Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 6**	
Title of Lab Assignment: IOT with Cloud: Interfacing IOT device with cloud.		
DOP: 13-02-2024	**DOS: 20-02-2024**	
CO Mapped: **CO3, CO4**	**PO Mapped:** **PO1, PO2, PO5, PO7, PSO1**	**Signature:**

Practical No. 6

Aim: IOT with Cloud : Interfacing IOT device with cloud.

Theory:

Internet-of-Things can benefit from the scalability, performance and pay-as you-go nature of cloud computing infrastructures. Indeed, as IoT applications produce large volumes of data and comprise multiple computational components (e.g., data processing and analytics algorithms), their integration with cloud computing infrastructures could provide them with opportunities for cost-effective on-demand scaling.

An IoT cloud is a massive network that supports IoT devices and applications. This includes the underlying infrastructure, servers and storage, needed for real-time operations and processing.

IoT/cloud infrastructures and related services can be classified to the following models:

1. **Infrastructure-as-a-Service (IaaS) IoT/Clouds:**
   These services provide the means for accessing sensors and actuators in the cloud. The associated business model involves the IoT/Cloud provider to act either as data or sensor provider.

2. **Platform-as-a-Service (PaaS) IoT/Clouds:**
   This is the most widespread model for IoT/cloud services, given that it is the model provided by all publicIoT/cloud infrastructures outlined above.

3. **Software-as-a-Service (SaaS) IoT/Clouds:**
   SaaS IoT services are the ones enabling their uses to access complete IoT-based software applications through the cloud, on- demand and in a pay-as-you-go fashion. As soon as sensors and IoT devices are not visible, SaaS IoT applications resemble very much conventional cloud-based SaaS applications.

ThingsSpeak:

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB code in ThingSpeak

you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics.

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.

Code:

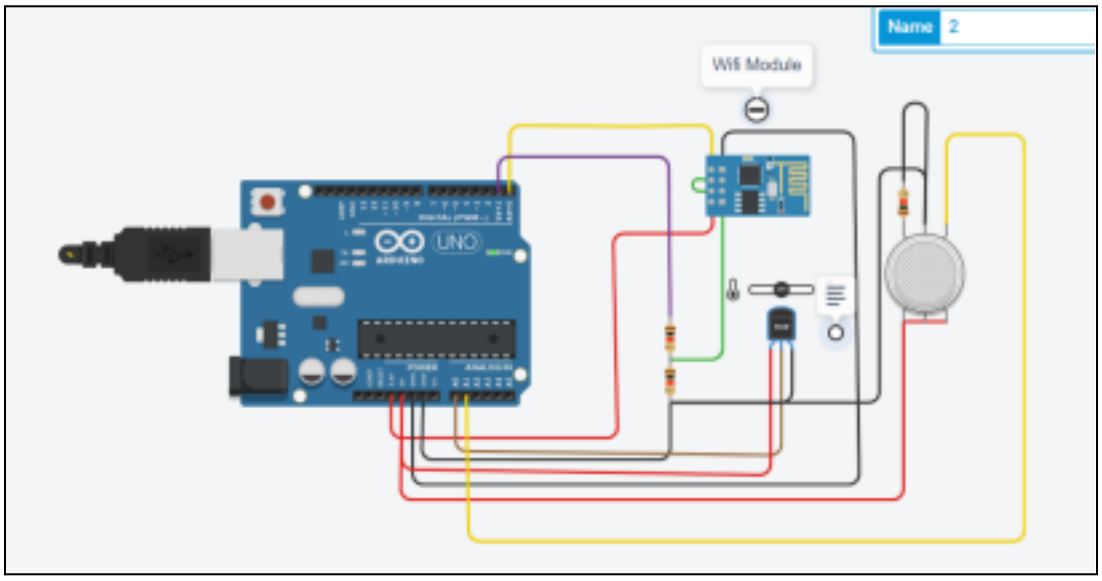
```
float val, voltage, temp;
String ssid = "Simulator Wifi"; // SSID to connect to
String password = ""; //virtual wifi has no password
String host = "api.thingspeak.com"; // Open Weather Map API
const int httpPort = 80;
String url = "/update?api_key=ZXP27Y8OGRHR7ISL&field1="; //Replace XXXXXXXXXXXXXXXX by your ThingSpeak Channel API Key

void setupESP8266(void) {
  // Start our ESP8266 Serial Communication
  Serial.begin(115200); // Serial connection over USB to computer
   Serial.println("AT"); // Serial connection on Tx / Rx port to ESP8266 delay(10); // Wait a little for the ESP to respond
  if (Serial.find("OK"))
     Serial.println("ESP8266 OK!!!");
     // Connect to Simulator Wifi
     Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
     delay(10); // Wait a little for the ESP to respond
  if (Serial.find("OK"))
     Serial.println("Connected to WiFi!!!");
```

