```
#include <time.h>
 #include <WiFi.h>
 #include <PubSubClient.h>

 #define ORG “wt19pm”
 #define DEVICE_TYPE “NodeMCU”
 #define DEVICE_ID “12345”
 #define TOKEN “12345678”

 Char server[] = ORG “.messaging.internetofthings.ibmcloud.com”;
 Char publishTopic[] = “iot-2/evt/data/fmt/json”;
 Char authMethod[] = “use-token-auth”;
 Char token[] = TOKEN;
 Char clientId[] = “d:” ORG “:” DEVICE_TYPE “:” DEVICE_ID;

 WiFiClient wifiClient;
 PubSubClient client(server, 1883, wifiClient);

 Float temperature  = 0;
 Int gas = 0;
 Int flame = 0;

 String flame_status = “”;
 String Gas_status = “”;
 String exhaust_fan_status = “”;
 String sprinkler_status = “”;

 Void setup() {
```

```
  Serial.begin(99900);
   wifiConnect();
   mqttConnect();
}

Void loop() {

  Srand(time(0));

    //initial variables and random generated data

    Temperature = random(-20,125);
    Gas = random(0,1000);
    Int flamereading = random(200,1024);
    Flame = map(flamereading,200,1024,0,2);

    //set a flame status

    Switch (flame) {
    Case 0:
       Flame_status = “No Fire”;
       Break;
    Case 1:
       Flame_status = “Fire is Detected”;
       Break;
    }

    //send the sprinkler status
```

```
If(flame==1){
   Sprinkler_status = “Working”;
}
Else{
   Sprinkler_status = “Not Working”;

}

//toggle the fan according to gas reading

If(gas > 100){
   Gas_status = “Gas Leakage is Detected”;
   Exhaust_fan_status = “Working”;

}
Else{
   Gas_status = “No Gas Leakage is Detected”;
   Exhaust_fan_status = “Not Working”;
}

//json format for IBM Watson

String payload = “{“;
Payload+=”\”gas\”:”;
Payload+=gas;
Payload+=”,”;
Payload+=”\”temperature\”:”;
Payload+=(int)temperature;
Payload+=”,”;
```

```
    Payload+="\"flame\":";
    Payload+=flamereading;
    Payload+=",";
    Payload+="\"fire_status\":\""+flame_status+"\",";
    Payload+="\"sprinkler_status\":\""+sprinkler_status+"\",";
    Payload+="\"Gas_status\":\""+Gas_status+"\",";
    Payload+="\"exhaust_fan_status\":\""+exhaust_fan_status+"\"}";

    If(client.publish(publishTopic, (char*) payload.c_str()))
    {
       Serial.println("Publish OK");
    }
    Else{
       Serial.println("Publish failed");
    }
    Delay(1000);


    If (!client.loop())
    {
      mqttConnect();
    }
}


Void wifiConnect()
{
  Serial.print("Connecting to ");
  Serial.print("Wifi");
```

```
  WiFi.begin(“Wokwi-GUEST”, “”, 6);
  While (WiFi.status() != WL_CONNECTED)
  {
    Delay(500);
    Serial.print(“.”);
  }
  Serial.print(“WiFi connected, IP address: “);
  Serial.println(WiFi.localIP());

}



Void mqttConnect()
{
  If (!client.connected())
  {
    Serial.print(“Reconnecting MQTT client to “);
    Serial.println(server);
    While (!client.connect(clientId, authMethod, token))
    {
      Serial.print(“.”);
      Delay(500);
    }

    Serial.println();
  }
}
```