

## Project Development Phase

### Sprint-3

Date	16 November 2022
Team ID	PNT2022TMID11064
Project Name	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

#### Python code:

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

organization = "gjx22e"
deviceType = "cropprotection"
deviceId = "12345"
authMethod = "token"
authToken = "987654321"
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkler_on":
        print ("sprinkler is turning ON")
    else :
        print ("sprinkler is turning 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("Exception detected in connecting device: %s" % str(e))
    sys.exit()
deviceCli.connect()
while True:
    #Getting values from sensors...
    temp_sensor = round( random.uniform(0,80),2)
    PH_sensor = round(random.uniform(1,14),3)
    moist_level = round(random.uniform(0,100),2)
    water_level = round(random.uniform(0,30),2)

    temp_data = { 'Temp' : temp_sensor }
    PH_data = { 'PH value' : PH_sensor }

    moist_data = { 'Moisture level' : moist_level}
    water_data = { 'Water level' : water_level}

    success = deviceCli.publishEvent("Temperature sensor", "json", temp_data, qos=0)
```

```

sleep(1)
if success:
    print ("... ..publish ok... ..")
    print ("Published Temp = %s C" % temp_sensor, "to IBM Watson")
    success = deviceCli.publishEvent("PH sensor", "json", PH_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 ("")

if (temp_sensor > 35):
    print("sprinkler-1 is ON")
    success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s)
is high, sprinklerlers are turned ON" %temp_sensor }, qos=0)
    sleep(1)
if success:
    print( 'Published Alert1 : ', "Temperature(%s) is high, sprinklerlers are turned
ON" %temp_sensor,"to IBM Watson")
    print("")
else:
    print("sprinkler-1 is OFF")
    print("")

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("")
else:
    print("sprinkler-2 is OFF")
    print("")
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("")
if (water_level > 20):
    print("Motor-2 is turning 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 is turning OFF")
        print("")
deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()

```

## OUTPUT

The screenshot shows a Visual Studio Code window with a Python file named `main.py` and a terminal window displaying the execution output.

**main.py Code:**

```

1 import random
2 import ibmiotf.application
3 import ibmiotf.device
4 from time import sleep
5 import sys
6
7 organization = "gjx22e"
8 deviceType = "cropprotection"
9 deviceId = "12345"
10 authMethod = "token"
11 authToken = "987654321"
12 def myCommandCallback(cmd):
13     print("Command received: %s" % cmd.data['command'])
14     status=cmd.data['command']
15     if status=="sprinkler_on":
16         print("sprinkler is turning ON")
17     else :
18         print("sprinkler is turning OFF")
19 try:
20     deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}

```

**Terminal Output:**

```

PS C:\Users\Santhosh Balaji A R\Documents\Santhosh Programs> & "C:/Users/Santhosh Balaji A R/AppData/Local/Microsoft/WindowsApps/python3.9.exe" "c:/Users/Santhosh Balaji A R/Desktop/Ro
shana/main.py"
2022-11-19 09:21:50,359 ibmiotf.device.Client INFO Connected successfully: d:gjx22e:cropprotection:12345
... ..publish ok... ..
Published Temp = 0.3 C to IBM Watson
Published Moisture level = 67.0 to IBM Watson
Published Water level = 5.66 cm to IBM Watson

Published Alert1 : Temperature(0.3) is high, sprinklers are turned ON to IBM Watson
Published Alert2 : Fertilizer PH level(2.583) is not safe,use other fertilizer to IBM Watson
Published Alert5 : Moisture level(67.0) is low, Irrigation started to IBM Watson
Published Alert6 : water level(5.66) is high, so motor is ON to take water out to IBM Watson

```

## OUTPUT IN IBM WATSON:

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Device

### Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	cropprotection	Device	Nov 18, 2022 11:08 PM	
53302945	Disconnected	smartcrop	Device	Nov 17, 2022 10:39 PM	

Items per page 50 | 1-2 of 2 items

1 of 1 page

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Device

12345 Connected cropprotection Device Nov 18, 2022 11:08 PM

Identity Device Information Recent Events State Logs

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 value":10.384}	json	a few seconds ago
Temperature ...	{"Temp":51.62}	json	a few seconds ago
Alert6	{"alert6":"Water level(27.14) is high, so motor is ...	json	a few seconds ago
Water sensor	{"Water level":27.14}	json	a few seconds ago
PH sensor	{"PH value":6.7}	json	a few seconds ago

53302945 Disconnected smartcrop Device Nov 17, 2022 10:39 PM

Items per page 50 | 1-2 of 2 items

1 of 1 page

