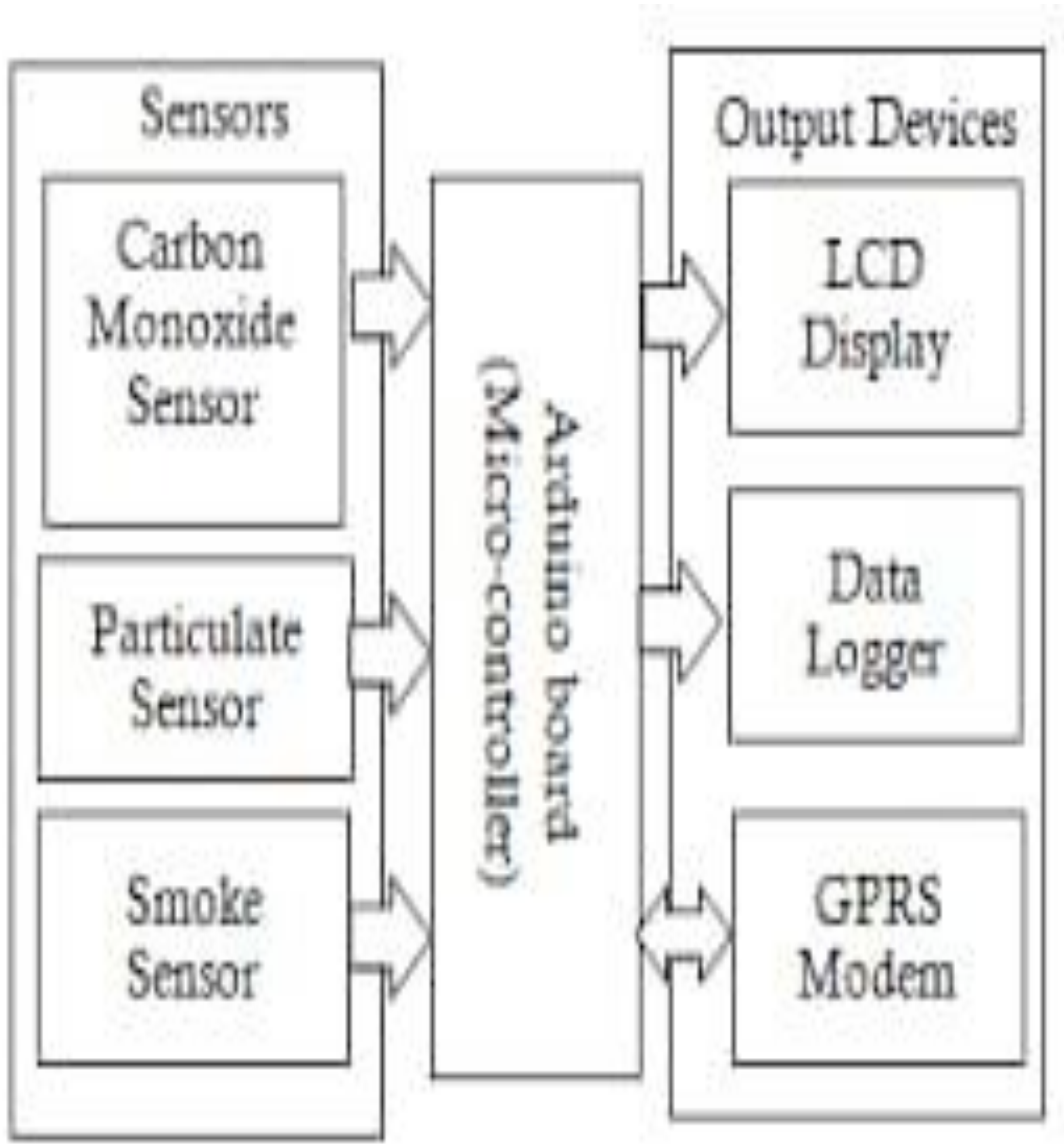


Fig 3.3.3 class diagram

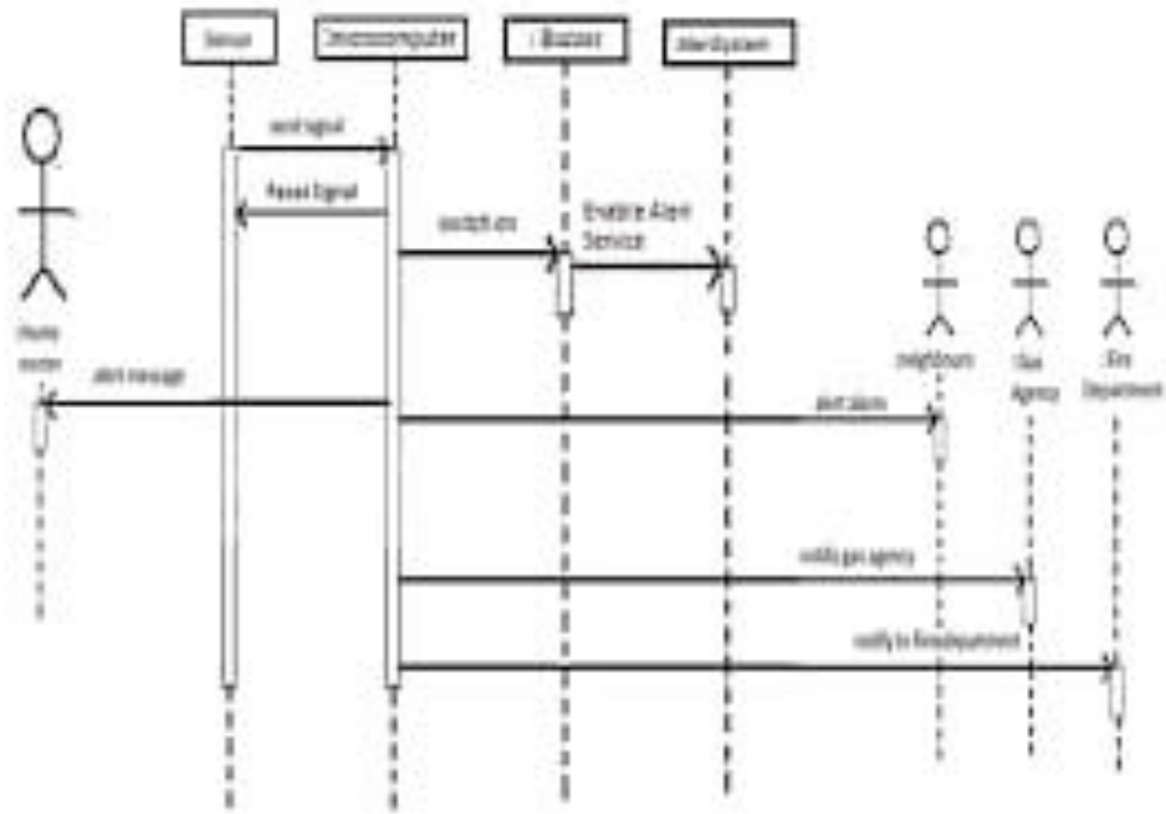


Fig 3.3.4 sequnce diagran

CHPTER 4

4.1 SYSTEM ANALYSIS:

4.1.1 EXISTING SYSTEM;

Real-time Data: IoT-based systems provide real-time air quality data, allowing for immediate responses to pollution events.

Cost-effective: IoT sensors are often more cost-effective to deploy and maintain than traditional monitoring stations.

Scalability: These systems can be easily scaled to cover larger areas, providing more comprehensive data.

Data Accessibility: Data from IoT sensors is often accessible online, allowing the public to access air quality information easily.

Wireless Connectivity: IoT sensors can communicate wirelessly, making deployment in remote or difficult-to-reach areas more feasible.

Sensor Integration: IoT sensors can measure multiple pollutants simultaneously, providing amore comprehensive view of air quality.

Limitations:

Sensor Accuracy: The accuracy of IoT sensors may vary, and calibration is essential to ensure data reliability.

Data Privacy and Security: IoT systems can be vulnerable to data breaches and privacy concerns, requiring robust security measures.

Maintenance: IoT sensors still require maintenance, including battery replacement and calibration.

Sensor Drift: Over time, sensor readings may drift, affecting data quality.

Limited Coverage: Despite scalability, IoT sensor networks may not cover all regions, leaving gaps in monitoring.

Power Requirements: IoT sensors need a power source, which can be a challenge in remote areas without reliable electricity.

Network Connectivity: Poor network connectivity can result in data transmission issues in some locations.

