

SPRINT DELIVERY – 4

| | |
|--------------|---|
| Team ID | PNT2022TMID16026 |
| Project Name | Smart Farmer-IoT Enabled Smart Farming Application |

Receiving commands from IBM cloud using Python program

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "14dcvs",
        "typeId": "Device1",
        "deviceId": "12345"
    },
    "auth": {
        "token": "87654321"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="Motor On"):
        print("*///Motors ARE ON///*")
    else:
        print("*///Motors ARE OFF///*")

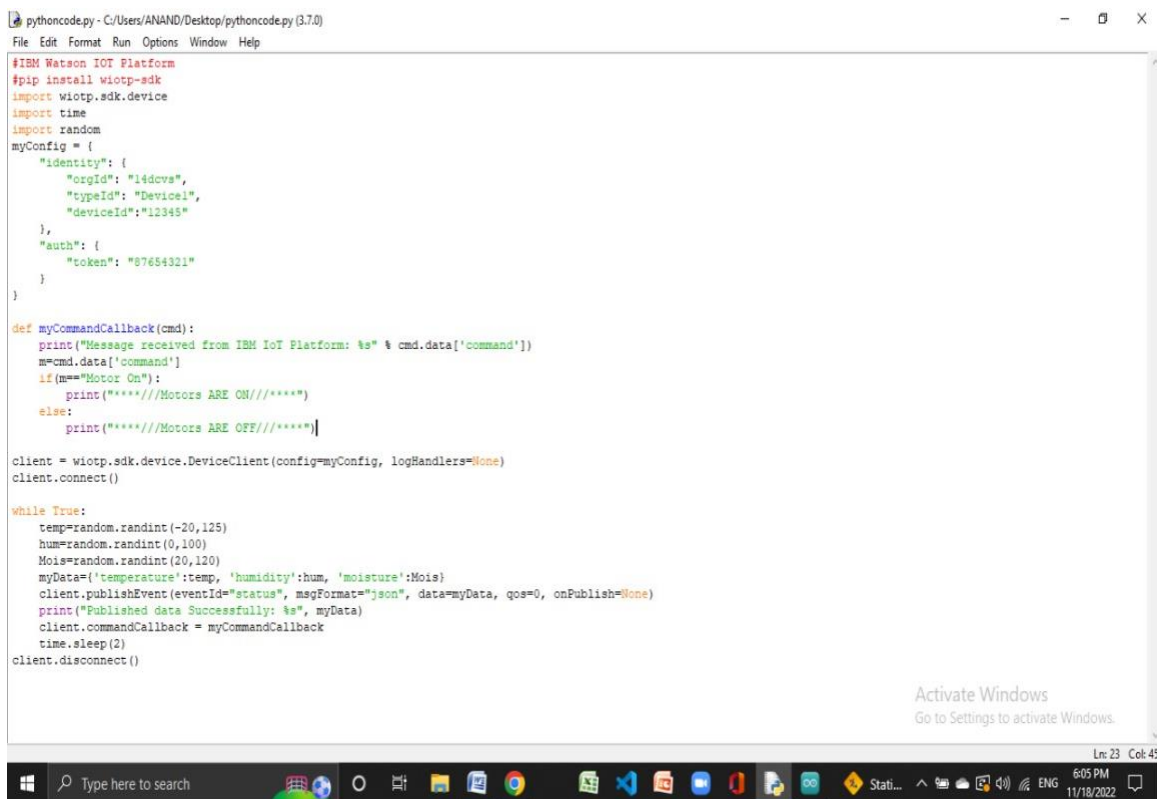
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
```

```

    Mois=random.randint(20,120)
    myData={'temperature':temp, 'humidity':hum, 'moisture':Mois}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()

```



```

pythoncode.py - C:/Users/ANAND/Desktop/pythoncode.py (3.7.0)
File Edit Format Run Options Window Help

#IBM Watson IoT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "14dcvs",
        "typeId": "Device1",
        "deviceId": "12345"
    },
    "auth": {
        "token": "97654321"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="Motor On"):
        print("*****//Motors ARE ON//*****")
    else:
        print("*****//Motors ARE OFF//*****")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    Mois=random.randint(20,120)
    myData={'temperature':temp, 'humidity':hum, 'moisture':Mois}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()

Activate Windows
Go to Settings to activate Windows.

