Source code

```
import
time
import sys
import
ibmiotf.app
lication
import
ibmiotf.dev
ice import
random
#Provide your IBM Watson
Device Credentials
organization = "vq4nsy"
deviceType =
"PNT2022TMID4465" deviceId
=
"PNT2022TMID4465DEVICEID"
authMethod = "token"
authToken = "rjghjHFTDHB!"
# Initialize GPIO
def myCommandCallback(cmd):
print("Command received: %s" %
cmd.data['command'])
status=cmd.data['command']    if
status=="alarmon":        print ("Alarm is
on")    elif (status == "alarmoff") :        print
("Alarm is off")    elif status ==
"sprinkleron":        print("Sprinkler is ON")
```

```
   elif status == "sprinklerOFF":

print("Sprinkler is OFF")

   #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()


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

deviceCli.con

nect()


while True:

      #Get Sensor Data from DHT11


      temp=random.randint(0,100)

Humid=random.randint(0,100)

gas=random.randint(0,100)


      data = { 'temp' : temp, 'Humid': Humid, 'gas' : gas }

      #print data

def

myOnPublishCall

back():

          print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,
"Gas_Level = %s %%" %gas, "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()
```