SOURCE CODE

Python Code for Temperature and Humidity Check :

```python
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

# Initialize GPIO

#Provide your IBM Watson Device Credentials

organization = "0vbvyp"

deviceType = "hazardous_monitoring"

deviceId = "hazard_report"

authMethod = "token"

authToken = "7jZ6JKfpj!Cq7tTO5M"

def myCommandCallback(cmd):

    print("Command received: %s" % cmd.data['command'])

    Status=cmd.data['command']

    if Status=="Alert":

        print("Alert")

        #print(cmd)


try:

        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}

        deviceCli = ibmiotf.device.Client(deviceOptions)

        #...........................................

except Exception as e:

        print("Caught exception connecting device: %s" % str(e))

        sys.exit()
```

```python
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times

deviceCli.connect()

while True:

    #Get Sensor Data from DHT11

    temp =random.randint(0,100)

    humid =random.randint(0,100)

    oxygen =random.randint(0,100)

    data = { 'temp' : temp, 'humidity': humid ,'oxygen': oxygen}

    data1 = { 'High temperature' : temp>60}

    #print data

    def myOnPublishCallback():

        print ("Published Temperature = %s C" % temp, "humidity = %s %%" % humid,"alert", "to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)


    if not success:

        print("Not connected to IoTF")

    time.sleep(1)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()
```