4.1.1 EXISTING SYSTEM ADVANTAGES:

Monitoring Stations: These are fixed, physical stations equipped with a variety of air quality monitoring instruments and sensors.

Instruments and Sensors: Monitoring stations contain a range of instruments to measure different air pollutants, including particulate matter (PM2.5, PM10), gases (NO2, SO2, CO, O3), volatile organic compounds (VOCs), and meteorological data (temperature, humidity, wind speed, etc.).

Data Logging and Transmission: Collected data is logged and transmitted to central databases or government agencies, often through wired connections.

Quality Assurance: Regular calibration and maintenance of sensors are essential to ensure data accuracy and reliability.

Geographic Distribution: Monitoring stations are strategically located across regions, cities, and urban areas to capture air quality variations.

Data Analysis: Specialists analyze the collected data to assess air quality and identify trends, pollution sources, and potential health risks.

Public Awareness: Some monitoring systems provide air quality information to the public through websites, mobile apps, or other means.

Regulatory Compliance: Air quality monitoring is often conducted to meet regulatory requirements, and data is used to develop air quality management policies.

Historical Data: Monitoring systems maintain historical records to track air quality changes over time and assess long-term trends.

Emergency Response: Monitoring data can trigger emergency responses, such as public alerts or regulations to reduce pollution in the event of poor air quality.

While these traditional systems have been in use for decades and offer reliable data, they may have limitations, such as cost, coverage gaps, and delays in data reporting. IoT-based systems have emerged to address some of these limitations and provide more real-time and cost-effective air quality monitoring solutions.

4.2 PROPOSED SYSTEM:

Sensors: These devices detect and measure air quality parameters. Different sensors are used for different pollutants, and they provide real-time data.

Data Processing: Collected sensor data is processed to calculate air quality indices, provide historical trends, and identify pollution sources.

Communication: Data can be transmitted via Wi-Fi, cellular networks, or other means to a central server or cloud for remote monitoring.

User Interface: A web or mobile app allows users to view air quality data, receive alerts, and access historical information.

Alarms and Alerts: The system can trigger alerts when air quality falls below acceptable levels, helping users take preventive measures.

Reporting: Regular reports and visualizations can provide insights into air quality changes over time.

Integration: Integration with weather data and geographical information can enhance the accuracy of air quality forecasts.

The specifics of the proposed system would depend on its intended use, location, and budget. Additionally, it may include features like GIS mapping, predictive modeling, and customization for specific industries or applications.

4.2.1 PROPOSED SYSTEM IN ADVANTAGES:

Health Benefits: Monitoring air quality can help protect public health by providing early warnings of poor air quality, which allows people to take precautions and avoid exposure to harmful pollutants.

Environmental Protection: It contributes to the protection of the environment by identifying pollution sources and helping authorities enforce regulations to reduce emissions.

Data-Driven Decision Making: Access to real-time air quality data empowers policymakers, businesses, and individuals to make informed decisions about outdoor activities, transportation, and land use.

Improved Safety: Air quality monitoring can enhance safety in workplaces, schools, and other settings by alerting users to hazardous conditions.

Research and Analysis: Long-term data collection enables researchers to analyze trends and correlations between air quality and health, climate, or local activities.

Accountability: Public access to air quality data encourages transparency and accountability among industries and government agencies responsible for emissions and pollution control.Efficient Resource Allocation: By identifying pollution hotspots, governments can allocate resources more efficiently to address air quality issues where they are most needed.

Regulatory Compliance: Industries can use air quality monitoring to ensure compliance with environmental regulations and avoid penalties.

Emergency Response: Air quality data is critical in responding to environmental emergencies, such as wildfires, chemical spills, or industrial accidents.

Public Awareness: Increased awareness of air quality issues can lead to behavioral changes and support initiatives to reduce pollution and improve air quality.

Economic Benefits: Better air quality can lead to increased property values, reduced healthcare costs, and improved quality of life, which can have economic benefits for communities.

Customization: Systems can be tailored to specific needs, allowing for integration with existing infrastructure and the adaptation of monitoring for different environments and pollutants.

Disadvantages

Limited Coverage: Many existing monitoring systems have limited geographic coverage, making it difficult to obtain comprehensive air quality data for specific areas.

