

Lab 5

ICS423 - Internet of Things

Jayant Kolapkar - 2021BCS0132

Question

Task 1 – Explore Thingspeak – IoTCloud platform

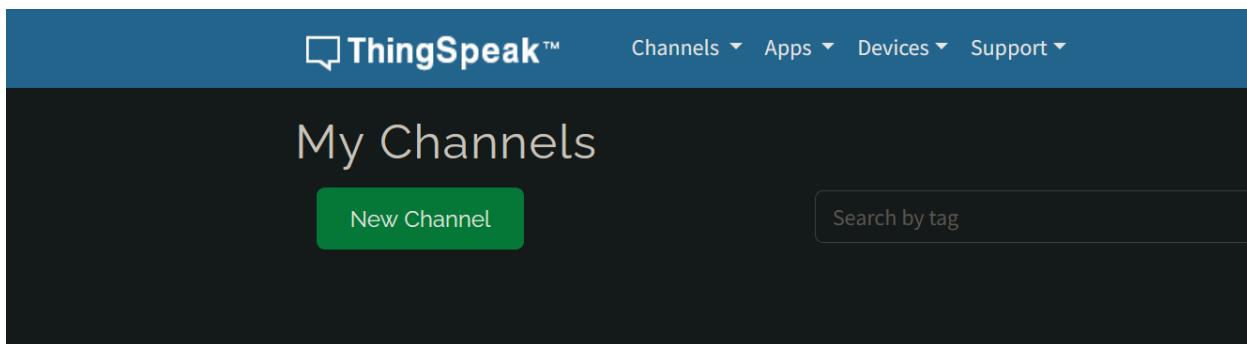
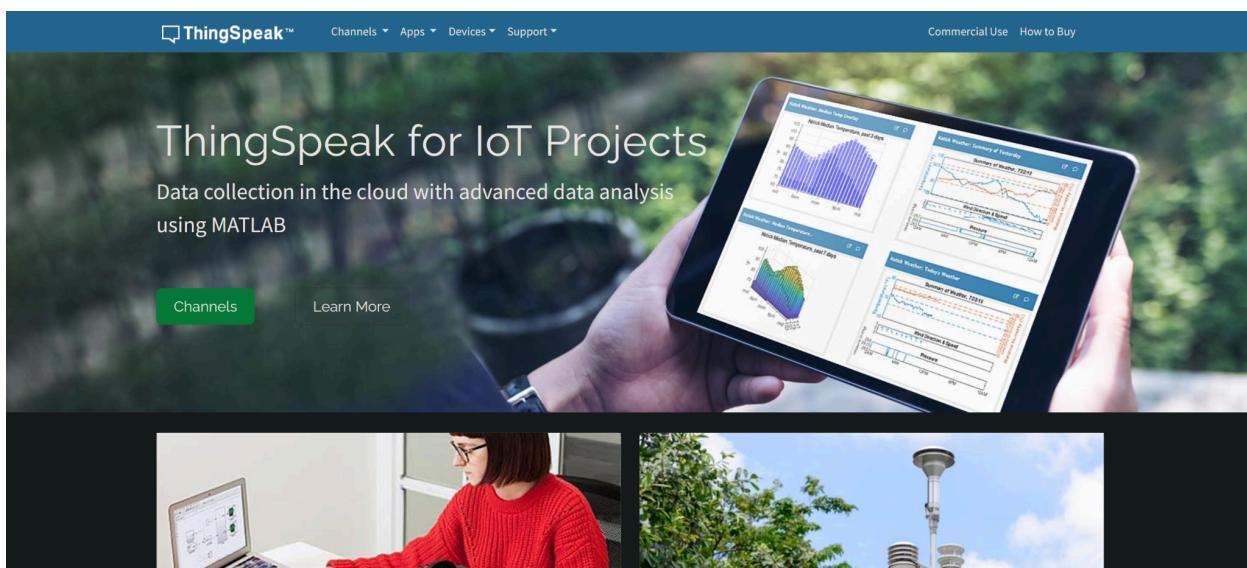
Task 2 – Design a temperature sensor circuit using tinkercad and periodically submit the sensor values to Thingspeak.

Task 3 – Demonstrate ThingSpeak dashboard and plots at realtime. Also, explore ThingReact concept in detail for sending alerts.

Task 1

Introduction to ThingSpeak - IoT Cloud Platform

1. **Cloud-Based IoT Analytics** – ThingSpeak is an open-source IoT platform designed for real-time collection, analysis, and visualization of sensor data.
2. **MATLAB Integration** – It seamlessly integrates with MATLAB, enabling advanced data processing, machine learning, and predictive analytics.
3. **RESTful API Support** – ThingSpeak offers RESTful APIs for reading and writing data, allowing easy connectivity with IoT devices, web applications, and cloud services.
4. **Data Storage & Visualization** – Users can store sensor data in channels and visualize it through built-in graphs and charts for deeper insights.
5. **Triggers & Alerts** – It supports event-driven automation, allowing users to trigger actions such as notifications or device control based on sensor readings.



The screenshot shows the ThingSpeak web interface for creating a new channel. At the top, there's a blue header bar with the 'ThingSpeak™' logo and navigation links for 'Channels', 'Apps', 'Devices', and 'Support'. Below the header, the title 'New Channel' is displayed. The main form consists of several input fields: 'Name' (empty), 'Description' (empty), and seven 'Field' sections labeled 'Field 1' through 'Field 7'. Each field section contains a 'Field Label' (e.g., 'Field Label 1' for Field 1) and a checked checkbox.

Name	Field Label 1	Field 1
Description		
Field 1	Field Label 1	<input checked="" type="checkbox"/>
Field 2		<input type="checkbox"/>
Field 3		<input type="checkbox"/>
Field 4		<input type="checkbox"/>
Field 5		<input type="checkbox"/>
Field 6		<input type="checkbox"/>
Field 7		<input type="checkbox"/>

Task 2

Design a Temperature Sensor Circuit Using Tinkercad & Transmit Data to ThingSpeak

Algorithm:

1. **Initialize Serial Communication & Wi-Fi Connection**
 - Start serial communication at 115200 baud rate.
 - Send AT commands to verify ESP8266 response.
 - Connect to the Wi-Fi network using the **AT+CWJAP** command.
2. **Establish Connection with ThingSpeak**

- Use the **AT+CIPSTART** command to create a TCP connection to ThingSpeak's API server (api.thingspeak.com) on port 80.

3. Read Sensor Data

- Capture analog input from pin A0, corresponding to the temperature sensor.
- Convert the analog reading to a temperature value using the **map()** function.

4. Transmit Data to ThingSpeak

- Construct an HTTP GET request containing the API key and temperature value.
- Send the request using **AT+CIPSEND**, followed by the actual HTTP request.

5. Repeat Data Transmission

- Continuously send sensor readings within the **loop()** function at intervals of 100 milliseconds.

Code:

```

1 //Make sure to subscribe Technomekanics:
2 String ssid      = "Simulator Wifi"; // SSID to connect to
3 String password = ""; // Our virtual wifi has no password
4 String host      = "api.thingspeak.com"; // Open Weather Map API
5 const int httpPort = 80;
6 String url       = "/update?api_key=IQ4BKS2FGBIQS943&field1=";
7
8 int setupESP8266(void) {
9   // Start our ESP8266 Serial Communication
10  Serial.begin(115200); // Serial connection over USB to computer
11  Serial.println("AT"); // Serial connection on Tx / Rx port to ESP8266
12  delay(10); // Wait a little for the ESP to respond
13  if (!Serial.find("OK")) return 1;
14
15 // Connect to 123D Circuits Simulator Wifi
16 Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
17 delay(10); // Wait a little for the ESP to respond
18 if (!Serial.find("OK")) return 2;
19
20 // Open TCP connection to the host:
21 Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\",," + httpPort);
22 delay(50); // Wait a little for the ESP to respond
23 if (!Serial.find("OK")) return 3;
24
25 return 0;
26 }
27
28 void anydata(void) {
29
30   int temp = map(analogRead(A0),20,358,-40,125);
31
32   // Construct our HTTP call
33   String httpPacket = "GET " + url + String(temp) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
34   int length = httpPacket.length();
35
36   // Send our message length
37   Serial.print("AT+CIPSEND=");
38   Serial.println(length);
39   delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

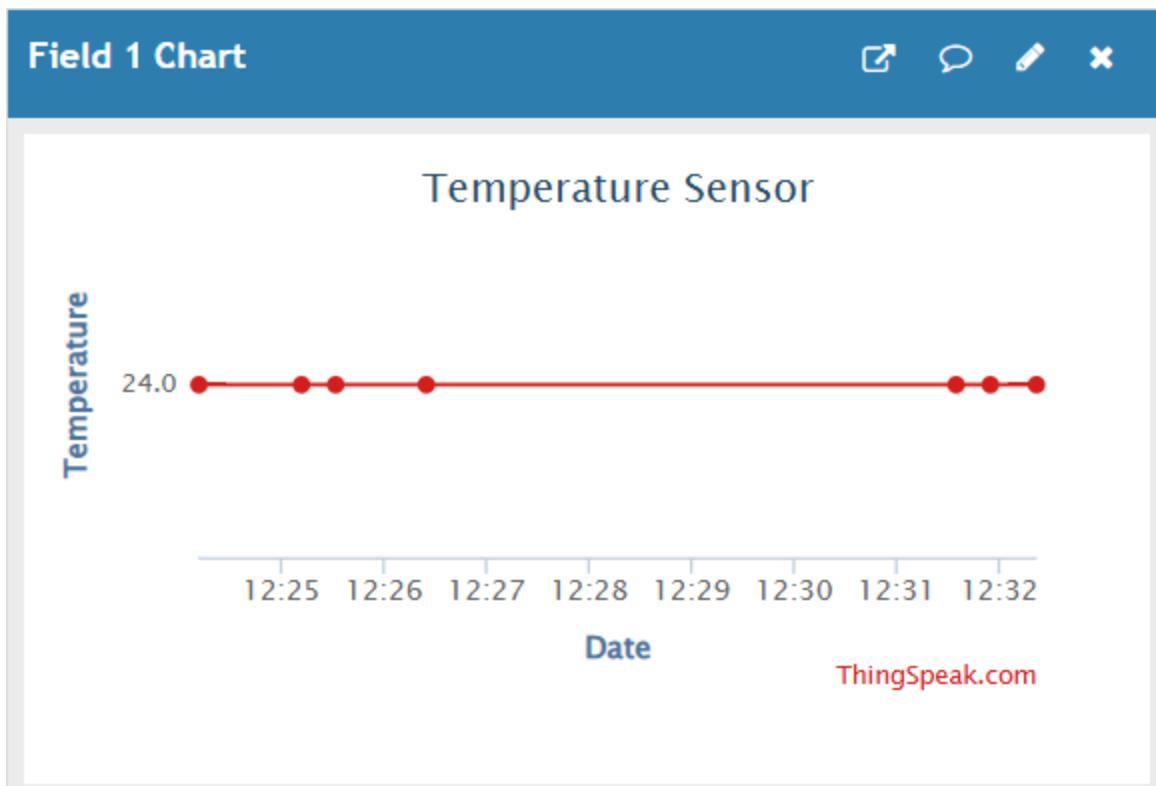
```

```

40 // Send our http request
41 Serial.print(httpPacket);
42 delay(10); // Wait a little for the ESP to respond
43 if (!Serial.find("SEND OK\r\n")) return;
44
45
46 }
47
48
49 void setup() {
50   setupESP8266();
51
52 }
53
54 void loop() {
55   anydata();
56   delay(100);
57 }
```