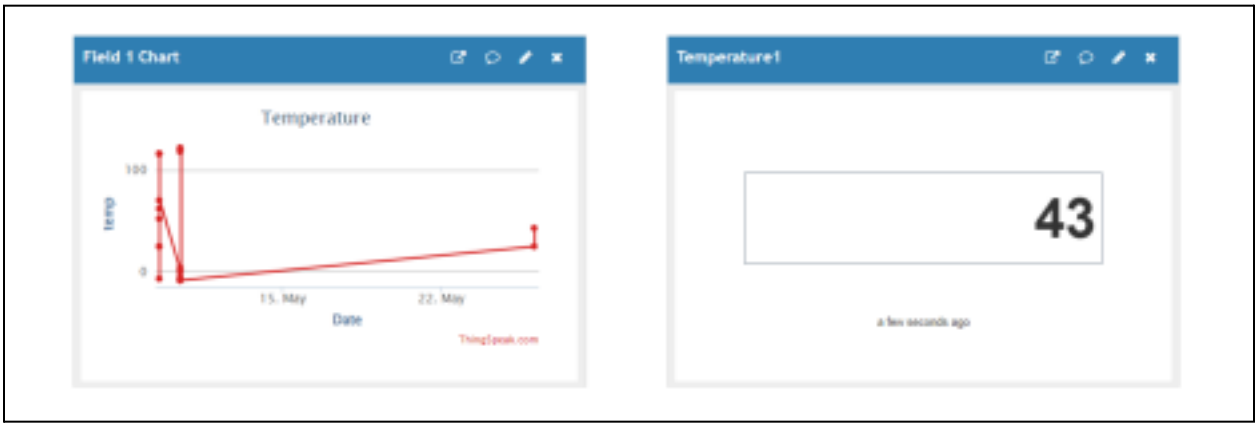
```
      // Open TCP connection to the host:
      //ESP8266 connects to the server as a TCP client.
       Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort); delay(50); // Wait a little
for the ESP to respond
   if (Serial.find("OK"))
      Serial.println("ESP8266 Connected to server!!!") ;
}

void anydata(void) {
   //val = analogRead(A1);
   val=analogRead(A0);
   voltage=val*0.0048828125;
   temp = (voltage - 0.5) * 100.0;
   // Construct our HTTP call
   String httpPacket = "GET " + url + String(temp) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
   int length = httpPacket.length();
   // Send our message length
   Serial.print("AT+CIPSEND=");
   Serial.println(length);
   delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;
   // Send our http request
   Serial.print(httpPacket);
   delay(10); // Wait a little for the ESP to respond
   if (Serial.find("SEND OK\r\n"))
      Serial.println("ESP8266 sends data to the server");
   }

void setup() {
   pinMode(A0, INPUT);
   //pinMode(A1,INPUT);
   setupESP8266();
}
```

```
void loop() {
   anydata();
   delay(4000); // delay changed for faster analytics
}
```