High Costs: Building and maintaining monitoring infrastructure can be expensive, which can limit the number of monitoring stations and their ability to provide real-time data.

Lack of Sensor Calibration: Inaccurate or poorly calibrated sensors can lead to unreliable data, affecting the quality of air quality assessments.

Limited Pollutant Detection: Some systems may not detect a wide range of air pollutants, potentially missing harmful substances.

Data Accessibility: The data from these systems may not always be easily accessible to the public, limiting awareness and transparency.

Maintenance Challenges: Monitoring equipment requires regular maintenance, and the lack of it can result in malfunctioning sensors and inaccurate data.

Weather Interference: Weather conditions, such as rain or extreme temperatures, can affect the accuracy of monitoring systems.

Energy Consumption: Continuous operation of monitoring stations can consume a significant amount of energy, contributing to environmental impacts.Data Integration: Existingsystems may not always provide integrated data sources, making it challenging to understand the overall air quality picture

CHAPTER 5

5.1 PROJECT DESCRYPTION:

Internet of things system is a rapidly expanding idea in this era of industrialization technology meanwhile. It has become important for many manufacturing companies and other industries care about employees health, safety and other side effects. The internet of things that can monitor the physical objects that are connected to the Internet (wireless networks) and can be controlled from anywhere in the world. Environment issues may cause big disaster these days. One of the huge issues faced are Air pollution and sound pollution. By finding and detecting air pollution levels is the main objective

Air pollution is the biggest problem of every nation, whether it is developed or developing. Many times the emission of gases affects both the human beings and animals are affected by lung cancer, irritation of eye, breathing. Some other harmful effects caused by pollution are mild allergic reactions near throat, eyes and nose as well as some serious problems like bronchitis, heart diseases, pneumonia, lung and aggravated asthma. These are the problems that usually occurs while the industry does not take proper steps to reduce the gases as per government rules. Health problems have been growing at faster rate especially in urban areas of developing countries where industrialization and growing number of vehicles leads to release of lot of gaseous pollutants.

CHAPTER 6

6.1 IMPLEMENTATION:

6.1.1Front End;

```
            from machine import Pin
from time import sleep
import dht
import network
sta_if = network.WLAN(network.STA_IF)
if not sta_if.isconnected():
    print('connecting to network...')
    sta_if.active(True)
    sta_if.connect('Wokwi-GUEST', "")
    while not sta_if.isconnected():
        pass
    print('network config:', sta_if.ifconfig())



sensor = dht.DHT22(Pin(15))
while True:
  try:
    sleep(2)
    sensor.measure()
    temp = sensor.temperature()
    hum = sensor.humidity()
```

```
  temp_f = temp * (9/5) + 32.0
  print('Temperature: %3.1f C' %temp)
  print('Temperature: %3.1f F' %temp_f)
  print('Humidity: %3.1f %%' %hum)
except OSError as e:
  print('Failed to read sensor.')
```

6.1.2 Back End;


```
    {
  "version": 1,
  "author": "B. Mohan",
  "editor": "wokwi",
  "parts": [
    {
      "type": "wokwi-esp32-devkit-v1",
      "id": "esp",
      "top": 31.9,
      "left": 139.6,
      "rotate": 270,
      "attrs": { "env": "micropython-20231005-v1.21.0" }
    },
```

```
    { "type": "wokwi-dht22", "id": "dht1", "top": -95.7, "left": 167.4, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "dht1:VCC", "esp:3V3", "red", [ "v28.8", "h67.2" ] ],
    [ "dht1:GND", "esp:GND.1", "black", [ "v38.4", "h28.8" ] ],
    [ "dht1:SDA", "esp:D15", "green", [ "v48", "h38.5" ] ]
  ],
  "dependencies": {}
}
```