Output:

- Temperature data is periodically submitted to ThingSpeak for real-time monitoring.



Task 3

Real-Time Data Visualization & ThingReact Alerts

ThingSpeak enables real-time visualization of sensor data through customizable dashboards.

Steps to Create a ThingSpeak Channel:

1. **Create a New Channel** – Define a name and specify fields (e.g., Temperature, Humidity).
2. **Send Sensor Data** – Use an ESP8266, ESP32, or Arduino to transmit data using an API key.
3. **Visualize Data** – Access the “Private View” or “Public View” tab to observe live graphs.
4. **Customize Widgets** – Utilize widgets like line charts, gauges, and histograms to analyze trends.

The screenshot shows the ThingSpeak interface. At the top, there's a blue header bar with the 'ThingSpeak™' logo and navigation links for 'Channels', 'Apps', 'Devices', and 'Support'. Below the header, the main title 'My Channels' is displayed. A green button labeled 'New Channel' is visible. To the right is a search bar with a placeholder 'Search by tag' and a magnifying glass icon. The main content area is a table titled 'Name' with one row containing 'Temperature Sensor'. To the right of the table are columns for 'Created' (2025-02-06) and 'Updated' (2025-02-06 07:25). At the bottom of the table row, there are buttons for 'Private', 'Public', 'Settings', 'Sharing', 'API Keys', and 'Data Import / Export'.

Name	Created	Updated
Temperature Sensor	2025-02-06	2025-02-06 07:25

ThingReact – Event-Based Trigger System

ThingReact enables automated responses based on sensor data thresholds.

How ThingReact Works:

- Continuously monitors real-time data from a ThingSpeak channel.
- Triggers a response when a predefined condition is met (e.g., temperature surpassing 50°C).
- Sends notifications via email, tweets, or HTTP requests to external services like IFTTT or Telegram.

Steps to Use ThingReact:

1. Navigate to **Apps → ThingReact** on ThingSpeak.
2. Click **Create New Reaction** and set conditions:
 - **Example Condition:** If Temperature (Field1) starts with "2".
 - **Example Action:** Send an email alert or trigger a webhook.
3. Select an Action Type (HTTP Request, Twitter Post, Email Notification).
4. Enable the Reaction and monitor real-time alerts.

Practical Use Cases:

- **Smart Agriculture:** Send SMS alerts when soil moisture drops below a threshold.
- **Industrial Safety:** Trigger an alarm if gas levels exceed safe limits.
- **Home Automation:** Automatically switch on a fan when room temperature crosses a set value.

Name:	React 1
Condition Type:	String
Test Frequency:	On data insertion
Last Ran:	2025-02-06 07:25
Channel:	Temperature Sensor
Condition:	Field 1 (Temperature) starts with "2"
ThingHTTP:	Request 220236
Run:	Each time the condition is met
Created:	2025-02-06 7:24 am

