

```

import time

import sys

import ibmiotf.application
import ibmiotf.device
import random
import os

from twilio.rest import Client
account_sid = 'AC5a226c4cfb911efa753ef6f8e486d27a'
auth_token = '494dd178c1bd36c06fa44301fca2b543'
client = Client(account_sid,auth_token)
#Provide your IBM Watson Device Credentials
organization = "rsu1tr"
deviceType = "sf"
deviceId = "smartfarm"
authMethod = "token"
authToken = "Q-1nP3j-JqTt4O7HyY"
# Initialize GPIO
def myCommandCallback(cmd): # function for Callback
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=='motoron':
        print("Turn Motor ON")

    elif status=='motoroff':
        print("Turn Motor 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 data
deviceCli.connect()
while True:
#Get Sensor Data from DHT11
Temperature=random.randint(0,100)
Humidity=random.randint(0,100)
SoilMoisture=random.randint(30,65)#(Value = 50-60)
ph=random.randint(0,10)#Ph value (6.2-6.8)
data = { 'Temperature' : Temperature, 'Humidity': Humidity,'SoilMoisture':SoilMoisture,'Ph':ph}
#print data
def myOnPublishCallback():

    print ("Published Temperature = %s C" % Temperature, "Humidity = %s %" %
Humidity,"SoilMoisture = %s %" % SoilMoisture,"Ph = %s %" % ph, "to IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
if not success:
print("Not connected to IoT")
    time.sleep(10)
if SoilMoisture==50:
print("Motor is ON")
    message = client.messages \
        .create(
from_='+18585440834',
body='Alert!!',
to = '+919498063191')
print(message.sid)
    else :
print("Motor is OFF")
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()

```