6.2 OUTPUT

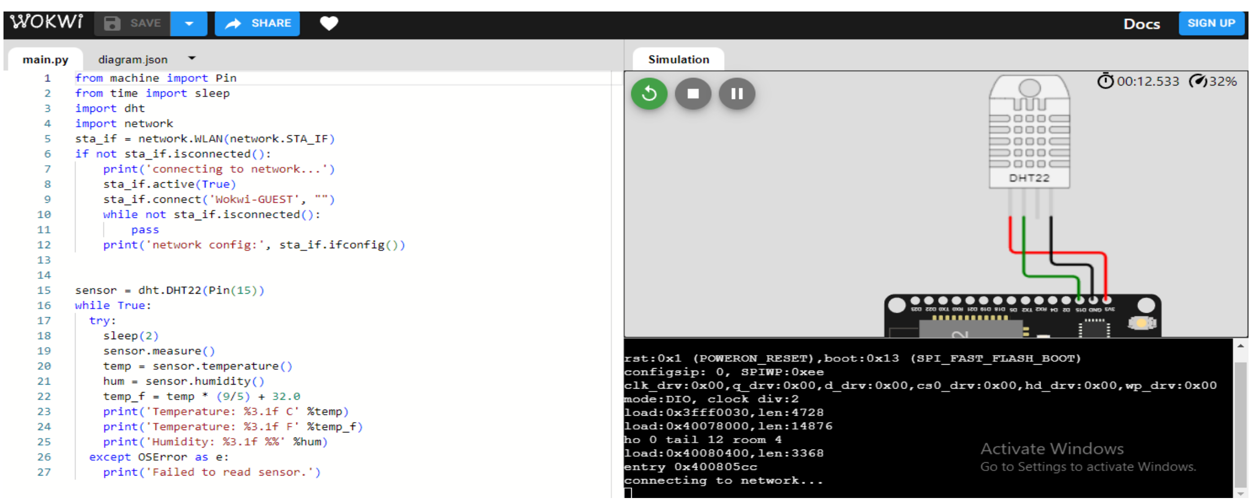


Fig 6.1.1: component name: DHT22

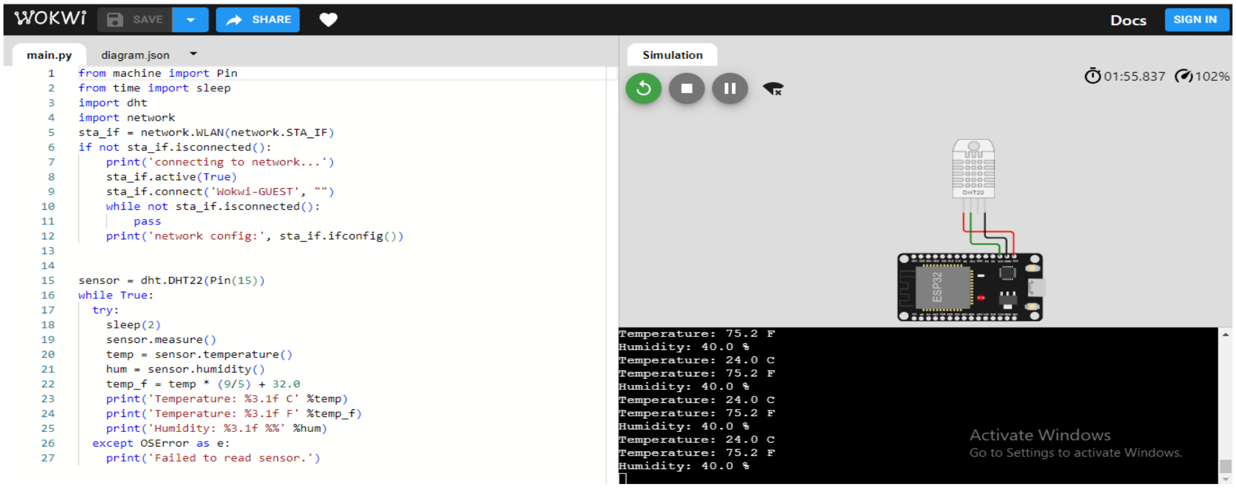


Fig6.1.2:Componentname:ARDUINO

