Explanation to create an IoT System:

Step 1: Create a ThingSpeak Channel

ThingSpeak serves as an IoT platform enabling data collection, analysis, and action from IoT devices. To set up a ThingSpeak channel:

Log in to ThingSpeak or sign up for an account if you're new.

Proceed to the Channels section and select "New Channel."

Input necessary details like channel name, description, and field labels.

Save the channel to create it.

Step 2: Integrate an MQTT Device

MQTT (Message Queuing Telemetry Transport) is a lightweight protocol ideal for IoT devices. To incorporate an MQTT device into your ThingSpeak channel:

Navigate to the "Apps" section in your ThingSpeak account and choose "ThingHTTP."

Create a new ThingHTTP, providing a name and description.

Configure the ThingHTTP to transmit data to your ThingSpeak channel via MQTT. This entails specifying MQTT broker details and authentication credentials.

Save the configuration.

Step 3: Develop Wokwi Code and Establish Connection

Wokwi offers an online platform for simulating and testing Arduino code. To write Wokwi code and establish a connection with ThingSpeak:

Craft your Arduino code within the Wokwi Arduino simulator. Ensure it includes MQTT client libraries to publish data to the ThingSpeak MQTT broker.

Configure the MQTT client in your code to connect to the ThingSpeak MQTT broker using the provided credentials from your ThingSpeak channel.

Validate your code within the Wokwi Arduino simulator to ensure its functionality.

Upon code satisfaction, upload it to your physical Arduino device.

Step 4: Analyze Data Using MATLAB

MATLAB provides robust tools for IoT data analysis. To analyze the data received from your ThingSpeak channel:

Utilize MATLAB's ThingSpeak support package to establish connectivity with your ThingSpeak channel.

Fetch data from your channel using MATLAB commands.

Apply analysis techniques using MATLAB's built-in functions and visualization tools, such as data filtering, trend analysis, and anomaly detection.

Present the results through visualizations like plots and graphs, leveraging MATLAB's visualization capabilities.

WOKWI Code:

import network

import time

import urandom

from umqtt.simple import MQTTClient

# ThingSpeak MQTT broker details

mqtt\_client\_id = "Fzw8Ay0wMjkaIgUWMBkhCzk"

mqtt\_user = "Fzw8Ay0wMjkaIgUWMBkhCzk"

mqtt\_password = "YUPZi9BJVykwp+Gsz2xbuDv1"

mqtt\_server = "mqtt3.thingspeak.com"

mqtt\_port = 1883

mqtt\_topic\_temperature = "channels/2488582/publish/fields/field1"

mqtt\_topic\_humidity = "channels/2488582/publish/fields/field2"

mqtt\_topic\_co2 = "channels/2488582/publish/fields/field3"

# Wi-Fi details

WIFI\_SSID = "Wokwi-GUEST"

WIFI\_PASSWORD = ""

# Historical data storage

historical\_data = []

# Function to generate random sensor values

def generate\_sensor\_data():

    temperature = urandom.uniform(-50, 50)

    humidity = urandom.uniform(0, 100)

    # Ensure CO2 value is within the acceptable range (300 to 2000 ppm)

    co2 = urandom.uniform(300, 2000)

    return temperature, humidity, co2

# Function to publish data to ThingSpeak

def publish\_to\_thingspeak(temperature, humidity, co2):

    client = MQTTClient(mqtt\_client\_id, mqtt\_server, user=mqtt\_user, password=mqtt\_password)

    client.connect()

    client.publish(mqtt\_topic\_temperature, str(temperature))

    client.publish(mqtt\_topic\_humidity, str(humidity))

    client.publish(mqtt\_topic\_co2, str(co2))

    client.disconnect()

# Connect to Wi-Fi

sta\_if = network.WLAN(network.STA\_IF)

sta\_if.active(True)

sta\_if.connect(WIFI\_SSID, WIFI\_PASSWORD)

# Wait for Wi-Fi connection

while not sta\_if.isconnected():

    pass

print("Connected to Wi-Fi")

# Main loop to generate and publish sensor data

while True:

    temperature, humidity, co2 = generate\_sensor\_data()

    historical\_data.append((temperature, humidity, co2))  # Store historical data

    if len(historical\_data) > 720:  # Approximately 5 hours with data every 5 seconds

        historical\_data.pop(0)  # Remove oldest data point if exceeds 5 hours

    publish\_to\_thingspeak(temperature, humidity, co2)

    print("Published: Temperature={:.2f}C, Humidity={:.2f}%, CO2={:.2f}ppm".format(temperature, humidity, co2))

    time.sleep(5)  # Adjust the delay as needed (Reduced to 5 seconds for faster data entry)

WOKWI Screenshots:

A computer screen shot of a computer

Description automatically generated

Thingspeak Screenshots:

A screenshot of a computer

Description automatically generated

MATLAB Code:

% Set your ThingSpeak channel ID and read API key

channelID = 2488582;

readAPIKey = 'MPNAS1NNOPSD721K';

% Get the current time and time five hours ago

currentTime = datetime('now', 'TimeZone', 'UTC');

fiveHoursAgo = currentTime - hours(5);

% Set up the ThingSpeak URL for fetching data

url = sprintf('https://api.thingspeak.com/channels/%d/feeds.json?api\_key=%s&start=%s&end=%s', ...

channelID, readAPIKey, datestr(fiveHoursAgo, 'yyyy-mm-ddTHH:MM:SSZ'), ...

datestr(currentTime, 'yyyy-mm-ddTHH:MM:SSZ'));

