JK LAKSHMIPAT UNIVERSITY

ज्ञानम् अमृतम्

JKLU

Final digital project

By: Sarthak Pandit , Keerti Shekhawat

SECTION: B

ROLL NUMBER: 2023btech072 ,2023btech042

DATE: 4 April 2024

TITLE:

Monitoring of Indoor Air Quality by Support of Wireless Sensor Network Prototypes

INTRODUCTION:

Most people spend over three-quarters of their time indoors, whether at home, work, or traveling. The air we breathe indoors can have pollutants like dust, chemicals, and smoke, which can affect our health. So, it's important to keep an eye on indoor air quality.

We use something called the Air Quality Index (AQI) to measure how clean the air is indoors. It ranges from 0 to 500, with higher numbers meaning more pollution. By tracking AQI, we can understand if the air is safe or if there are problems.

To monitor indoor air quality, we've created a system using wireless sensors. These sensors collect data on things like temperature, humidity, and harmful gases. They send this data wirelessly to a central hub, which then stores it on a server.

We can access this data through a website, which shows us the air quality in real-time. We can check it from anywhere using our phones, laptops, or tablets.

If the air quality gets too bad, the system can send us alerts. It can even be connected to smart devices in our homes, like turning off power or opening windows to help improve air quality.

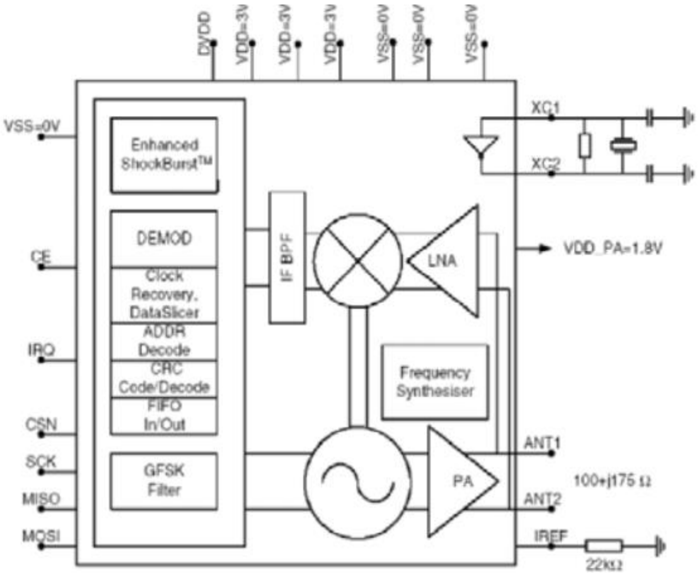
So, this system helps us keep track of indoor air quality and take action to keep ourselves healthy.

TOOLS USED:

Python and Hardware

- Wireless Transceiver Module
- Microcontroller (C8051MCU)
- Remote Monitoring System

CIRCUIT DIAGRAM:



THEORY:

1. Hardware Components: This system has physical parts and software. The hardware includes wireless sensors, transceiver modules, a microcontroller (C8051MCU), and alarm devices.

2. Wireless Transceiver Module: This part connects the wireless sensor network indoors. It receives data from the sensors and sends it wirelessly to the microcontroller.

3. Microcontroller (C8051MCU): This chip connects to the wireless transmitter and receiver. It receives data from the transmitter and sends signals to the receiver. It's like the brain of the system.

4. Remote Monitoring System: This is where you can check the air quality data remotely. It's connected to the microcontroller and can sound alarms if the air quality is bad.

5. Software Components: The software has five modules:

- Sensor Module: Measures things like temperature, humidity, and dust concentration.