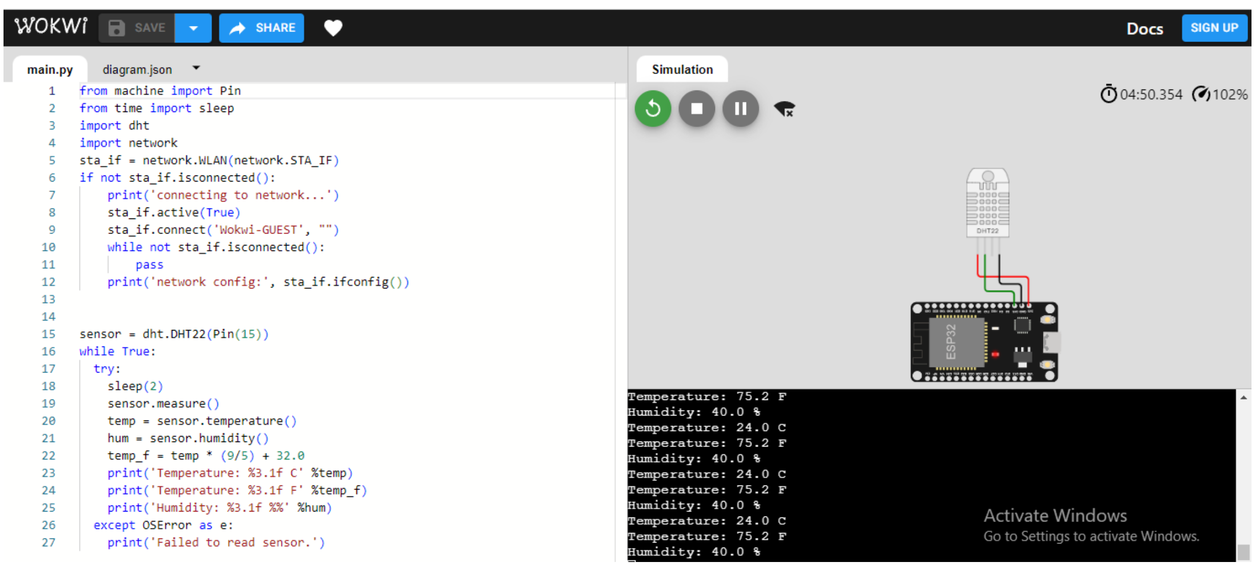


Fig 6.1.3:Needed components placed in arduino software

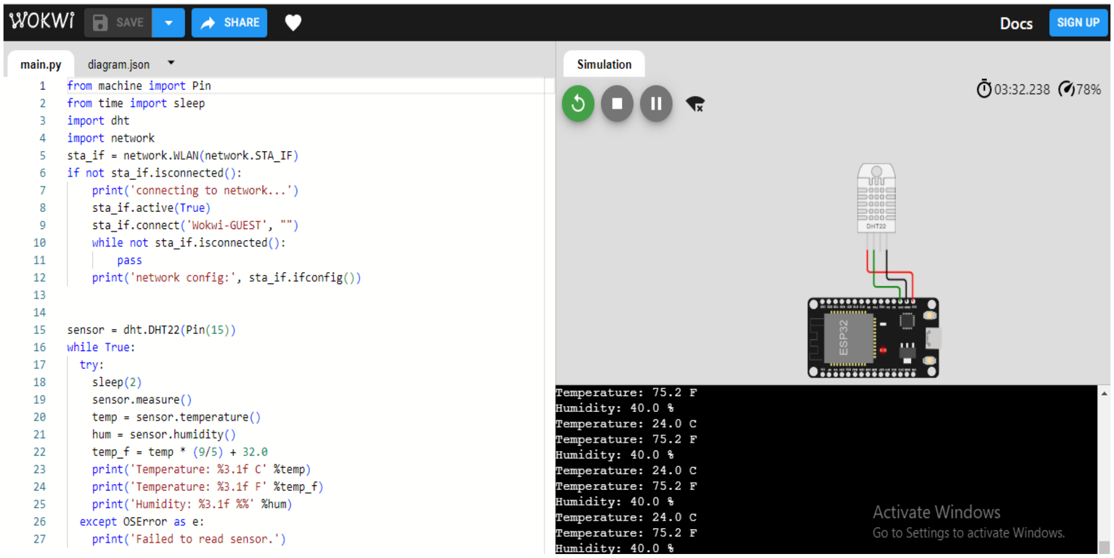


Fig6.2.1:connected:negative pin to the ground .

