Python Coding:

```
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
authMethod = "token"
organization = "yet4pm"
authToken = "12345678910"
deviceType1 = "Sensor"
deviceId1 = "DHT"
deviceType3 = "Actuator"
deviceId3 = "Water_pump"
deviceType2 = "Sensor1"
deviceId2 = "soil_moisture"
# Initialize GPIO
def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="Waterpump_on":
 print ("Water Pump is Turned ON \n")
 else :
 print ("Water pump is off")
 #print(cmd)
try:
deviceOptions1 = {"org": organization, "type": deviceType1, "id": deviceId1, "auth
method": authMethod, "auth-token":…
try:
deviceOptions1 = {"org": organization, "type": deviceType1, "id": deviceId1, "auth
method": authMethod, "auth-token": authToken}
deviceCli1 = ibmiotf.device.Client(deviceOptions1)deviceOptions2 = {"org": organization,
"type": deviceType2, "id": deviceId2, "auth
method": authMethod, "auth-token": authToken}
deviceCli2 = ibmiotf.device.Client(deviceOptions2)
deviceOptions3 = {"org": organization, "type": deviceType3, "id": deviceId3, "auth
method": authMethod, "auth-token": authToken}
```

```
deviceCli3 = ibmiotf.device.Client(deviceOptions3)
#.....
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
deviceCli1.connect()
deviceCli2.connect()
deviceCli3.connect()
while True:
 #Get Sensor Data from esp32
 temp=random.randint(0,45) Humid=random.randint(0,100)
 data1 = { 'Temperature' : temp , 'Humidity': Humid}
 #print data
 def myOnPublishCallback():
 print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,"to
IBM Watson \n")
 success1 = deviceCli1.publishEvent("DHT Sensor", "json", data1, qos=0,
on_publish=myOnPublishCallback)
 if not success1:
 print("Not connected to IoTF\n")
 time.sleep(1)
 Soil_moisture=random.randint(0,100)
 data2 = { 'Soil_moisture' : Soil_moisture}
 def myOnPublishCallback2():
 print ("Published Soil_moisture = %s %%" % temp, "to IBM Watson")
 success2 = deviceCli2.publishEvent("Soil Moisture Sensor", "json", data2, qos=0,
on_publish=myOnPublishCallback2)
 if not success2:
 print("Not connected to IoTF")
 time.sleep(1) deviceCli3.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli1.disconnect()
deviceCli2.disconnect(
```