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.

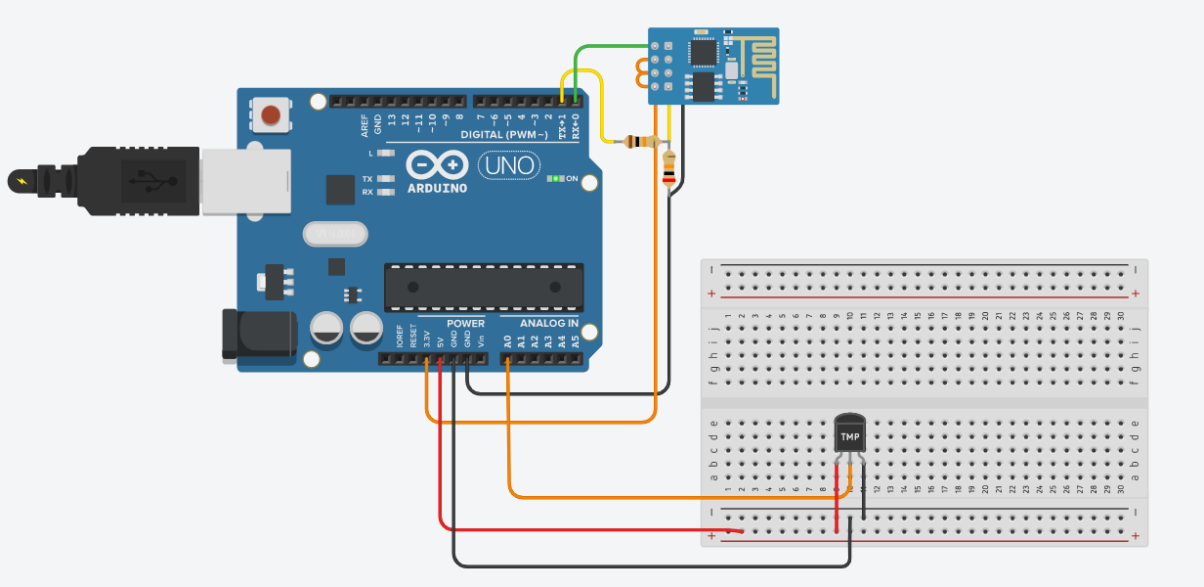
IoT/cloud infrastructures and related services can be classified to the following models:

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

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 public IoT/cloud infrastructures outlined above.

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.

CIRCUIT DIAGRAM:



SOURCE CODE:

```
 String ssid     = "Simulator Wifi";
 String password = "";
 String host     = "api.thingspeak.com";
 const int httpPort   = 80;
 String uri     = "/update?api_key=88A5L9UK3R8ME7MS&field1=";


 void setup() {

 // 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



 // Connect to Virtual WiFi IoTFDP
 // AT+CWJAP="SSID","password"
 // AT+CWJAP=\"SSID\",\"password\"\r\n
 Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
 delay(10);       // Wait a little for the ESP to respond



 // Open TCP connection to the host:
 // AT+CIPSTART="TCP","192.168.3.116",8080
 Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);
 delay(50);       // Wait a little for the ESP to respond


}

void loop() {

 int temp = map(analogRead(A0),20,358,-40,125);
 int light = analogRead(A1);
 Serial.println("Temperature " +temp);
 Serial.println("Light " +light);

 // Construct our HTTP call
 // GET Method to send data to thingspeak
 String httpPacket = "GET " + uri + String(temp) + "&field2=" + String(light)+ " 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);
```

```
 // Send our http request
 Serial.print(httpPacket);
 delay(10);


 delay(1000);
}
```

OUTPUT:

```
AT+CIPSEND=96
GET /update?api_key=88A5L9UK3R8ME7MS&field1=24&field2=430 HTTP/1.1
Host: api.thingspeak.com
```

CONCLUSION:

From this practical, I have learned and understood the implementation of IoT with Cloud.