DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 2

Student Name: Pranjal Singh **UID: 22BCS13041**
Branch: CSE **Section/Grp: FL_IOT_601/A**
Semester: 6th **Date of Performance: 23/1/25**
Subject: Fdtn Of CLOUD IOT EDGE ML **Subject Code: 22CSP-367**

1. **Aim:** Simulate a cloud scenario using Matlab and run an algorithm for temperature variations.

2. **Objective:** To simulate a cloud computing scenario using MATLAB and implement an algorithm to monitor and analyze temperature variations.

3. **Script and Output:**

- **Script**

```
channelID = 2819555; % Replace with your ThingSpeak Channel ID
writeAPIKey = '78UMR9Q3LLHGNGQX'; % Replace with your ThingSpeak Write
API Key

% Parameters
time = 0:0.1:24; % Time in hours (0 to 24, with 0.1-hour intervals)
baseTemp = 20; % Base temperature in degrees Celsius
amplitude = 10; % Temperature fluctuation amplitude
noiseFactor = 2; % Random noise amplitude
baseHumidity = 50; % Base humidity in percentage
humidityAmplitude = 10; % Humidity fluctuation amplitude

% Simulating temperature variations
temperature = baseTemp + amplitude * sin((pi/12) * time) + noiseFactor *
randn(size(time));

% Simulating humidity variations (using sine wave for fluctuation)
humidity = baseHumidity + humidityAmplitude * sin((pi/12) * time) + noiseFactor *
randn(size(time));

% Plotting the temperature variations
```

```
figure;

% Plot Temperature
subplot(2,1,1); % Subplot for temperature
plot(time, temperature, 'b', 'LineWidth', 1.5);
xlabel('Time (hours)');
ylabel('Temperature (°C)');
title('Simulated Cloud Temperature Variations');
grid on;
% Plot Humidity
subplot(2,1,2); % Subplot for humidity
plot(time, humidity, 'g', 'LineWidth', 1.5);
xlabel('Time (hours)');
ylabel('Humidity (%)');
title('Simulated Cloud Humidity Variations');
grid on;

% Running an algorithm to detect significant temperature changes (spike detection)
threshold = 5; % Change threshold for spikes in temperature
tempDiff = diff(temperature); % Calculate differences
spikeIndices = find(abs(tempDiff) > threshold); % Detect significant spikes

% Mark spikes on the temperature plot
subplot(2,1,1);
hold on;
plot(time(spikeIndices), temperature(spikeIndices), 'ro', 'MarkerSize', 8, 'LineWidth',
1.5);
legend('Temperature', 'Detected Spikes');
% Output detected spikes information
disp('Detected spikes at the following times (hours) and temperatures (°C):');
disp([time(spikeIndices)', temperature(spikeIndices)']);
% Sending data to ThingSpeak
for i = 1:length(time)
% Write the temperature and humidity to ThingSpeak (two fields)
   response = thingSpeakWrite(channelID, [temperature(i), humidity(i)], 'WriteKey',
writeAPIKey);
   % Display the data being sent
```

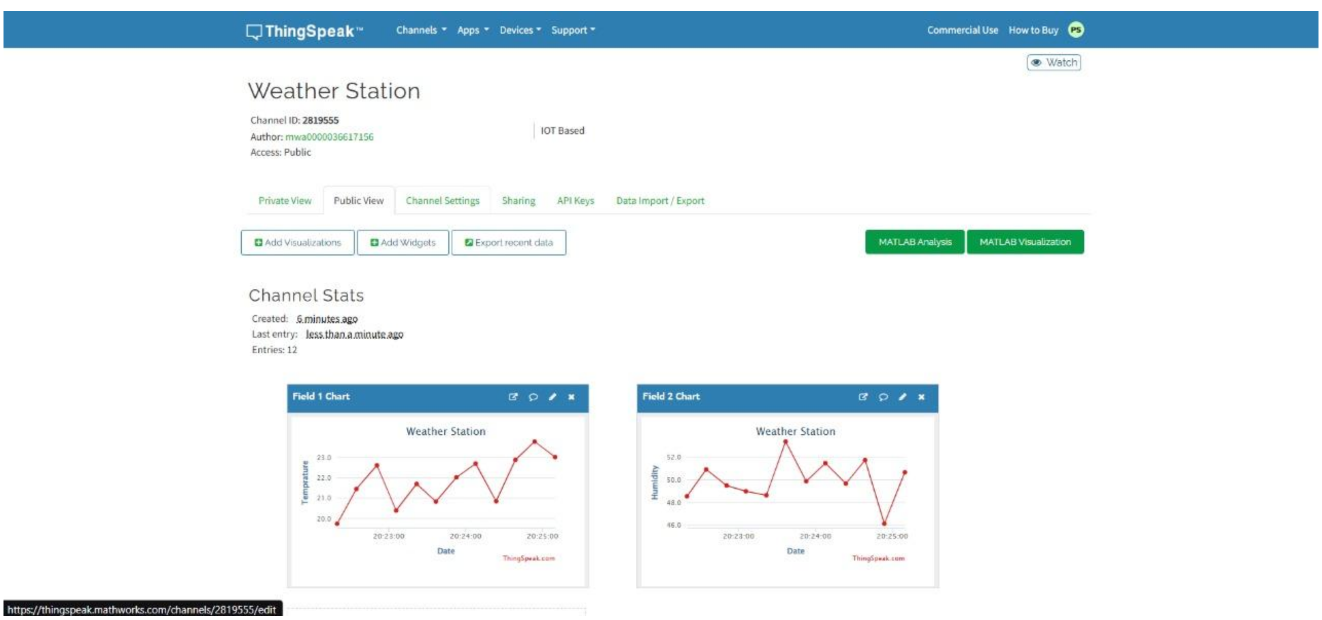
```
    fprintf('Time: %.1f hours, Temperature: %.2f°C, Humidity: %.2f%%\n', time(i),
temperature(i), humidity(i));
    % Optional: Add a delay to respect ThingSpeak rate limits (1 update per 15 seconds)
    pause(15);
end
```

