```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "vi4esk"
deviceType = "sudhan"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
# Initialize GPIO
def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="alarmon":
  print ("DUST BIN IS FULL")
 else:
  print ("Alarm 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.connect()
while True:
 #Get Sensor Data from DHT11


 latidude=random.randint(0,100)
 logditude=random.randint(0,100)


 data = { 'latidude' : latitude, 'logditude': logditude }
 #print data
 def myOnPublishCallback():
    print ("Published latitude = %s C" % latidude, "logditude = %s %%" % logditude, "to IBM Watson")
 success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
```

```
on_publish=myOnPublishCallback)
 if not success:
  print("Not connected to IoTF")
  time.sleep(10)

 deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```