

## 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 = "gix22e"
deviceType = "smartcrop"
deviceId = "53302945"
authMethod = "use-token-auth"
authToken = "987654321"
def myCommandCallback(cmd):
    print("%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:
    temp = round( random.uniform(0,80),2)
    PH = round(random.uniform(1,14),3)
    moisture= round(random.uniform(0,100),2)
    water_level = round(random.uniform(0,30),2)
    temp_data = { 'Temp' : temp }
    PH_data = { 'PH value' : Ph }
    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, "to IBM Watson")
success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
sleep(1)
if success:
    print ("Published PH value = %s" % Ph, "to IBM Watson")
    success = deviceCli.publishEvent("camera", "json", camera_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 > 35):
    print("sprinkler-1 is ON")
    success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s) is high, sprinklerlers are turned
ON" %temp }, qos=0)
    sleep(1)
if success:
    print('Published Alert1 : ', "Temperature(%s) is high, sprinklerlers are turned ON" %temp,"to IBM Watson")
    print("")
else:
    print("sprinkler-1 is OFF")
    print("")
if (Ph > 7.5 or Ph < 5.5):
    success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH level(%s) is not safe,use other
fertilizer" %Ph }, qos=0)
    sleep(1)
if success:
    print('Published Alert2 : ', "Fertilizer PH level(%s) is not safe,use other fertilizer" %Ph,"to IBM Watson")
    print("")
deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()

```

## OUTPUT IN IBM WATSON:

The screenshot displays the IBM Watson IoT Platform interface. The browser address bar shows the URL: `gix22e.internetofthings.ibmcloud.com/dashboard/devices/drilldown/smartcrop:53302945?returnTo=/devices/browse`. The page title is "Device Drilldown - 53302945".

On the left, a sidebar contains navigation links: "Back", "Connection Information", "Recent Events", "State", "Device Information", "Metadata", "Diagnostics", "Connection Logs", and "Device Actions".

The main content area is divided into two sections:

- Connection Status:** Displays "Disconnected".
- Recent Events:** Includes a description: "The recent events listed show the live stream of data that is coming and going from this device." Below this is a table with the following data:

Event	Value	Format	Last Received
Temperature	{"Temperature":9}	json	a few seconds ago
Temperature	{"Temperature":31}	json	a few seconds ago
Temperature	{"Temperature":23}	json	a few seconds ago
Temperature	{"Temperature":15}	json	a few seconds ago
Temperature	{"Temperature":7}	json	a few seconds ago

At the bottom right, a status box indicates "1 Simulation running".