

FINAL CODE:

```
import time

import sys

import ibmiotf.application
import ibmiotf.device

import random


organization="ckuumt"
devicetype="zxcv"
deviceid="0987"
authMethod="token"
authToken="1234567891"


def myCommandCallback(cmd):
    print("Command received:%s"%cmd.data['command'])
    status=cmd.data['command']
    if status == "lighton":
        print("led in on")
    else:
        print("led is 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:
    time.sleep(5)
    Ultrasonic=random.randint(0,80)
```

```

Weight=random.randint(0,100)
lat=round(random.uniform(11.03,11.50),6)
long=round(random.uniform(76.80,76.90),6)
GPS=str(lat)+str(',')+str(long)
myData={'Ultrasonic':Ultrasonic,'Weight':Weight,'GPS':GPS}
def myOnpublishCallback():
    print("Published Ultrasonic=%sCm"%Ultrasonic,"Weight:%s kg"%Weight,"GPS:%s"%GPS)

success=deviceCli.publishEvent("IoTSensor","json",data=myData,qos=0,on_publish=myOnpublishCallback)
if not success:
    print("Not connected to IoT")
    time.sleep(1)
    deviceCli.commandCallback=myCommandCallback
deviceCli.disconnect()

```

FINAL OUTPUT:





