AIR QUALITY MONITORING

Developing an IoT-based Python script for Air Quality Monitoring involves several steps. Below are the key steps along with a simple script:

Step 1: Hardware Setup

1. Connect the MQ-135 gas sensor to your Raspberry Pi or other IoT device. Connect the sensor's analog output to one of the GPIO pins (e.g., pin 17).
2. Make sure your Raspberry Pi is properly powered and connected to the internet.

Step 2: Install Required Libraries

```bash
pip install RPi.GPIO
```

Step 3: Write the Python Script

Create a Python script, for example, `air_quality_monitor.py`.

```python
import RPi.GPIO as GPIO
import time

SENSOR_PIN = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(SENSOR_PIN, GPIO.IN)

AIR_QUALITY_GOOD = 0
AIR_QUALITY_MODERATE = 1
AIR_QUALITY_POOR = 2

def read_sensor():
    try:
        while True:
            sensor_value = GPIO.input(SENSOR_PIN)
            air_quality = get_air_quality(sensor_value)
            print(f"Air Quality: {air_quality_str(air_quality)}")
            time.sleep(5)

    except KeyboardInterrupt:
        GPIO.cleanup()
```

```
def get_air_quality(sensor_value):
    if sensor_value < 100:
        return AIR_QUALITY_GOOD
    elif sensor_value < 500:
        return AIR_QUALITY_MODERATE
    else:
        return AIR_QUALITY_POOR

def air_quality_str(air_quality):
    if air_quality == AIR_QUALITY_GOOD:
        return "Good"
    elif air_quality == AIR_QUALITY_MODERATE:
        return "Moderate"
    else:
        return "Poor"

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

Step 4: Run the Script

```bash
python air_quality_monitor.py
```

This script continuously reads the analog value from the MQ-135 sensor, categorizes it into air quality levels, and prints the result.

Step 5: Calibration and Fine-tuning

Calibrate the sensor by adjusting the thresholds in the `get_air_quality` function based on your specific sensor and environmental conditions.

Step 6: Integration with IoT Platform

For advanced monitoring, you can integrate the script with an IoT platform like ThingSpeak, Blynk, or AWS IoT. This typically involves sending data to the platform's API or using a specific library.

Step 7: Data Logging

Extend the script to log data to a file or database for historical analysis.