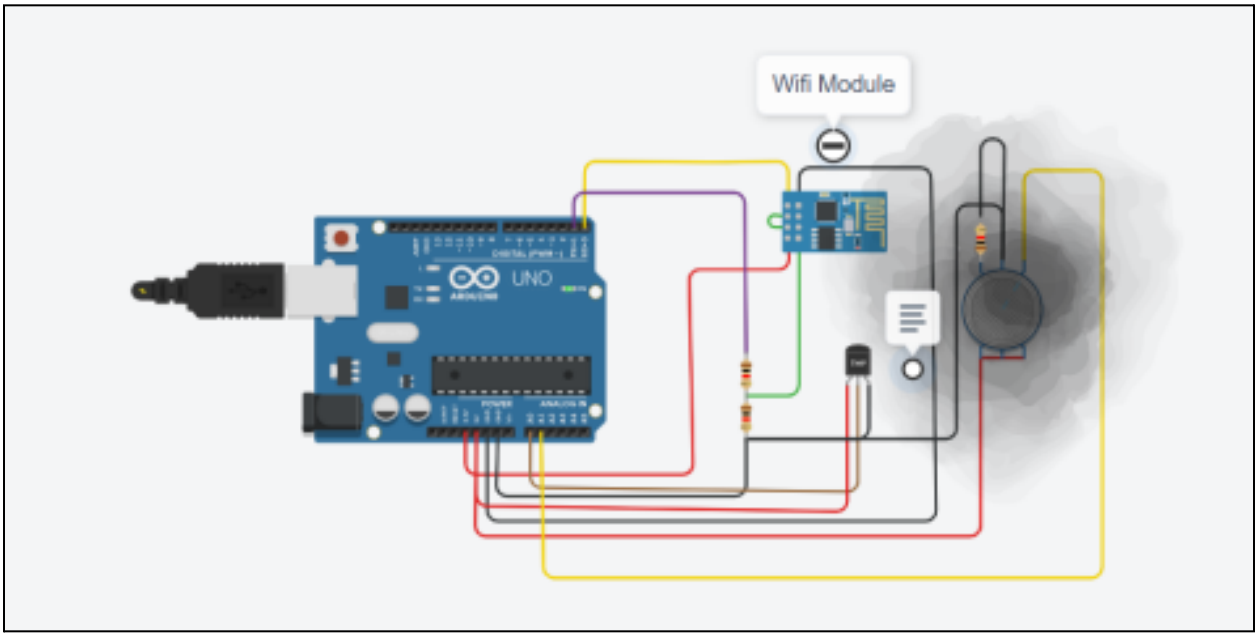
Output:

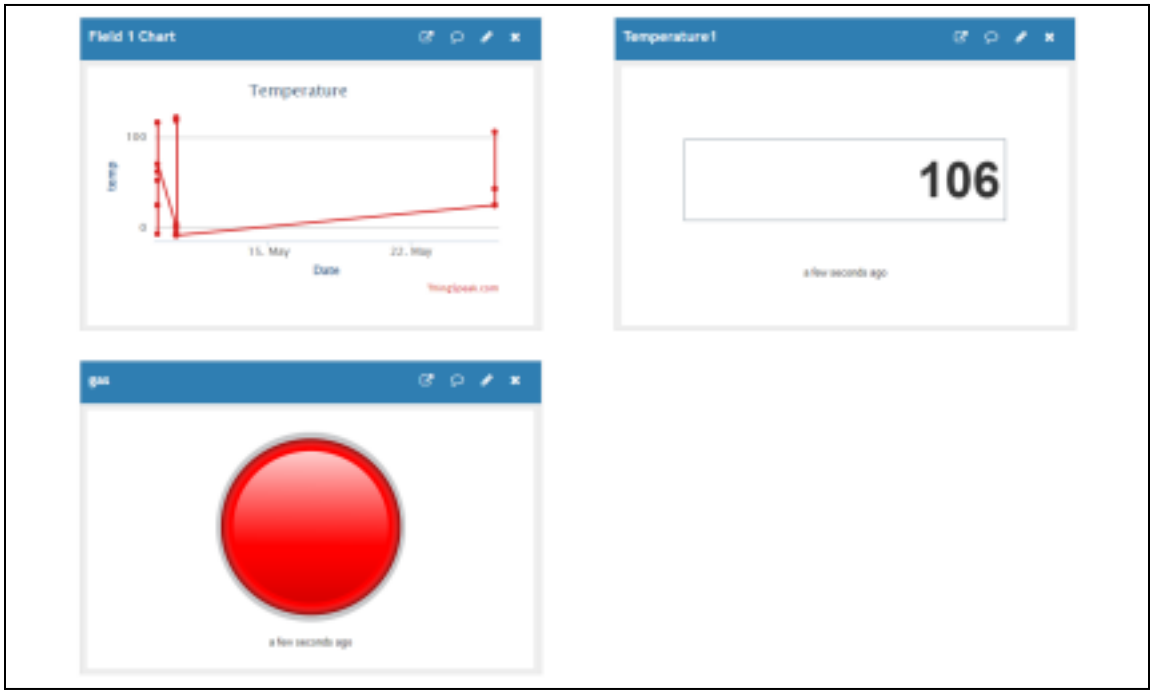
For Temperature Sensor:





For Gas Sensor:





Conclusion:

Hence, we have successfully implemented IoT devices with Cloud.