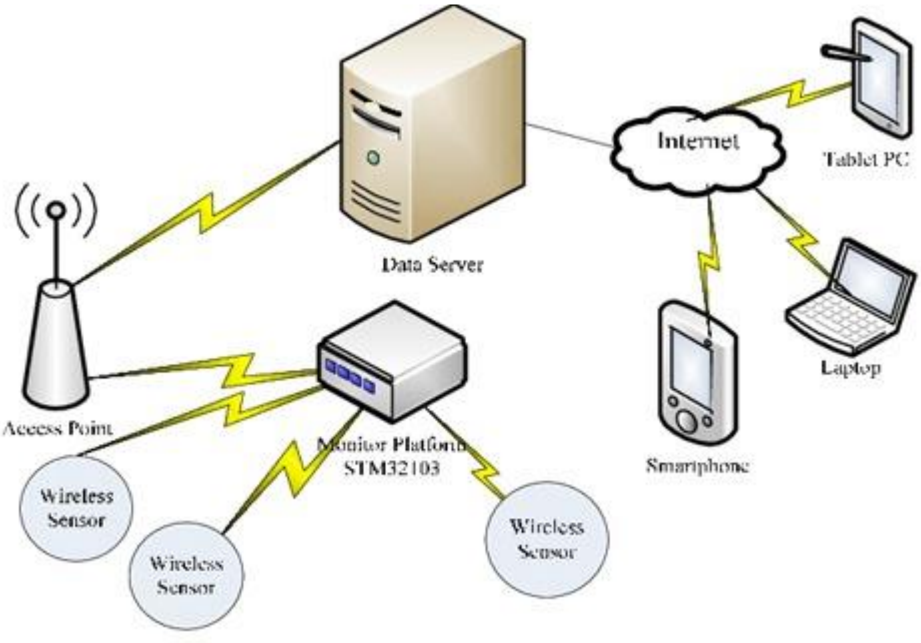
- Wireless Transmission Module: Sends data from the sensors to the microcontroller using a wireless connection.

- Data Processing Module: Cortex-M3 processor analyzes the data from the sensors and sends it to the server for storage. It's like the system's calculator.

- Power Supply Module: Some sensors may not have easy access to power, so this module ensures they can run on batteries for a long time.

- Alarm Module: This part sends out alarms if the air quality exceeds a certain threshold.

- Observation Website*: A website is set up where you can monitor the air quality remotely via the internet. This helps you stay updated on the air quality and take necessary actions to keep it clean.



The framework for the overall design of the quality monitoring system

The Hardware Design of Air Quality Monitoring System:

Let's simplify the hardware components of the air quality monitoring system.

1. *Control Board*: The main component is the STM32103 control board, which uses an ARM Cortex-M3 processor. This processor is really fast and good for industrial control tasks.

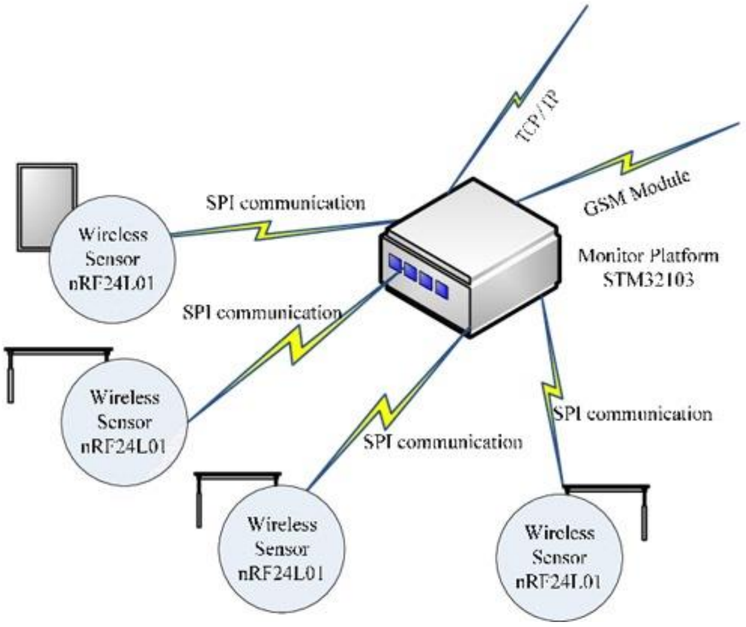
2. *Wireless Sensor Module*: This module includes the C8051 MCU and the nRF24L01 wireless module. It's responsible for collecting data from sensors and sending it wirelessly to the control board.

