```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials

organization = "8wd932"
deviceType = "Node_Mcu"
deviceId = "123456789"
authMethod = "token"
authToken = "123456789"

# Initialize GPIO

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()

#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


        time.sleep(5)
        ult_son=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)
        data = {'Ultrasonic' : ult_son, 'Weight' : weight , 'GPS' : gps}
        #print data
        def myOnPublishCallback():
            print ("Published Ultrasonic = %s Cm" %ult_son, "Weight:%s kg" %weight, "GPS: %s" %gps)
        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()
```