# PRACTICAL-9

## AIM

Implement mini project and connect with IBM Bluemix & Thingspeak for data collection on cloud and plot the graph of it. Plotting data on thingspeak.com. Take analog input from ESP and pass that data to api.thingspeak.com and prepare 2 online graph.

## THEORY

**Bluemix:**

- Bluemix is the IBM open cloud platform that provides mobile and web developers access to IBM software for integration security transaction and other key functions as well as software from business partners.
- Built on Cloud Foundry open source technology Bluemix provides pre-built Mobile Backend as a Service (MBaaS) capabilities. Bluemix offers more control to application developers by using its Platform as a Service (PaaS) offering. The goal is to simplify the delivery of an application by providing services that are ready for immediate use and hosting capabilities to enable internal scale development.
- With the broad set of services and runtimes in Bluemix the developer gains control and flexibility and has access to various data options from predictive analytics to big data.
- Bluemix provides the following features:
    - A range of services that enable you to build and extend web and mobile apps fast
    - Processing power for you to deliver app changes continuously
    - Fit-for-purpose programming models and services
    - Manageability of services and applications
    - Optimized and elastic workloads
    - Continuous availability

**Thingspeak:**

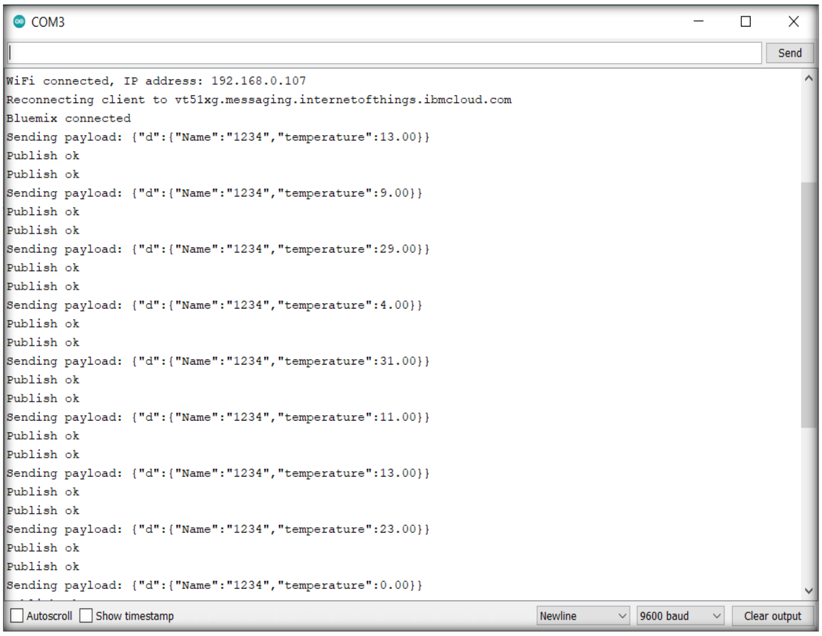
- ThingSpeak allows you to aggregate, visualize and analyze live data streams in the cloud. Some of the key capabilities of ThingSpeak include the ability to:
    - Easily configure devices to send data to ThingSpeak using popular IoT protocols.
    - Visualize your sensor data in real-time.
    - Aggregate data on-demand from third-party sources.
    - Use the power of MATLAB to make sense of your IoT data.
    - Run your IoT analytics automatically based on schedules or events.
    - Prototype and build IoT systems without setting up servers or developing web software.
    - Automatically act on your data and communicate using third-party services like Twilio or Twitter

# PRACTICAL

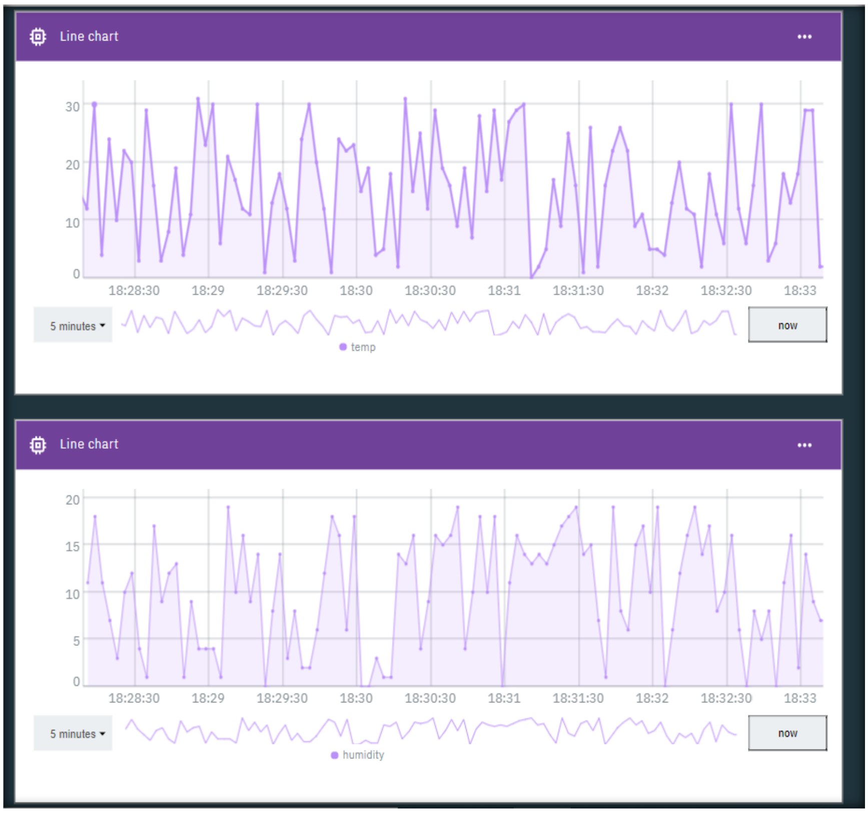
**Bluemix:**

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include "DHT.h"

const char* ssid = "speedzone";
const char* password = "1234567890";
#define DHTPIN 4
#define DHTTYPE DHT11
#define ORG "r9xhnn*"
#define DEVICE_TYPE "Node_MCU"
#define DEVICE_ID "1234"
#define TOKEN "sXo4VQIlswL&KUtnbV"

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char pubTopic1[] = "iot-2/evt/status1/fmt/json";
char pubTopic2[] = "iot-2/evt/status2/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);
DHT dht(DHTPIN, DHTTYPE);


void setup() {
   Serial.begin(9600);
   dht.begin();
   Serial.println();
   Serial.print("Connecting to ");
   Serial.print(ssid);
   WiFi.begin(ssid, password);
   while (WiFi.status() != WL_CONNECTED) {
     delay(500);
     Serial.print(".");
   }
   Serial.println("");

   Serial.print("WiFi connected, IP address: ");
   Serial.println(WiFi.localIP());

   if (!client.connected()) {
      Serial.print("Reconnecting client to ");
```

```
        Serial.println(server);
        while (!client.connect(clientId, authMethod, token)) {
            Serial.print(".");
            delay(500);
        }
        Serial.println("Bluemix connected");
    }
}

long lastMsg = 0;
void loop() {
    client.loop();
    long now = millis();
    if (now - lastMsg > 3000) {
        lastMsg = now;
        float humidity = dht.readHumidity();
        float temperature = dht.readTemperature();
        String payload = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
              payload += ",\"temperature\":";
              payload += temperature;
              payload += "}}";

        Serial.print("Sending payload: ");
        Serial.println(payload);

        if (client.publish(pubTopic1, (char*) payload.c_str())) {
            Serial.println("Publish ok");
        } else {
            Serial.println("Publish failed");
        }
        String payload1 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
              payload1 += ",\"humidity\":";
              payload1 += humidity;
              payload1 += "}}";

         if (client.publish(pubTopic2, (char*) payload1.c_str())) {
            Serial.println("Publish ok");
        } else {
            Serial.println("Publish failed");
        }
    }
}
```



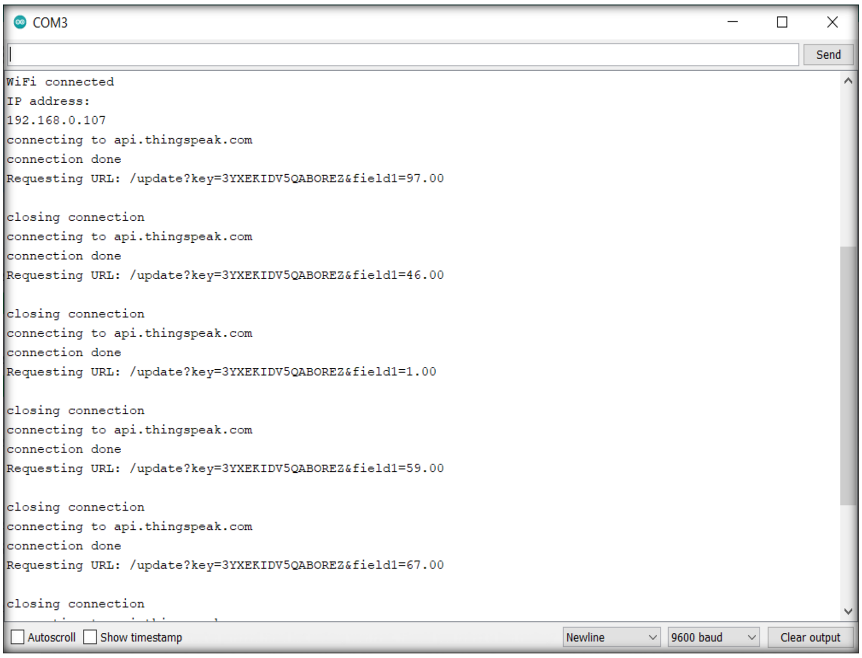
```
WiFi connected, IP address: 192.168.0.107
Reconnecting client to vt51xg.messaging.internetofthings.ibmcloud.com
Bluemix connected
Sending payload: {"d":{"Name":"1234","temperature":13.00}}
Publish ok
Publish ok
Sending payload: {"d":{"Name":"1234","temperature":9.00}}
Publish ok
Publish ok
Sending payload: {"d":{"Name":"1234","temperature":29.00}}
Publish ok
Publish ok
Sending payload: {"d":{"Name":"1234","temperature":4.00}}
Publish ok
Publish ok
Sending payload: {"d":{"Name":"1234","temperature":31.00}}
Publish ok
Publish ok
Sending payload: {"d":{"Name":"1234","temperature":11.00}}
Publish ok
Publish ok
Sending payload: {"d":{"Name":"1234","temperature":13.00}}
Publish ok
Publish ok
Sending payload: {"d":{"Name":"1234","temperature":23.00}}
Publish ok
Publish ok
Sending payload: {"d":{"Name":"1234","temperature":0.00}}
```



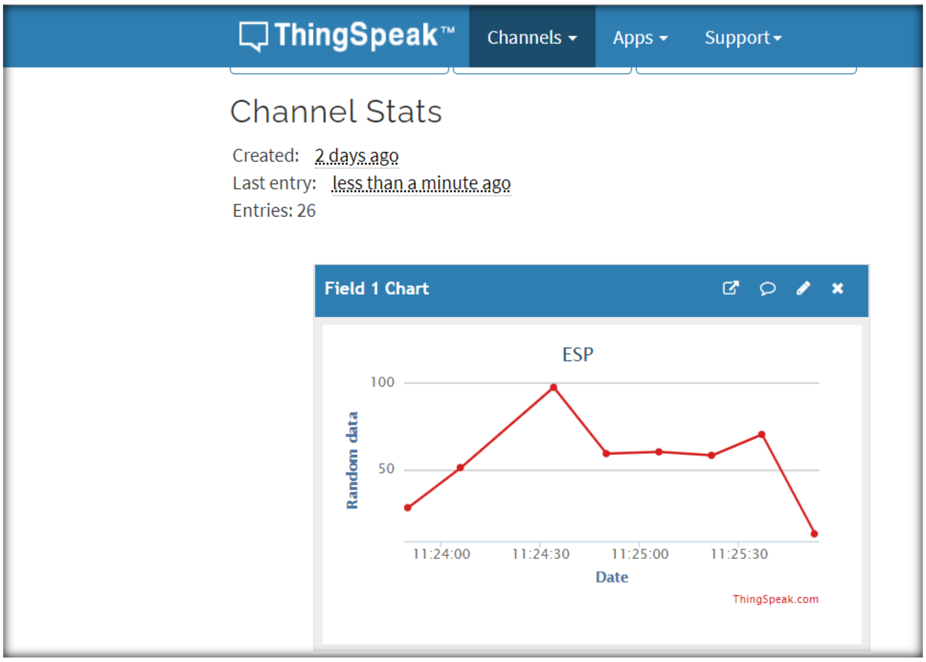
**Thingspeak:**

```
#include <ESP8266WiFi.h>

#define SENSOR A0

const char* ssid     = "Rudrabarad"; //WiFiSSID
const char* password = "Rab@0412";  //PASSWORD

const char* host = "api.thingspeak.com";
// Data URL: https://thingspeak.com/channels/XXXXX/
const char* privateKey = "3YXEKIDV5QABOREZ"; //ENTER_YOUR_PRIVATE_KEY

void setup() {
  Serial.begin(9600);
  delay(10);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

float value = 0;

void loop() {
  delay(5000);
  value = analogRead(A0);

  Serial.print("connecting to ");
  Serial.println(host);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
```

```
 Serial.println("connection done");
 // We now create a URI for the request
 String url = "/update?";
 url += "key=";
 url += privateKey;
 url += "&field1=";
 url += value;


 Serial.print("Requesting URL: ");
 Serial.println(url);

 // This will send the request to the server
 client.print(String("GET ") + url + " HTTP/1.1\r\n" +
              "Host: " + host + "\r\n" +
              "Connection: close\r\n\r\n");
 delay(10);

 // Read all the lines of the reply from server and print them to Serial
 while (client.available()) {
   String line = client.readStringUntil('\r');
   Serial.print(line);
 }

 Serial.println();
 Serial.println("closing connection");
}
```



```
WiFi connected
IP address:
192.168.0.107
connecting to api.thingspeak.com
connection done
Requesting URL: /update?key=3YXEKIDV5QABOREZ&field1=97.00

closing connection
connecting to api.thingspeak.com
connection done
Requesting URL: /update?key=3YXEKIDV5QABOREZ&field1=46.00

closing connection
connecting to api.thingspeak.com
connection done
Requesting URL: /update?key=3YXEKIDV5QABOREZ&field1=1.00

closing connection
connecting to api.thingspeak.com
connection done
Requesting URL: /update?key=3YXEKIDV5QABOREZ&field1=59.00

closing connection
connecting to api.thingspeak.com
connection done
Requesting URL: /update?key=3YXEKIDV5QABOREZ&field1=67.00

closing connection
```



Channel Stats

Created: 2 days ago

Last entry: less than a minute ago

Entries: 26

## CONCLUSION

In this practical, we learned about IBM bluemix and Thingspeak platforms. We uploaded data to both the platforms and plotted the graph.