**Source Code**

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "rv07c6"

deviceType = "riverwaterquality-22\_23"

deviceId = "123456"

authMethod = "token"

authToken = "wQ\_)43L5c0@ku8)sgd"

# Initialize GPIO

def myCommandCallback(cmd):

    print("Command received: %s" % cmd.data['command'])

    status=cmd.data['command']

    if status=="lighton":

        print ("led is on")

    else :

        print ("led 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

        temp=random.randint(0,100)

        ph=random.randint(0,14)

        turb=random.randint(0,100)

        data = { 'temp' : temp, 'ph': ph,'turb' :turb }

        #print data

        def myOnPublishCallback():

            print ("Published Temperature = %s C" % temp, "ph = %s %%" % ph,"turbidity = %s NTU " % turb ,"to IBM Watson")

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

        if not success:

            print("Not connected to IoTF")

        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()