import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "kv09p4"

deviceType = "Groot"

deviceId = "13"

authMethod = "token"

authToken = "12345678"

global y

# Initialize GPIO

def myCommandCallback(cmd):

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

status=cmd.data['command']

if status=="motoron":

print ("motor is on")

if status=="motoroff" :

print ("motor is off")

if status=="manual" :

print ("Motor Control is in Manual Mode")

if status=="automatic" :

print ("Motor control is in Automatic Mode")

if soilmoisture > 600:

print ("motor is on")

#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)

Humid=random.randint(0,100)

soilmoisture=random.randint(0,1023)

Phlevel=random.randint(0,14)

y=soilmoisture

data = { 'temp' : temp, 'Humid': Humid,'soilmoisture' : soilmoisture ,'Phlevel' : Phlevel }

#print data

def myOnPublishCallback():

print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,"Soil Moisture is %s %%" % soilmoisture,"PH level is %s" %Phlevel ,"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()