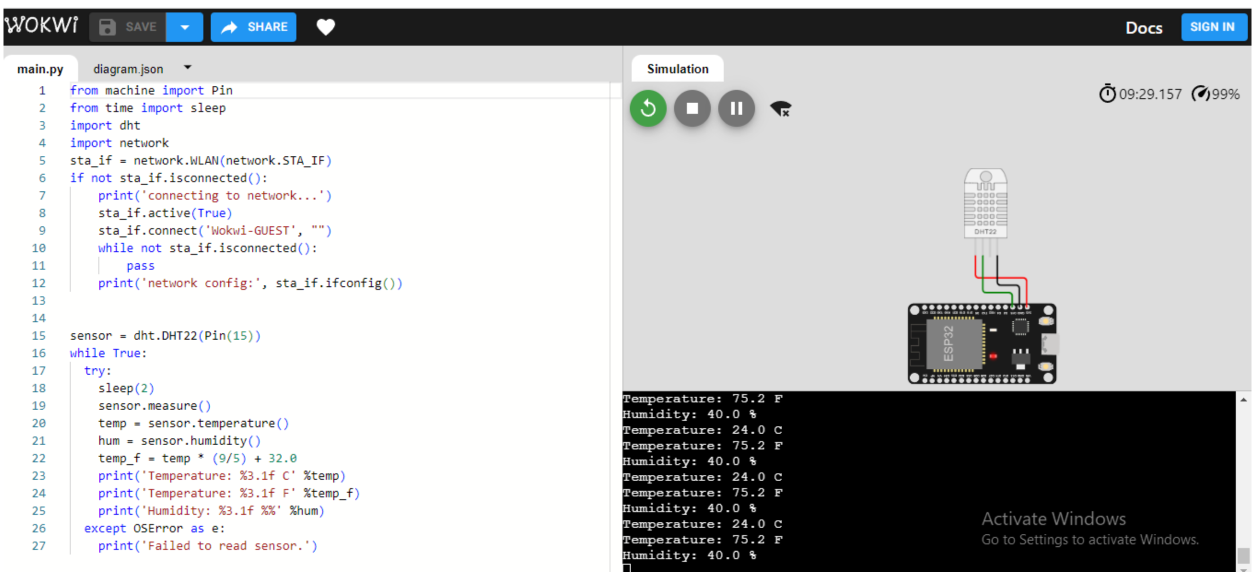


Fig 4.2.2:sensor 2nd pin connected to arduino's D

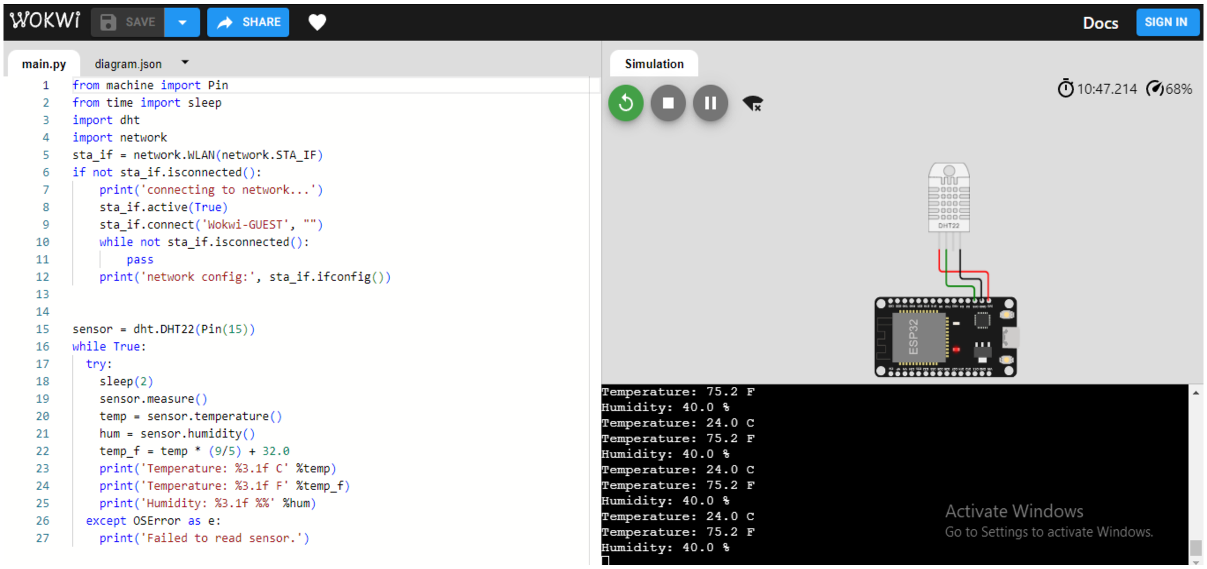


Fig4.2.3: sensor positive pin connected to vin

CHAPTER 7

7.1 COCLUSION:

Air quality monitoring systems are designed using different sensors for indoor and outdoor air quality monitoring in the previous works by using Bluetooth, GPS, GPRS wireless technologies. In a previous work WASP module is used which is costly. Instead of that different sensors can be used. The proposed system is developed for indoor air quality monitoring remotely. It is cost and energy efficient request and respond protocol is used along with combination of address and data centric protocols. Paper presents the summary of various techniques of air quality monitoring. These techniques are elaborately discussed in the paper. In the proposed system, one of the most preferred technique is cloud based air quality monitoring system. Using the same cloud data, website is hosted and data is displayed on the website.

CHAPTER 8

REFERANCE:

[1] Poonam Paul, Ritik Gupta, Sanjana Tiwari, Ashutosh Sharma, "IoT based Air Pollution Monitoring System with Arduino", IJART, May 2005

[2] Zishan Khan, Abbas Ali, Moin Moghal, "IoT based Air Pollution using NodeMCU and Thingspeak", IRANS, pp. 11-16, March 2014.

[3] SaiKumar, M. Reji, P.C. KishoreRaja "AirQuality Index in India", IEEE conference Chennai, August 2014.

[4] Mohan Joshi, "Research Paper on IoT based Air and Sound Pollution monitoring system", IETS Journal, pp. 11-17, September 2015.

[5] "Malaya Ranjan, Rai kumar, "Understanding Parts per million in real time air , Journal of Mathematics and advanced sciences, pp. 23-29, September 2009

[6] D. Bandyopadhyay and J. Sen, “Internet of Things: Applications and Challenges in Technology and Standardization,” Wirel. Pers. Commun., vol. 58, no. 1, pp. 49–69, May 2011.

