program:

import ibmiotf

from ibmiotf import device

import random

#Provide your IBM Watson Device Credentials

organization = "2n3nim"

deviceType = "abcd"

deviceId = "12345"

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 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

tmp=random.randint(30,120)

if(tmp>70):

gas=random.randint(0,300)

#flame=random.randint(0,1)

flame=1;

else :

gas=random.randint(0,300)

#flame=random.randint()

flame=0;

if(flame == 1):

sprinkler="On"

else:

sprinkler="Off"

if(gas>150):

exhaust="On"

else:

exhaust="Off"

data = { 'temp' : tmp, 'gas': gas, 'flame' : flame, 'sprinkler' : sprinkler, 'exhaust' : exhaust }

#print data

def myOnPublishCallback():

print ("Detected Temp = %s C" % tmp, "Gas = %s " % gas , "Flame = %s " % flame, "Sprinkler = %s " % sprinkler, "Exhaust = %s " % exhaust, "to IBM Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF,please connect to that!!!")

time.sleep(10)

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()

output:

