

Submission By: Yash Rajeshkumar Patel | SUID: 517958851

Introduction: This assignment outlines the development of an IoT solution hosted in the cloud, which aggregates sensor data for visualization. It leverages the MQTT protocol for the transmission of data and utilizes ThingSpeak for cloud-based data display.

Steps:

- **Sensor Simulation:** A Python script was written to emulate sensors, generating periodic and stochastic readings of CO2 levels, humidity, and temperature within set boundaries.
- **ThingSpeak Setup:** Chosen for its user-friendliness and suitability for IoT projects, ThingSpeak was adopted as the cloud backend, making use of its MQTT capabilities for data reception and storage.
- **Configuration and Authentication:** Established MQTT channels, clients, and devices with essential details such as channel ID, API key, Client ID, username, and password for secure connection to the ThingSpeak broker.
- **Data Monitoring and Analysis:** Utilized the Paho MQTT library and MQTT Topic to transmit data to the ThingSpeak channel. Monitored and analyzed the data to ensure consistent transmission of sensor information.

Conclusion: This IoT framework successfully acquires and transmits data from simulated sensors through the MQTT protocol to the ThingSpeak cloud backend. It illustrates the potential for environmental monitoring and offers a foundation for expansion into various IoT applications.

GitHub: https://github.com/yashpatel4900/Virtual_sensor

ThingSpeak: <https://thingspeak.com/channels/2488607>

Terminal:

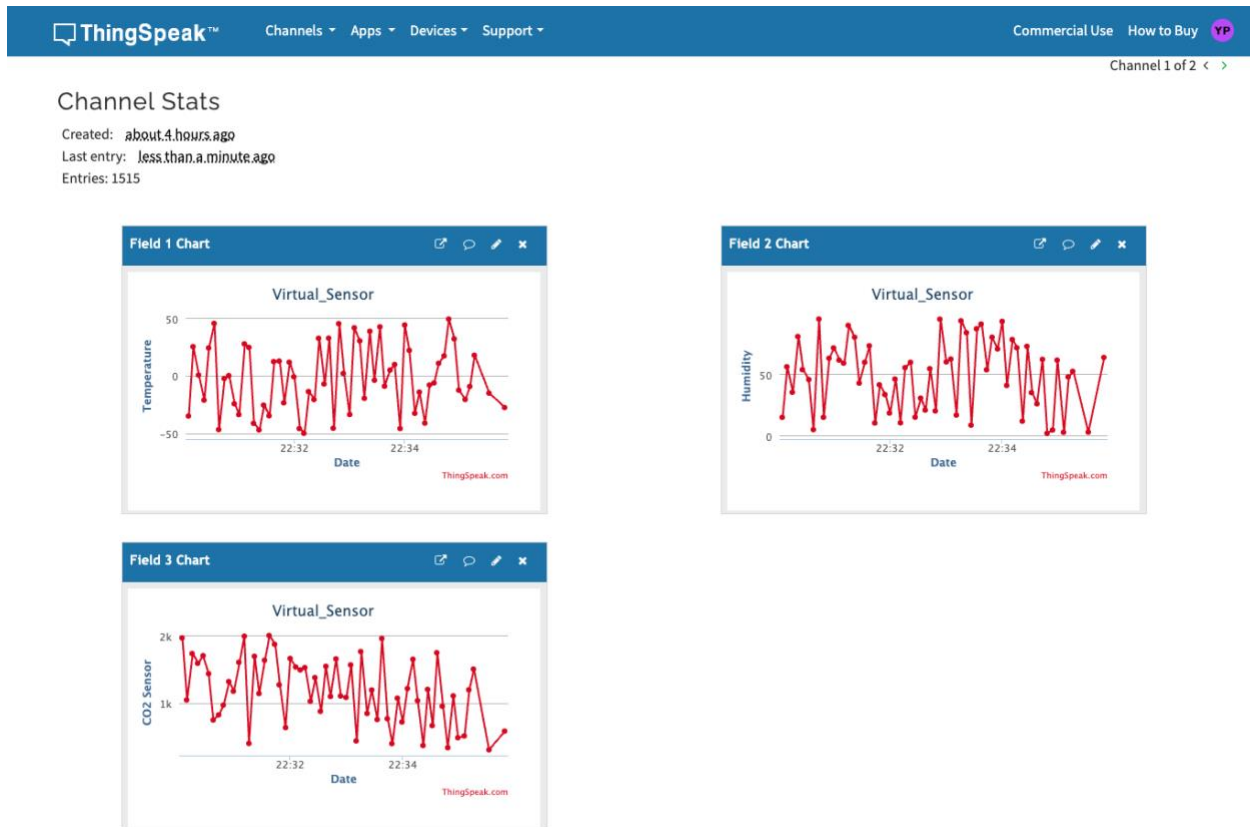
```

← → Assignment 3
iot_station.py ×
iot_station.py > TRANSPORT
6 # Configuration for connecting to the MQTT broker.
7 MQTT_CHANNEL_ID = "2488607"
8 MQTT_HOST = "mqtt3.thingspeak.com"
9 MQTT_CLIENT_ID = "HCKmNBwVJxw0BAYnLAc6FxU"
10 MQTT_USERNAME = "HCKmNBwVJxw0BAYnLAc6FxU"
11 MQTT_PASSWORD = "jzDbgfgljTSuakSJGZkH+Zvw"
12 TRANSPORT = "websockets"
13 PORT = 80
14
15 # MQTT topic for publishing data.

PROBLEMS TERMINAL DEBUG CONSOLE PORTS SEARCH ERROR COMMENTS OUTPUT
2024-03-27 22:36:53 - Sending data: field1=-1.5081942376677802&field2=74.37621397705487&field3=1955.3939438982209 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:36:59 - Sending data: field1=2.3656170159962713&field2=14.285071364118817&field3=1251.4366121072235 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:04 - Sending data: field1=-13.760761485042607&field2=16.89295466458478&field3=671.0925737327941 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:10 - Sending data: field1=-11.93938877528715&field2=20.50366490346728&field3=1415.63248163948 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:15 - Sending data: field1=-48.30513690346992&field2=65.46359911088814&field3=817.5437870468359 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:21 - Sending data: field1=7.135889643964212&field2=55.20020570962379&field3=1950.0743100143466 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:26 - Sending data: field1=16.999038457994402&field2=29.55042036593103&field3=1896.0561991723062 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:32 - Sending data: field1=12.159518337966581&field2=43.32070172999083&field3=518.9336640343097 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:37 - Sending data: field1=-22.665683701539674&field2=16.67110678485183&field3=770.6950151152769 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:42 - Sending data: field1=-22.350630034118147&field2=42.18138075949277&field3=1572.8519149512667 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:48 - Sending data: field1=24.811367907719585&field2=10.177398623121125&field3=1809.7440610231522 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:53 - Sending data: field1=-28.686323539898794&field2=94.6517040514718&field3=833.5834321269153 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:37:59 - Sending data: field1=7.590473825888353&field2=77.04469472218517&field3=943.8764528536404 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:38:04 - Sending data: field1=46.442204319612955&field2=63.89674923271863&field3=1247.5604637749361 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:38:10 - Sending data: field1=16.908247734293752&field2=33.118120673652406&field3=1071.8347209086733 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU
2024-03-27 22:38:15 - Sending data: field1=-21.343790898675774&field2=78.70169643438349&field3=1229.327998861601 to mqtt3.thingspeak.com, Client ID: HCKmNBwVJxw0BAYnLAc6FxU

```

Charts:



Code:

```
import paho.mqtt.publish as mqtt_publish
import random
import time
import datetime

# Configuration for connecting to the MQTT broker.
MQTT_CHANNEL_ID = "2488607"
MQTT_HOST = "mqtt3.thingspeak.com"
MQTT_CLIENT_ID = "HCkmNBwVJxw0BAYnLAc6FxU"
MQTT_USERNAME = "HCkmNBwVJxw0BAYnLAc6FxU"
MQTT_PASSWORD = "jzDbgfgljTSuakSJGZkH+Zvw"
TRANSPORT = "websockets"
PORT = 80

# MQTT topic for publishing data.
```

```
TOPIC = f"channels/{MQTT_CHANNEL_ID}/publish"

class SensorSimulator:
    """
    Emulates an IoT device generating environmental sensor data: temperature, humidity, and CO2 levels.
    """

    def __init__(self):
        self.temp_range = (-50, 50)
        self.humidity_range = (0, 100)
        self.co2_range = (300, 2000)

    def get_readings(self):
        """
        Simulates generating sensor data readings for temperature, humidity, and CO2.
        """
        temp = random.uniform(*self.temp_range)
        humidity = random.uniform(*self.humidity_range)
        co2 = random.uniform(*self.co2_range)
        return temp, humidity, co2

def publish_sensor_data():
    simulator = SensorSimulator()

    while True:
        temperature, humidity, co2 = simulator.get_readings()

        # Assembling the data payload for publishing.
        payload = f"field1={temperature}&field2={humidity}&field3={co2}"

        # Logging the payload and publishing details.
        current_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        print(f"{current_time} - Sending data: {payload} to {MQTT_HOST}, Client ID: {MQTT_CLIENT_ID}")

        # Publishing the data to the MQTT broker.
        mqtt_publish.single(
            TOPIC,
            payload,
```

```
hostname=MQTT_HOST,
transport=TRANSPORT,
port=PORT,
client_id=MQTT_CLIENT_ID,
auth={"username": MQTT_USERNAME, "password": MQTT_PASSWORD},
)

time.sleep(5)

# Start the data publishing routine.
if __name__ == "__main__":
    publish_sensor_data()
```