SOURCE CODE

PYTHON CODE TO IBM:

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

#Provide your IBM Watson Device Credentials
organization = " 60hw5g"
deviceType = "IOT"
deviceId = "ultrasonic"
authMethod = "token"
authToken =
"731719205001"


# 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

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

        data = { 'temp' : temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
            print ("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 clouddeviceCli.disconnect()
```

NODE RED CODE:

TEMPERATURE:

```
msg.payload=msg.pa

yload.”temp”return

msg;
```

HUMIDITY:

```
msg.payload=msg.payl

oad.”Humid”return

msg;
```