```
import timeimport sysimport ibmiotf.applicationimport ibmiotf.device# Provide your IBM Watson Device Credentialsorganization = "o9hvpe"  #  organization IDdeviceType = "controllerdata"  # device typedeviceId = "0001"  #  device idauthMethod = "token"authToken = "8072958226"  #  tokendef myCommandCallback(cmd):
print("Command received: %s" % cmd.data)    if cmd.data['command'] == 'motor_on':
print("MOTOR ON")    elif cmd.data['command'] == 'motor_off':        print("MOTOR OFF")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()deviceCli.connect()while True:    T = 59;    H = 51;    O = 2;    data = {'d':{'temperature': T, 'humidity': H,'objectTemp':O,}}    def myOnPublishCallback():        print("Published Temperature = %s C" % T, "Humidity = %s %%" % H, "objectTemp = %s C" %O ,"to IBM Watson")    success = deviceCli.publishEvent("event", "json", data, qos=0, on_publish=myOnPublishCallback)
if not success:        print("Not connected to IoTF")    time.sleep(1)
deviceCli.commandCallback = myCommandCallbackdeviceCli.disconnect()
```