% Fetch data from ThingSpeak

data = webread(url);

% Extract sensor data

if ~isempty(data.feeds)

sensorData = [data.feeds.field1]; % Assuming the sensor data is in Field 1

timestamps = datetime({data.feeds.created\_at}, 'InputFormat', 'yyyy-MM-dd''T''HH:mm:ss''Z''', 'TimeZone', 'UTC');

% Display sensor data

disp('Sensor Data:');

disp(sensorData);

disp('Timestamps:');

disp(timestamps);

else

disp('No data found in the specified time range.');

end

A screenshot of a computer

Description automatically generated

MATLAB output:

Sensor Data:

27.5504241.174359.112859-34.40225-26.78329-12.933247.541168-10.8920829.54958-23.66084-41.6596537.85606-2.159023-19.96767-27.66344-10.47438-21.32213-9.1738225.218029-20.98063-40.8530513.953888.23007812.7586742.41808-7.04966823.69806-37.92054-25.1162525.832390.09102821-17.1399523.8016538.72827-3.24143220.3617723.32346-47.482434.03074-36.33099-10.287522.18366-5.2284365.322635

Timestamps:

Columns 1 through 8

27-Mar-2024 21:48:32 27-Mar-2024 21:49:48 27-Mar-2024 21:49:54 27-Mar-2024 21:50:00 27-Mar-2024 21:50:06 27-Mar-2024 21:50:12 27-Mar-2024 21:50:18 27-Mar-2024 21:50:24

Columns 9 through 16

27-Mar-2024 21:50:30 27-Mar-2024 21:50:38 27-Mar-2024 21:50:45 27-Mar-2024 21:50:51 27-Mar-2024 21:50:58 27-Mar-2024 21:51:04 27-Mar-2024 21:51:10 27-Mar-2024 21:51:16

Columns 17 through 24

27-Mar-2024 21:51:22 27-Mar-2024 21:51:29 27-Mar-2024 21:51:38 27-Mar-2024 21:53:19 27-Mar-2024 21:53:25 27-Mar-2024 21:54:46 27-Mar-2024 21:54:52 27-Mar-2024 21:55:27

Columns 25 through 32

27-Mar-2024 21:55:33 27-Mar-2024 21:55:39 27-Mar-2024 21:55:45 27-Mar-2024 21:55:51 27-Mar-2024 21:55:58 27-Mar-2024 21:56:04 27-Mar-2024 21:56:10 27-Mar-2024 21:56:16

Columns 33 through 40

27-Mar-2024 21:56:23 27-Mar-2024 21:56:29 27-Mar-2024 21:56:35 27-Mar-2024 21:56:41 27-Mar-2024 21:56:47 27-Mar-2024 21:56:53 27-Mar-2024 21:56:59 27-Mar-2024 21:57:06

Columns 41 through 48

27-Mar-2024 21:57:12 27-Mar-2024 21:57:18 27-Mar-2024 21:57:27 27-Mar-2024 21:57:32 27-Mar-2024 21:57:54 27-Mar-2024 21:58:00 27-Mar-2024 21:59:30 27-Mar-2024 21:59:46

Columns 49 through 56

27-Mar-2024 22:00:52 27-Mar-2024 22:00:59 27-Mar-2024 22:01:04 27-Mar-2024 22:02:00 27-Mar-2024 22:02:06 27-Mar-2024 22:02:19 27-Mar-2024 22:02:24 27-Mar-2024 22:03:06

Columns 57 through 64

27-Mar-2024 22:03:13 27-Mar-2024 22:03:19 27-Mar-2024 22:03:25 27-Mar-2024 22:03:31 27-Mar-2024 22:03:37 27-Mar-2024 22:03:43 27-Mar-2024 22:03:49 27-Mar-2024 22:03:55

Columns 65 through 72

27-Mar-2024 22:04:02 27-Mar-2024 22:04:07 27-Mar-2024 22:04:13 27-Mar-2024 22:04:19 27-Mar-2024 22:04:26 27-Mar-2024 22:04:32 27-Mar-2024 22:04:38 27-Mar-2024 22:04:44

Columns 73 through 80

27-Mar-2024 22:04:50 27-Mar-2024 22:04:57 27-Mar-2024 22:05:03 27-Mar-2024 22:05:09 27-Mar-2024 22:05:15 27-Mar-2024 22:05:22 27-Mar-2024 22:05:28 27-Mar-2024 22:05:34

Columns 81 through 88

27-Mar-2024 22:05:40 27-Mar-2024 22:05:46 27-Mar-2024 22:05:53 27-Mar-2024 22:05:59 27-Mar-2024 22:06:07 27-Mar-2024 22:06:13 27-Mar-2024 22:06:19 27-Mar-2024 22:06:25

Columns 89 through 96

27-Mar-2024 22:06:31 27-Mar-2024 22:06:37 27-Mar-2024 22:06:43 27-Mar-2024 22:06:49 27-Mar-2024 22:06:55 27-Mar-2024 22:07:01 27-Mar-2024 22:07:07 27-Mar-2024 22:07:13

Columns 97 through 104

27-Mar-2024 22:22:45 27-Mar-2024 22:22:51 27-Mar-2024 22:22:57 27-Mar-2024 22:23:03 27-Mar-2024 22:23:09 27-Mar-2024 22:23:15 27-Mar-2024 22:23:21 27-Mar-2024 22:58:07

Columns 105 through 106

27-Mar-2024 22:58:13 27-Mar-2024 22:58:21