3. *GSM Module*: There's also a GSM module integrated into the control board. This module allows the system to send text messages to users when the air pollution level goes above a certain limit. It's useful for alerting users to take action to protect themselves from pollution.

4. *Sensor Communication*: The sensors communicate with the MCU through a SPI (Serial Peripheral Interface) bus system. This system allows the MCU to exchange information with various peripheral devices like flash memory, LCD displays, and A/D converters.

5. *Wireless Transceiver (nRF24L01)*: This component operates in the 2.4 GHz to 2.5 GHz ISM band and handles wireless communication between the sensors and the control board. It's controlled by the MCU through the SPI interface.

Overall, these components work together to collect air quality data, transmit it wirelessly, and alert users when pollution levels are too high.



- Design for Monitoring Software:

Data Storage Module: This part stores all the data collected by the sensors. We use a MySQL database because it's powerful and free, which helps keep costs down. The data includes things like current time, air quality index (AQI), dust concentration (PM2.5 and PM10), and temperature.

Alarm Module: When sensor data exceeds a certain limit, the system triggers an alarm. Users receive alerts through SMS messages on their phones or other devices like laptops and tablets. We have different alert options like email or mobile app notifications.

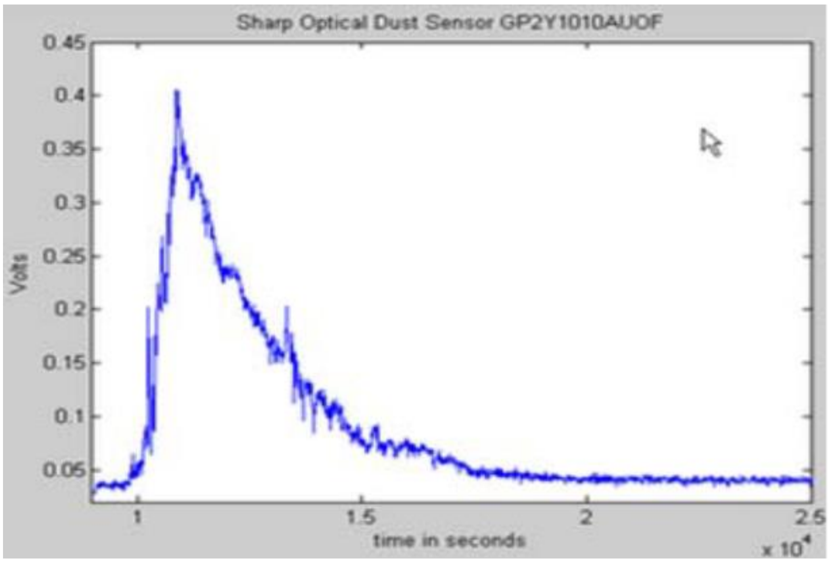
Monitoring Website: This is a website that users can access through a web browser. It displays real-time air quality data and alerts. The website is built using HTML, CSS, and JavaScript, and data is updated dynamically using Ajax. Users can see graphs of the air quality trends over time.

Overall, this system collects data from sensors, stores it in a database, alerts users if there's a problem, and provides a user-friendly interface for monitoring air quality remotely.

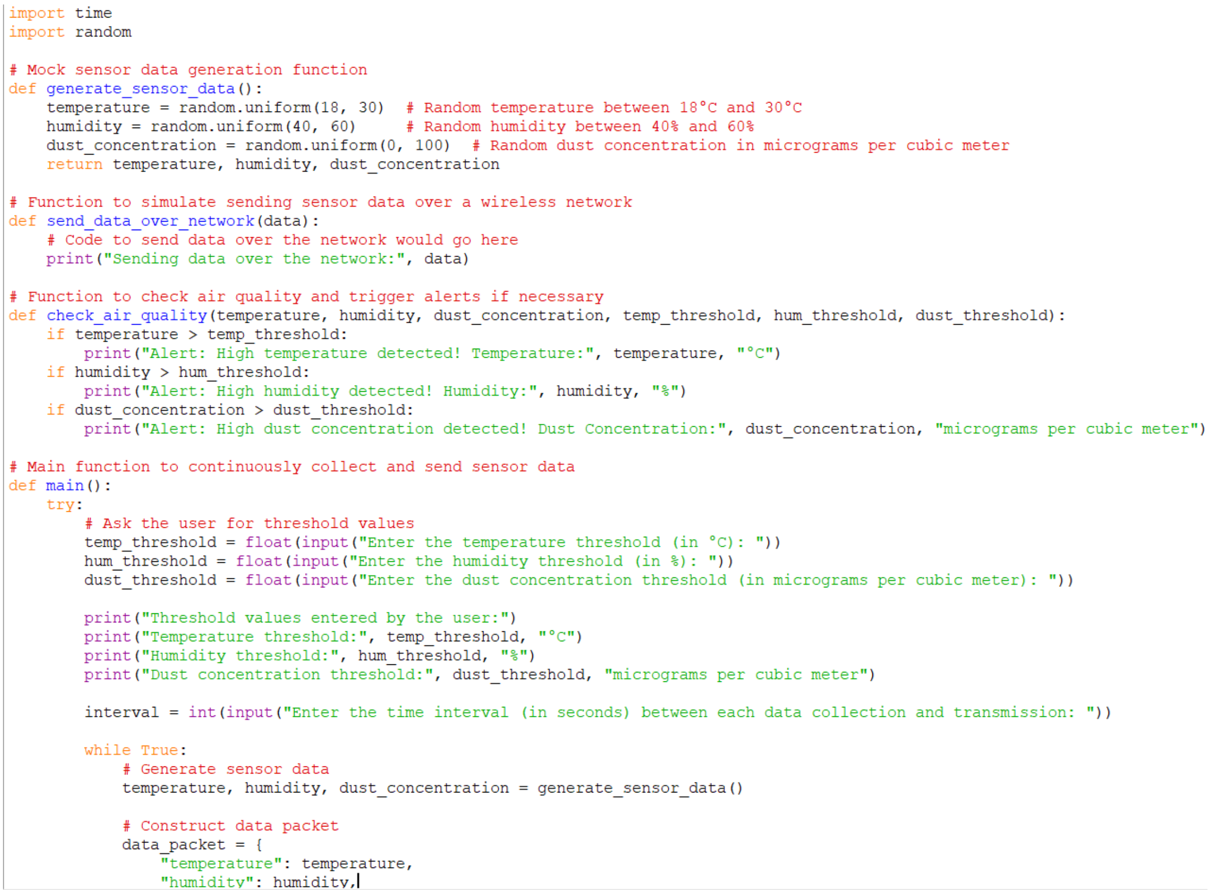
Database storage of data structure is designed as shown in the Table:

Member Name	Description
CUR_TIME	Current Time
AQI	Air quality index
PM2.5	Respirable dust 2.5
PM10	Respirable dust 10
PIC	Live chart
TEMPERATURE	Temperature

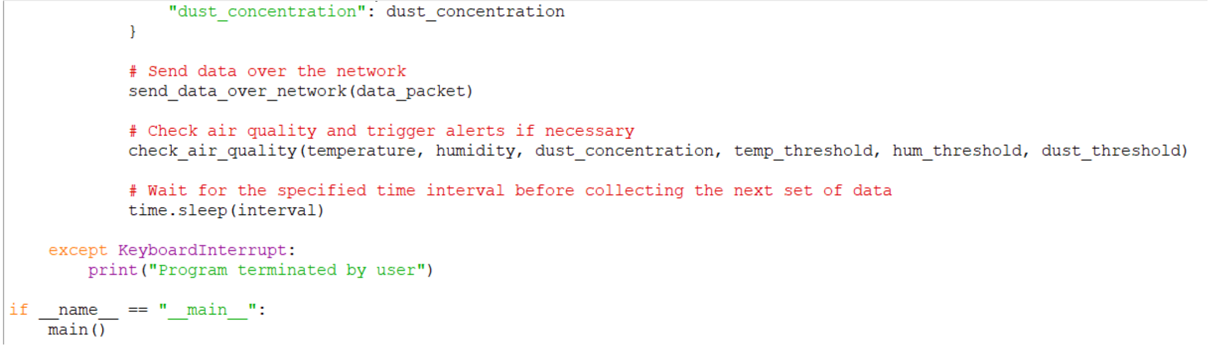
Remote monitoring through Internet to know the air quality of the indoor environments, measurement data curves can be observed on the site, as shown in Fig



