Python Script

Date	19 November 2022
Team ID	PNT2022TMID06157
Project Name	IOT BASED CROP PROTECTION SYSTEM FOR AGRICULTURE

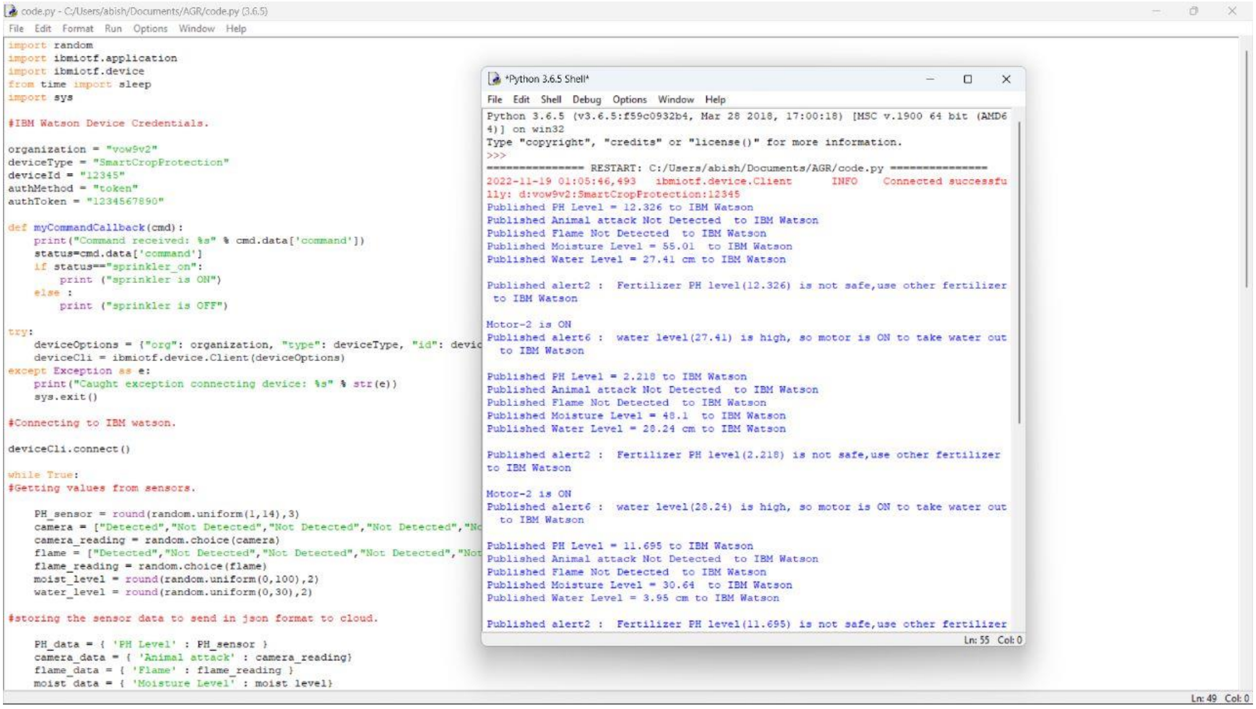
Description:

The random sensor data’s are generated and automation has been implemented through the python code instead of using hardware to implement IOT based crop protection system. And the python code need to upload the data’s in IBM cloud are written in this python script.

Python Code:

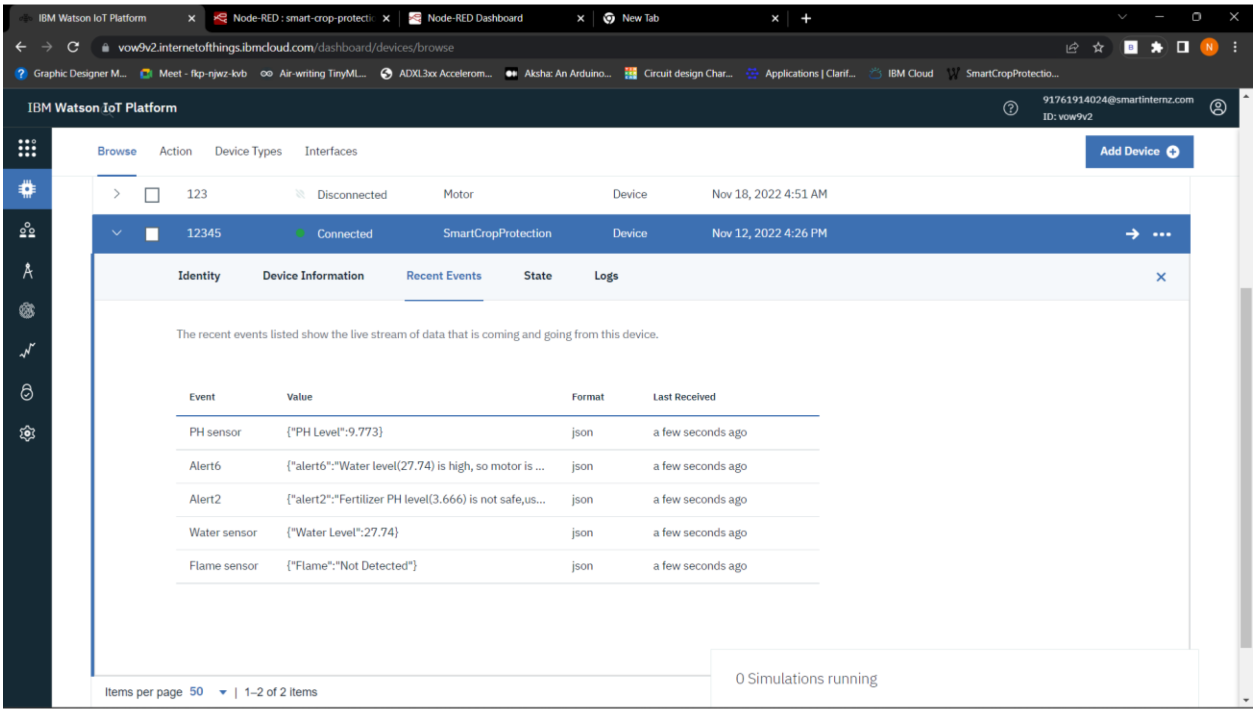
```
import random
import ibmiotf.application
import ibmiotf.device
from time import sleep
import sys

#IBM Watson Device Credentials.

organization = "vow9v2"
deviceType = "SmartCropProtection"
deviceId = "12345"
authMethod = "token"
authToken = "1234567890"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkler_on":
        print ("sprinkler is ON")
    else :
        print ("sprinkler is OFF")

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()

#Connecting to IBM watson.

deviceCli.connect()

while True:
#Getting values from sensors.

    PH_sensor = round(random.uniform(1,14),3)
    camera = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not
Detected",]
    camera_reading = random.choice(camera)
    flame = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
    flame_reading = random.choice(flame)
    moist_level = round(random.uniform(0,100),2)
    water_level = round(random.uniform(0,30),2)

#storing the sensor data to send in json format to cloud.

    PH_data = { 'PH Level' : PH_sensor }
    camera_data = { 'Animal attack' : camera_reading}
    flame_data = { 'Flame' : flame_reading }
    moist_data = { 'Moisture Level' : moist_level}
    water_data = { 'Water Level' : water_level}

    # publishing Sensor data to IBM Watson for every 5-10 seconds.

    success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
    sleep(1)
    if success:
        print ("Published PH Level = %s" % PH_sensor, "to IBM Watson")

    success = deviceCli.publishEvent("camera", "json", camera_data, qos=0)
    sleep(1)
    if success:
        print ("Published Animal attack %s " % camera_reading, "to IBM Watson")

    success = deviceCli.publishEvent("Flame sensor", "json", flame_data, qos=0)
    sleep(1)
    if success:
        print ("Published Flame %s " % flame_reading, "to IBM Watson")

    success = deviceCli.publishEvent("Moisture sensor", "json", moist_data, qos=0)
    sleep(1)
    if success:
        print ("Published Moisture Level = %s " % moist_level, "to IBM Watson")

    success = deviceCli.publishEvent("Water sensor", "json", water_data, qos=0)
    sleep(1)

    if success:
        print ("Published Water Level = %s cm" % water_level, "to IBM Watson")
        print ("")

#To send alert message if farmer uses the unsafe fertilizer to crops.

    if (PH_sensor > 7.5 or PH_sensor < 5.5):
        success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH level(%s) is not safe,use
other fertilizer" %PH_sensor } , qos=0)
        sleep(1)
        if success:
            print('Published alert2 : ' , "Fertilizer PH level(%s) is not safe,use other fertilizer" %PH_sensor,"to
IBM Watson")
            print("")
#To send alert message to farmer that animal attack on crops.
    if (camera_reading == "Detected"):
        success = deviceCli.publishEvent("Alert3", "json", { 'alert3' : "Animal attack on crops detected" },
qos=0)
        sleep(1)
        if success:
            print('Published alert3 : ' , "Animal attack on crops detected","to IBM Watson","to IBM Watson")
            print("")
#To send alert message if flame detected on crop land and turn ON the splinkers to take immediate
action.
    if (flame_reading == "Detected"):
        print("sprinkler-2 is ON")
        success = deviceCli.publishEvent("Alert4", "json", { 'alert4' : "Flame is detected crops are in
danger,sprinklers turned ON" }, qos=0)
        sleep(1)

        if success:
            print( 'Published alert4 : ' , "Flame is detected crops are in danger,sprinklers turned ON","to IBM
Watson")
            print("")
        else:
            print("sprinkler-2 is OFF")
            print("")
#To send alert message if Moisture level is LOW and to Turn ON Motor-1 for irrigation.
    if (moist_level < 20):
        print("Motor-1 is ON")
        success = deviceCli.publishEvent("Alert5", "json", { 'alert5' : "Moisture level(%s) is low, Irrigation
started" %moist_level }, qos=0)
        sleep(1)
        if success:
            print('Published alert5 : ' , "Moisture level(%s) is low, Irrigation started"%moist_level,"to IBM
Watson" )
            print("")
        else:
            print("Motor-1 is OFF")
            print("")

#To send alert message if Water level is HIGH and to Turn ON Motor-2 to take water out.
    if (water_level > 20):
        print("Motor-2 is ON")
        success = deviceCli.publishEvent("Alert6", "json", { 'alert6' : "Water level(%s) is high, so motor is ON
to take water out " %water_level }, qos=0)
        sleep(1)
        if success:
            print('Published alert6 : ' , "water level(%s) is high, so motor is ON to take water out "
%water_level,"to IBM Watson" )
            print("")
        else:
            print("Motor-2 of OFF")
            print("")
#command recived by farmer
    deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud

deviceCli.disconnect(
```

