DEVELOP A PYTHON SCRIPT:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
from twilio.rest import Client


#Provide your IBM Watson Device Credentials
organization = "uyyqeq"
deviceType = "12345"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led is off")
    else :
        print ("please send proper command")

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
      pH = random.randint(1, 14)
```

```
        temp=random.randint(90,110)
        Humid=random.randint(60,100)

        data = {'pH': pH, 'temp' : temp, 'Humid': Humid }
        def SMS():
            message = Client.messages.create(
            body="ALERT!! THE WATER QUALITY IS DEGRADED",
            from_=keys.twilio_number,
            to = keys.target_number)
            print(message.body)
            if temperature>70 or pH<6 or Humidity>500:
                SMS()
        #print data
        def myOnPublishCallback():
            print ("Published pH= %s" % pH,"Published Temperature = %s C" % temp, "Humidity =
%s %%" % Humid, "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()
```