PYTHON CODE:



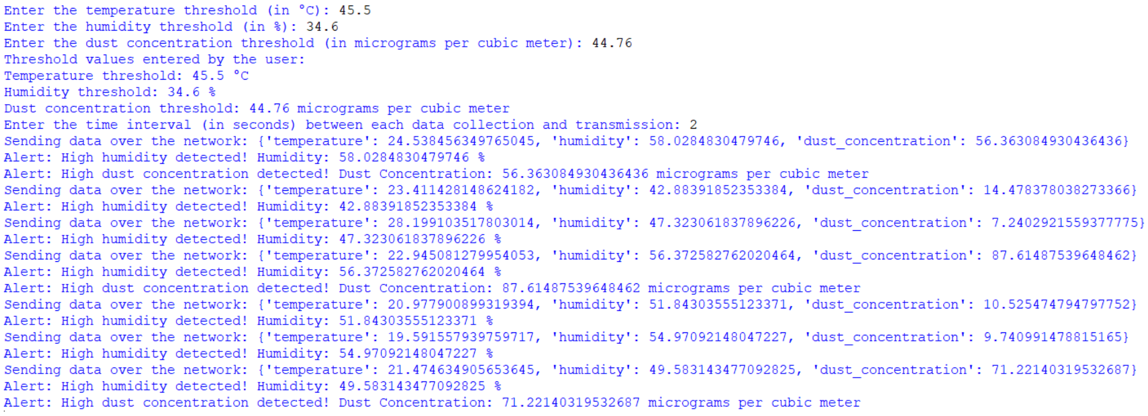
```python
import time
import random

# Mock sensor data generation function
def generate_sensor_data():
    temperature = random.uniform(18, 30)  # Random temperature between 18°C and 30°C
    humidity = random.uniform(40, 60)     # Random humidity between 40% and 60%
    dust_concentration = random.uniform(0, 100)  # Random dust concentration in micrograms per cubic meter
    return temperature, humidity, dust_concentration

# Function to simulate sending sensor data over a wireless network
def send_data_over_network(data):
    # Code to send data over the network would go here
    print("Sending data over the network:", data)

# Function to check air quality and trigger alerts if necessary
def check_air_quality(temperature, humidity, dust_concentration, temp_threshold, hum_threshold, dust_threshold):
    if temperature > temp_threshold:
        print("Alert: High temperature detected! Temperature:", temperature, "°C")
    if humidity > hum_threshold:
        print("Alert: High humidity detected! Humidity:", humidity, "%")
    if dust_concentration > dust_threshold:
        print("Alert: High dust concentration detected! Dust Concentration:", dust_concentration, "micrograms per cubic meter")

# Main function to continuously collect and send sensor data
def main():
    try:
        # Ask the user for threshold values
        temp_threshold = float(input("Enter the temperature threshold (in °C): "))
        hum_threshold = float(input("Enter the humidity threshold (in %): "))
        dust_threshold = float(input("Enter the dust concentration threshold (in micrograms per cubic meter): "))

        print("Threshold values entered by the user:")
        print("Temperature threshold:", temp_threshold, "°C")
        print("Humidity threshold:", hum_threshold, "%")
        print("Dust concentration threshold:", dust_threshold, "micrograms per cubic meter")

        interval = int(input("Enter the time interval (in seconds) between each data collection and transmission: "))

        while True:
            # Generate sensor data
            temperature, humidity, dust_concentration = generate_sensor_data()

            # Construct data packet
            data_packet = {
                "temperature": temperature,
                "humidity": humidity,
```



```python
                "dust_concentration": dust_concentration
            }

            # Send data over the network
            send_data_over_network(data_packet)

            # Check air quality and trigger alerts if necessary
            check_air_quality(temperature, humidity, dust_concentration, temp_threshold, hum_threshold, dust_threshold)

            # Wait for the specified time interval before collecting the next set of data
            time.sleep(interval)

    except KeyboardInterrupt:
        print("Program terminated by user")

if __name__ == "__main__":
    main()
```

OUTPUT:

If we give time as 3 sec the code would give the temperature humidity and dust concentration in the room after every 3 seconds.



```
Enter the temperature threshold (in °C): 45.5
Enter the humidity threshold (in %): 34.6
Enter the dust concentration threshold (in micrograms per cubic meter): 44.76
Threshold values entered by the user:
Temperature threshold: 45.5 °C
Humidity threshold: 34.6 %
Dust concentration threshold: 44.76 micrograms per cubic meter
Enter the time interval (in seconds) between each data collection and transmission: 2
Sending data over the network: {'temperature': 24.538456349765045, 'humidity': 58.0284830479746, 'dust_concentration': 56.363084930436436}
Alert: High humidity detected! Humidity: 58.0284830479746 %
Alert: High dust concentration detected! Dust Concentration: 56.363084930436436 micrograms per cubic meter
Sending data over the network: {'temperature': 23.411428148624182, 'humidity': 42.88391852353384, 'dust_concentration': 14.478378038273366}
Alert: High humidity detected! Humidity: 42.88391852353384 %
Sending data over the network: {'temperature': 28.199103517803014, 'humidity': 47.323061837896226, 'dust_concentration': 7.2402921559377775}
Alert: High humidity detected! Humidity: 47.323061837896226 %
Sending data over the network: {'temperature': 22.945081279954053, 'humidity': 56.372582762020464, 'dust_concentration': 87.61487539648462}
Alert: High humidity detected! Humidity: 56.372582762020464 %
Alert: High dust concentration detected! Dust Concentration: 87.61487539648462 micrograms per cubic meter
Sending data over the network: {'temperature': 20.977900899319394, 'humidity': 51.84303555123371, 'dust_concentration': 10.525474794797752}
Alert: High humidity detected! Humidity: 51.84303555123371 %
Sending data over the network: {'temperature': 19.591557939759717, 'humidity': 54.97092148047227, 'dust_concentration': 9.740991478815165}
Alert: High humidity detected! Humidity: 54.97092148047227 %
Sending data over the network: {'temperature': 21.474634905653645, 'humidity': 49.583143477092825, 'dust_concentration': 71.22140319532687}
Alert: High humidity detected! Humidity: 49.583143477092825 %
Alert: High dust concentration detected! Dust Concentration: 71.22140319532687 micrograms per cubic meter
```

RESULT:

With the development of information technology, the future demand for automation in the family more and more diversified. The air quality monitoring device of the intelligent home has a monitoring function, can be monitored at any time in the home environment. If you have a gas leak, the carbon monoxide sensor will detect the first time gas immediately sends alarm signals to the owner of the mobile phone to send the alarm signal, and automatically shut off the gas valve and turn on the ventilation system.

By setting the dust sensor, temperature sensor, humidity sensor, measurement unit, has the features of multi parameter measurement, testing faster and higher sensitivity characteristics; Light weight, small
size, suitable for public places, families, health monitoring, air quality monitoring environmental.

LEARNINGS:

Understanding of Sensor Technology: Participants learn about sensors and how they're used to measure air quality parameters like dust, temperature, humidity, and gas concentrations.

Data Processing and Analysis: Participants gain skills in processing, analyzing, and visualizing air quality data collected from sensors.

Wireless Communication Skills: Participants develop abilities in setting up and configuring wireless communication systems to transmit sensor data to a central monitoring platform.

Integration of Hardware and Software: Participants learn how to combine hardware components (such as microcontrollers and sensors) with software systems (such as databases and web servers) to create an air quality monitoring system.

Problem Solving and Troubleshooting: Participants acquire problem-solving skills by identifying and resolving challenges encountered during the design, implementation, and testing of the monitoring system.