[7] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A Survey,” Comput. Netw., vol. 54, no. 15, pp. 2787–2805, October 2010.

[8] H. Kopetz, Real-Time Systems: Design Principles for Distributed Embedded Applications. Boston, MA: Springer US, 2011, ch. Internet of Things, pp. 307– 323.

[9] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, “A Survey on Facilities for Experimental Internet of Things Research,” IEEE Communications Magazine, vol. 49, no. 11, pp. 58–67, November 2011.

[10] J. Kim, J. Lee, J. Kim, and J. Yun, “M2M Service Platforms: Survey, Issues, and Enabling Technologies,” IEEE Communications Surveys Tutorials, vol. 16, no. 1, pp. 61–76, January 2014.

[11] Jen-Hao Liu, Yu-Fan Chen, Tzu-Shiang Lin, And Da-Wei Lai ,Tzai-Hung Wen, Chih-Hong Sun, And Jehn-Yih Juang,Joe-Air Jiangdeveloped Urban Air Quality Monitoring System Based On Wireless Sensor Networks 2011 IEEE.

[12] Srinivas Devarakonda, Parveen Sevusu, Hongz Hang Liu, Ruilin Liu, Liviu Iftode, Badri Nath Urbcomp‟ Real-Time Air Quality Monitoring Through Mobile Sensing In Metropolitan Areas‟13, August 2013 Acm.

[13] Fouzi Harrou, Mohamed Nounou,Hazem Nounou “Detecting Abnormal Ozone Levels Using Pca Based Glr Hypothesis Testing”2013 Ieee Symposium On Computational Intelligence And Data Mining.

[14] Elias Yaacoub,Abdullah Kadri,Mohammad Mushtaha,And Adman AbuDayya,”Air Quality Monitoring And Analysis In Qatar Using A Wireless Sensor Network Deployment”596-601,2013 IEEE.

[15] Parr, T. W., Ferretti, M., Simpson, I. C., Forsius, M., & Kovács-Láng, E. (2002). Towards a long-term integrated monitoring programme in Europe: network design in theory and practice. Environmental monitoring and assessment, 78(3), 253-290.

[16] Jerrett, M., Arain, A., Kanaroglou, P., Beckerman, B., Potoglou, D., Sahsuvaroglu, T., ... & Giovis, C. (2005). A review and evaluation of intraurban air pollution exposure models. Journal of Exposure Science and Environmental Epidemiology, 15(2), 185