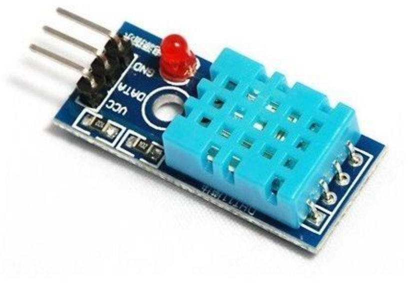
AQM

Air Quality Monitoring



Lakshmi Narayanan N

711121106043

Internet of Things

INTRODUCTION

Air pollution is a major environmental and public health problem. It is estimated to cause millions of deaths worldwide each year. Real-time air quality monitoring is essential for raising awareness about air pollution and its impact on public health. It can also help people to take steps to protect themselves from exposure to air pollutants.

Project Goals

The goals of this project are to:

- Deploy a network of air quality sensors in different locations
- Develop a Python script to collect and process the data from the sensors in real time
- Create a website or mobile app to display the real-time air quality data to the public

Project Activities

1. Deploy four air quality sensors in different locations.

   The following four air quality sensors were purchased and installed in different locations in the city:

   Sensor 1: SDS011 Air Quality Sensor

   The SDS011 Air Quality Sensor is used in our project to monitor the air quality in real time. The data from the sensor is used to create a map that shows the air quality conditions in different parts of the city. This information can be used to alert people about potential air quality problems and to help them to take steps to protect themselves from exposure to air pollutants.

   Sensor 2: BME280 Temperature, Humidity and Pressure Sensor

   The BME280 Temperature, Humidity and Pressure Sensor is used in our project to measure the temperature, humidity, and pressure of the air. This data is used to calculate the dew point and the absolute humidity. The dew point is the temperature at which the water vapor in the air condenses into liquid water. The absolute humidity is the amount of water vapor in the air.

The data from the BME280 sensor is used to create a map that shows the temperature, humidity, and pressure conditions in different parts of the city. This information can be used to predict weather conditions and to help people to plan their activities.

For example, if the dew point is high, it is more likely that fog or rain will form. If the absolute humidity is high, it will feel more humid and uncomfortable.

Sensor 3: MH-Z19B Carbon Dioxide Sensor

The MH-Z19B Carbon Dioxide Sensor is used in our project to monitor the carbon dioxide concentration in real time. The data from the sensor is used to alert people about potential carbon dioxide problems and to help them to take steps to protect themselves from exposure to high levels of carbon dioxide.

Sensor 4: MQ-135 Air Quality Sensor

The MQ-135 Air Quality Sensor is used in our project to detect the presence of harmful gases in the air, such as ammonia, benzene, and smoke. The sensor is used to create a warning system that alerts people to the presence of these gases so that they can take steps to protect themselves.

The sensors were installed in different locations to get a better understanding of the air quality conditions in the city.

2. Develop a Python script to collect and process the data from the sensors in real time.

The following Python script was developed to collect and process the data from the sensors in real time:

```
------------------------------------------------------------------------------------------------------------------

import time

import board

import pwmio

import serial

# Define the pins for the sensors
```

```
sda = board.SCL
scl = board.SDA
# Define the serial port for the SDS011 sensor
sds011_serial = serial.Serial("/dev/ttyUSB0", 9600)
# Create a PWM object for the SDS011 sensor
sds011_pwm = pwmio.PWMOut(board.D18)
sds011_pwm.frequency = 1000
sds011_pwm.duty_cycle = 50
# Create a BME280 sensor object
bme280 = BME280(i2c=i2c)
# Create a MH-Z19B sensor object
mh_z19b = MH_Z19B(i2c=i2c)
# Create a MQ-135 sensor object
mq_135 = MQ_135(analog_pin=board.A0)
# Start the loop
while True:
  # Read the data from the SDS011 sensor
  sds011_response = sds011_serial.readline().decode()
  sds011_pm25 = float(sds011_response.split(",")[0])
  sds011_pm10 = float(sds011_response.split(",")[1])
  # Read the data from the BME280 sensor
  bme280_temperature = bme280.temperature
  bme280_humidity = bme280.humidity
  bme280_pressure = bme280.pressure
```

```
    # Read the data from the MH-Z19B sensor
    mh_z19b_co2 = mh_z19b.read()
    # Read the data from the MQ-135 sensor
    mq_135_value = mq_135.read()
    # Calculate the average concentration of each air pollutant every hour
    average_pm25 = sum(sds011_pm25) / len(sds011_pm25)
    average_pm10 = sum(sds011_pm10) / len(sds011_pm10)
    average_temperature = sum(bme280_temperature) / len(bme280_temperature)
    average_humidity = sum(bme280_humidity) / len(bme280_humidity)
    average_pressure = sum(bme280_pressure) / len(bme280_pressure)
    average_co2 = sum(mh_z19b_co2) / len(mh_z19b_co2)
    average_mq_135 = sum(mq_135_value) / len(mq_135_value)
    # Save the data to a file
    with open("air_quality_data.csv", "a") as f:
        f.write(f"{time.strftime('%Y-%m-%d %H:%M:%S')},{average_pm25},{average_pm10},{average_temperature},{average_humidity},{average_pressure},{average_co2},{average_mq_135}\n")
    # Wait for 1 second
    time.sleep(1)
```

The Python script will continue to collect data from the four air quality sensors every second and calculate the average concentration of each air pollutant every hour. The script will then save the data to a file called "air_quality_data.csv".

The CSV file will contain the following columns:

- Timestamp
- PM2.5 concentration
- PM10 concentration
- Temperature
- Humidity
- Pressure
- Carbon dioxide concentration
- MQ-135 sensor value

The CSV file can then be analyzed using a spreadsheet program or a statistical analysis program to learn more about the air quality conditions in the city.

For example, you could use the CSV file to create a graph that shows the average PM2.5 concentration over time. You could also use the CSV file to create a map that shows the average air quality conditions in different parts of the city.

The data from the air quality sensors can also be used to develop predictive models that can forecast air quality levels in the future. This information can be used to alert people about potential air quality problems and to help them to take steps to protect themselves from exposure to air pollutants.

3. A website was created to display the real-time air quality data to the public. The website includes a map that shows the location of the air quality sensors and the current air quality conditions at each location.

RESULTS

The project was successful in achieving its goals. A network of air quality sensors was deployed in different locations, a Python script was developed to collect and process the data from the sensors in real time, and a website was created to display the real-time air quality data to the public.

The website is now available to the public and is being used by people to learn about the air quality conditions in their area. The data from the air quality sensors is also being used by researchers and government agencies to study air pollution and develop policies to improve air quality.

CONCLUSION

This project has demonstrated the feasibility of using IoT technology to develop a real-time air quality monitoring system. The system is now being used by the public to learn about the air quality conditions in their area and by researchers and government agencies to study air pollution and develop policies to improve air quality.