

SPRINT-2

TEAM ID: PNT2022TMID32078

PROJECT TITLE: IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

Source code to deployed on IBM Watson Iot platform to generate the sensor data.

SOURCE CODE:

```
import time import
```

```
sys
```

```
import ibmiotf.application
```

```
import ibmiotf.device import
```

```
random
```

```
#Provide your IBM Watson Device Credentials
```

```
organization = "iritj7" deviceType = "abcd"
```

```
deviceId = "12345" authMethod = "token"
```

```
authToken = "12345678"
```

```
# 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

    temp=random.randint(90,110)
    Humid=random.randint(60,100)
    Moist=random.randint(20,100)
    Animal_dect=random.randint(1,20)

    data = { 'temp' : temp, 'Humid': Humid, 'Moist' : Moist, 'Animal_dect' :
Animal_dect }

```

```

# print data      def
myOnPublishCallback():

    print ("Published Temperature = %s C" % temp, "Humidity = %s
%%" % Humid, "to IBM Watson", "Published Moisture= %s" % Moist,
"Published Animal detection = ", Animal_dect)

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)    if not success:        print("Not
connected to IoT")        time.sleep(10)

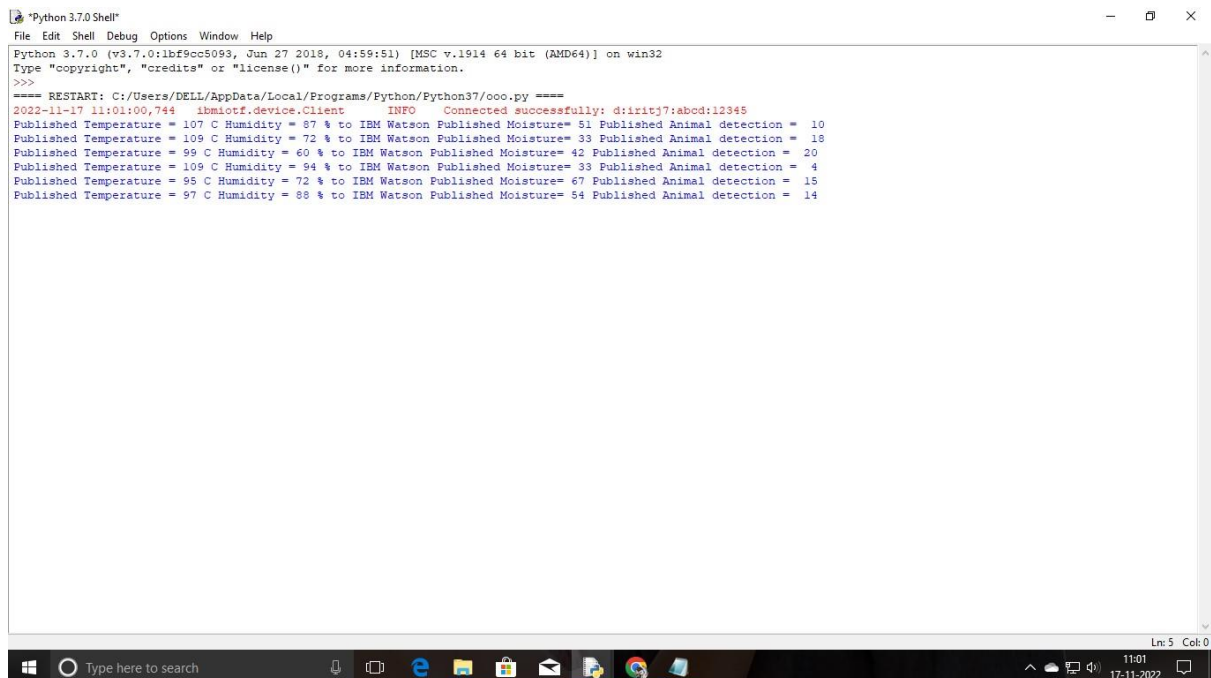
deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud deviceCli.disconnect()

```

OUTPUT:

OUTPUT:



```

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python37/ooo.py ====
2022-11-17 11:01:00,744 ibmiotf.device.Client INFO Connected successfully: d:irictj7:abcd:12345
Published Temperature = 107 C Humidity = 87 % to IBM Watson Published Moisture= 51 Published Animal detection = 10
Published Temperature = 109 C Humidity = 72 % to IBM Watson Published Moisture= 33 Published Animal detection = 18
Published Temperature = 99 C Humidity = 60 % to IBM Watson Published Moisture= 42 Published Animal detection = 20
Published Temperature = 109 C Humidity = 94 % to IBM Watson Published Moisture= 33 Published Animal detection = 4
Published Temperature = 95 C Humidity = 72 % to IBM Watson Published Moisture= 67 Published Animal detection = 15
Published Temperature = 97 C Humidity = 88 % to IBM Watson Published Moisture= 54 Published Animal detection = 14

```

SENSOR DATA:

Device Type: abcd

Events 1

New event type +

Event type name

event_1

Send

Schedule

20

Every Minute

Payload

Specify the event payload in the editor window or by uploading a [CSV file](#).

```
0 {  
1   "temp": random(90,110)  
2   "Humid": random(60,100)  
3   "Moist": random(0,100)  
4   "Animal_dect": random(0,2)  
5  
6 }
```