To build an IoT air quality monitoring system, first we need to configure IoT devices with suitable sensors and develop a Python script to send collected data to a data-sharing platform. Here I use a Raspberry Pi as the IoT device and a PM2.5/PM10 sensor for measuring particulate matter. We'll also used to send the data to a cloud-based platform like ThingSpeak.

Here are the steps to get started:

**Hardware and Sensor Setup:**

* **Choose Hardware:**

Acquire a Raspberry Pi and a PM2.5/PM10 sensor. The sensor can be connected via GPIO pins.

* **Connect the Sensor:**

Follow the sensor's datasheet to connect it to the Raspberry Pi's GPIO pins. Ensure the necessary power, ground, and data connections are correctly made.

**Software Setup:**

* **Raspberry Pi Setup:**

Install a suitable Raspberry Pi operating system (e.g., Raspbian).

Connect to the Raspberry Pi either through SSH or using a monitor and keyboard.

* **Install Required Python Libraries:**

Use the following commands to install necessary Python libraries:

```bash

pip install RPi.GPIO # For GPIO control

pip install pms5003 # For PM2.5/PM10 sensor

pip install requests # For sending data to ThingSpeak

```

* **Develop Python Script:**

Create a Python script to read data from the sensor and send it to the ThingSpeak platform. Here's a basic example script:

```python

import RPi.GPIO as GPIO

import pms5003

import requests

import time

# Initialize the GPIO pin for sensor data

GPIO.setmode(GPIO.BCM)

pm\_sensor = pms5003.PMS5003()

# ThingSpeak API endpoint and API Key

thingspeak\_url = 'https://api.thingspeak.com/update.json'

api\_key = 'YOUR\_THINGSPEAK\_API\_KEY'

try:

while True:

data = pm\_sensor.read()

if data:

pm2\_5 = data.pm\_ug\_per\_m3(2.5)

pm10 = data.pm\_ug\_per\_m3(10)

# Create a dictionary with the data to send

payload = {

'api\_key': api\_key,

'field1': pm2\_5,

'field2': pm10

}

# Send data to ThingSpeak

response = requests.post(thingspeak\_url, json=payload)

if response.status\_code == 200:

print("Data sent successfully")

else:

print("Failed to send data")

time.sleep(60) # Send data every minute

except KeyboardInterrupt:

GPIO.cleanup()

```

Replace `'YOUR\_THINGSPEAK\_API\_KEY'` with your actual ThingSpeak API key.

**Running the Script:**

* Save the Python script to your Raspberry Pi, e.g., `air\_quality\_monitor.py`.
* Run the script with the following command:

```bash

python air\_quality\_monitor.py

```

The script will continuously read data from the PM2.5/PM10 sensor and send it to the ThingSpeak platform. You can then log in to your ThingSpeak account to view and analyze the air quality data.