- **Output:**

```
Command Window
disp('All data has been sent to ThingSpeak.');
Detected spikes at the following times (hours) and temperatures (°C):
    1.5000   20.3271
    2.4000   26.0812
    2.5000   31.6623
    3.9000   23.8685
    4.0000   30.4641
    8.6000   30.6032
    8.7000   24.3950
   17.7000    7.7330
   17.8000   14.0624
   21.7000   12.5506

Time: 0.0 hours, Temperature: 19.74°C, Humidity: 48.54%
Time: 0.1 hours, Temperature: 21.45°C, Humidity: 50.92%
Time: 0.2 hours, Temperature: 22.62°C, Humidity: 49.49%
Time: 0.3 hours, Temperature: 20.39°C, Humidity: 48.99%
Time: 0.4 hours, Temperature: 21.70°C, Humidity: 48.64%
Time: 0.5 hours, Temperature: 20.83°C, Humidity: 53.38%
Time: 0.6 hours, Temperature: 22.02°C, Humidity: 49.87%
Time: 0.7 hours, Temperature: 22.70°C, Humidity: 51.48%
Time: 0.8 hours, Temperature: 20.85°C, Humidity: 49.66%
Time: 0.9 hours, Temperature: 22.88°C, Humidity: 51.74%
Time: 1.0 hours, Temperature: 23.79°C, Humidity: 46.12%
Time: 1.1 hours, Temperature: 23.02°C, Humidity: 50.67%
Time: 1.2 hours, Temperature: 26.55°C, Humidity: 50.24%
Time: 1.3 hours, Temperature: 22.12°C, Humidity: 51.31%
Time: 1.4 hours, Temperature: 22.11°C, Humidity: 53.16%
Time: 1.5 hours, Temperature: 20.33°C, Humidity: 53.18%
Time: 1.6 hours, Temperature: 25.89°C, Humidity: 57.96%
% ThingSpeak Configuration
channelID = 2819555; % Replace with your ThingSpeak Channel ID
writeAPIKey = '78UMR9Q3LLHGNGQX'; % Replace with your ThingSpeak Write API Key
```

4. Results:



5. Conclusion Analysis:

- Simulated Data: Temperature and humidity data are simulated using sine waves with added noise to mimic real-world fluctuations over 24 hours.
- Plotting: Two separate plots show temperature and humidity variations, providing a visual comparison of their fluctuations over time.
- Spike Detection: A threshold-based algorithm detects and marks significant temperature changes, helping identify sudden spikes in the data.
- Data Transmission: Temperature and humidity data are sent to ThingSpeak in real-time, respecting the platform's rate limits, enabling cloud-based monitoring.

6. Learning Outcome:

- Learned how to simulate realistic environmental data (temperature and humidity) using mathematical functions and random noise to model real-world fluctuations.
- Gained experience in plotting multiple data sets (temperature and humidity) in MATLAB using subplots, improving understanding of data visualization techniques.
- Gained hands-on experience in sending real-time data to ThingSpeak, learning how to use API keys and respect platform rate limits for cloud-based data storage and monitoring.