

## SPRINT 1

Date	28 OCTOBER 2022
Team ID	PNT2022TMID18437
Project Name	Smart Farmer-IOT Enabled Smart Farming Application

```
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.0 (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: E:\IBM\Code\Motor_Project.py =====
2022-11-18 19:55:21,981 ibmiotf.device.Client INFO Connected successfully: d:ptrr0s:Obstacle:100
Published Level = 90 % to IBM Watson
Published Level = 19 % to IBM Watson
Published Level = 73 % to IBM Watson
Published Level = 16 % to IBM Watson
Command received: MotorOn
Motor is running
Published Level = 92 % to IBM Watson
Published Level = 81 % to IBM Watson
Command received: MotorOff
Motor is switched Off
Published Level = 72 % to IBM Watson

Motor_Project.py - E:\IBM\Code\Motor_Project.py (3.7.0)
File Edit Format Run Options Window Help
    print("Motor is running")
    #fillup(dis)
elif status == "MotorOff":
    print("Motor is switched Off")
else:
    print("please enter a valid 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:
    dis = random.randint(0,100)
    data = { 'Distance' : dis}
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if dis > 95 :
        print("Motor is switched Off, Tank Full!")
    if not success:
        print("Not connected to IoT")
    deviceCli.commandCallback = myCommandCallback
    time.sleep(10)

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

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

Python 3.7.0 (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: E:\IBM\Code\Project.py =====
2022-11-18 19:58:53,183 ibmiotf.device.Client INFO Connected successfully: d:ptrr0s:Field1:1001
2022-11-18 19:58:54,465 ibmiotf.device.Client INFO Connected successfully: d:ptrr0s:Field2:1002

Command received: Servo2Off
Control valve for Field2 is Closed!
Command received: Servo1Off
Control valve for Field1 is Closed!
Command received: Servo1On
Control valve for Field1 is Open!

Command received: Servo2On
Control valve for Field2 is Open!
```

```
Project.py - E:\IBM\Code\Project.py (3.7.0)
File Edit Format Run Options Window Help

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceOptions1 = {"org": organization, "type": deviceType1, "id": deviceId1, "auth-method": authMethod, "auth-token": authToken1}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    deviceCli1 = ibmiotf.device.Client(deviceOptions1)
    #.....

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()
deviceCli1.connect()

def myOnPublishCallback():
    #print("Published Moisture = %s" % soil, "Nitrogen = %s" % N, "Phosphorus = %s" % P, "Potassium = %s" % Ka, "to IBM Watson")
    print()

while True:
    soil = random.randint(0,100)
    N = random.randint(0,100)
    P = random.randint(0,100)
    Ka = random.randint(0,100)
    data = { 'Moisture': soil, 'Nitrogen': N, 'Phosphorus': P, 'Potassium': Ka}, 'randomNumber': No)
    #print("-----Field 1 Parameters-----")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    soil = random.randint(0,100)
    N = random.randint(0,100)
    P = random.randint(0,100)
    Ka = random.randint(0,100)
    data = { 'Moisture': soil, 'Nitrogen': N, 'Phosphorus': P, 'Potassium': Ka}
    #print("-----Field 2 Parameters-----")
```

```
Project.py - E:\IBM\Code\Project.py (3.7.0)
File Edit Format Run Options Window Help

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "ptrr0s"
deviceType = "Field1"
deviceId = "1001"
deviceType1 = "Field2"
deviceId1 = "1002"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "Servo1On":
        print ("Control valve for Field1 is Open!")
    elif status == "Servo1Off":
        print ("Control valve for Field1 is Closed!")
    elif status == "Servo2On":
        print ("Control valve for Field2 is Open!")
    elif status == "Servo2Off":
        print ("Control valve for Field2 is Closed!")
    else:
        print("please enter a valid command")
```