Python Script Output:


code.py - C:/Users/abish/Documents/AGR/code.py (3.6.5)
File Edit Format Run Options Window Help
Ln: 49 Col: 0
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Ln: 55 Col: 0

```python
import random
import ibmiotf.application
import ibmiotf.device
from time import sleep
import sys

#IBM Watson Device Credentials.

organization = "vow9v2"
deviceType = "SmartCropProtection"
deviceId = "12345"
authMethod = "token"
authToken = "1234567890"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkler_on":
        print ("sprinkler is ON")
    else :
        print ("sprinkler is OFF")

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": devic
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

#Connecting to IBM watson.

deviceCli.connect()

while True:
#Getting values from sensors.

    PH_sensor = round(random.uniform(1,14),3)
    camera = ["Detected","Not Detected","Not Detected","Not Detected","Nc
    camera_reading = random.choice(camera)
    flame = ["Detected","Not Detected","Not Detected","Not Detected","Not
    flame_reading = random.choice(flame)
    moist_level = round(random.uniform(0,100),2)
    water_level = round(random.uniform(0,30),2)

#storing the sensor data to send in json format to cloud.

    PH_data = { 'PH Level' : PH_sensor }
    camera_data = { 'Animal attack' : camera_reading}
    flame_data = { 'Flame' : flame_reading }
    moist_data = { 'Moisture Level' : moist_level}
```

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=============== RESTART: C:/Users/abish/Documents/AGR/code.py ===============
2022-11-19 01:05:46,493   ibmiotf.device.Client      INFO    Connected successfu
lly: d:vow9v2:SmartCropProtection:12345
Published PH Level = 12.326 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 55.01  to IBM Watson
Published Water Level = 27.41 cm to IBM Watson

Published alert2 :  Fertilizer PH level(12.326) is not safe,use other fertilizer
 to IBM Watson

Motor-2 is ON
Published alert6 :  water level(27.41) is high, so motor is ON to take water out
  to IBM Watson

Published PH Level = 2.218 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 48.1  to IBM Watson
Published Water Level = 28.24 cm to IBM Watson

Published alert2 :  Fertilizer PH level(2.218) is not safe,use other fertilizer
to IBM Watson

Motor-2 is ON
Published alert6 :  water level(28.24) is high, so motor is ON to take water out
  to IBM Watson

Published PH Level = 11.695 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 30.64  to IBM Watson
Published Water Level = 3.95 cm to IBM Watson

Published alert2 :  Fertilizer PH level(11.695) is not safe,use other fertilizer
```



☐	123	Disconnected	Motor	Device	Nov 18, 2022 4:51 AM
☐	12345	Connected	SmartCropProtection	Device	Nov 12, 2022 4:26 PM

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
PH sensor	{"PH Level":9.773}	json	a few seconds ago
Alert6	{"alert6":"Water level(27.74) is high, so motor is ...	json	a few seconds ago
Alert2	{"alert2":"Fertilizer PH level(3.666) is not safe,us...	json	a few seconds ago
Water sensor	{"Water Level":27.74}	json	a few seconds ago
Flame sensor	{"Flame":"Not Detected"}	json	a few seconds ago