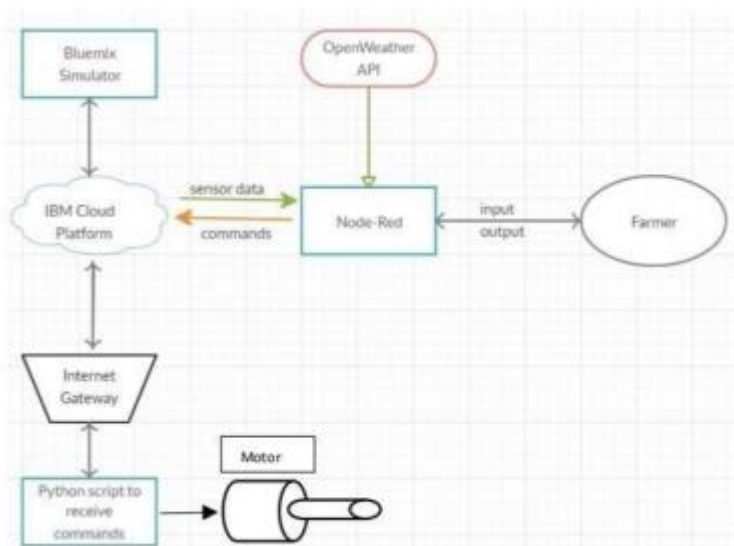
Ln: 23 Col: 45
6:05 PM
11/18/2022

```

```
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/ANAND/Desktop/pythoncode.py =====
2022-11-18 18:06:01,244 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d14dcvs:Device1:12345Published data Successfully: %s
({'temperature': 75, 'humidity': 58, 'moisture': 22})
Published data Successfully: %s ('temperature': 44, 'humidity': 33, 'moisture': 39)
Published data Successfully: %s ('temperature': 11, 'humidity': 62, 'moisture': 93)
Published data Successfully: %s ('temperature': 102, 'humidity': 10, 'moisture': 32)
Published data Successfully: %s ('temperature': 107, 'humidity': 51, 'moisture': 32)
Published data Successfully: %s ('temperature': -7, 'humidity': 58, 'moisture': 77)
Published data Successfully: %s ('temperature': 97, 'humidity': 90, 'moisture': 23)
Published data Successfully: %s ('temperature': 58, 'humidity': 100, 'moisture': 21)
```

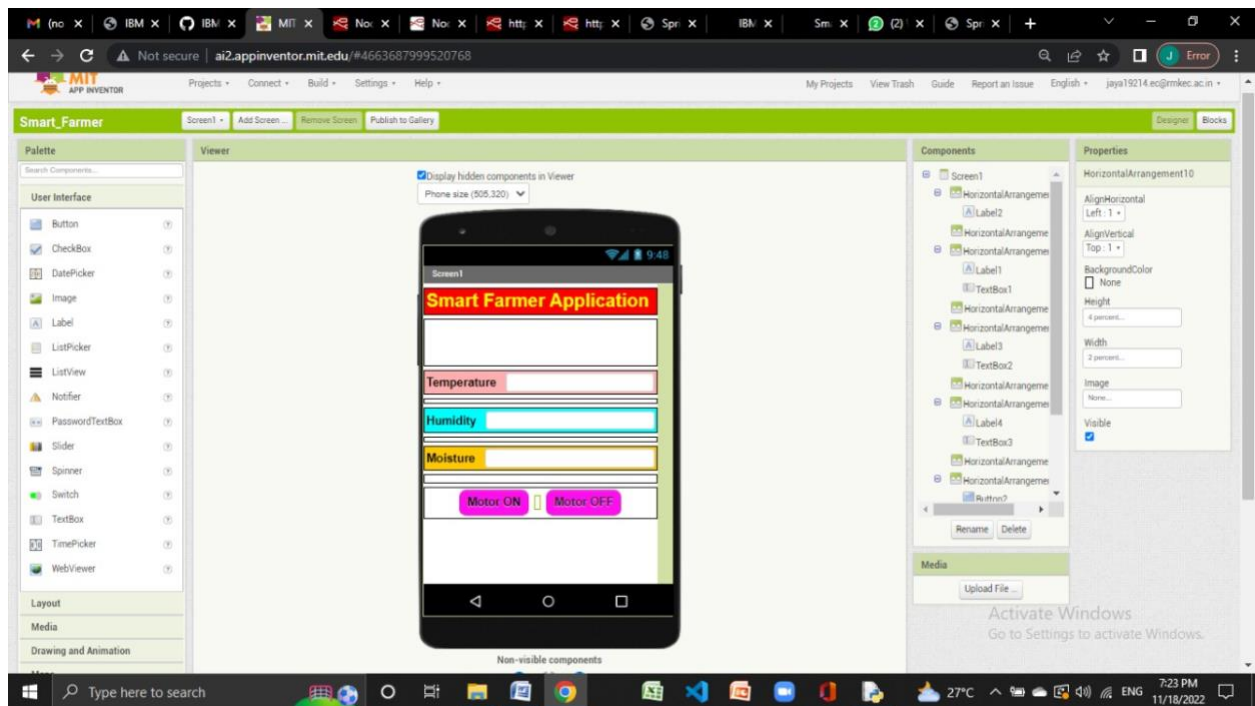
FLOWCHART



The screenshot displays a Windows 10 desktop environment. At the top, a taskbar contains the Start button, a search bar with the text "Type here to search", and several pinned application icons including File Explorer, Microsoft Edge, and various utility tools. The main area of the screen is occupied by a "Python 3.7.0 Shell" window. The window's title bar includes standard Windows controls (minimize, maximize, close) and a menu bar with options: File, Edit, Shell, Debug, Options, Window, and Help. The shell's output area shows the following text:

```
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: C:/Users/ANAND/Desktop/pythoncode.py =====
2022-11-18 18:06:01.244  wsotp.sdr.device.client.DeviceClient  INFO      Connected successfully: d:\4dovs:Device1:12345Published data Successfully: %s
({'temperature': 75, 'humidity': 55, 'moisture': 22})
Published data Successfully: %s ({'temperature': 44, 'humidity': 33, 'moisture': 39})
Published data Successfully: %s ({'temperature': 11, 'humidity': 62, 'moisture': 93})
Published data Successfully: %s ({'temperature': 102, 'humidity': 10, 'moisture': 32})
Published data Successfully: %s ({'temperature': 107, 'humidity': 51, 'moisture': 32})
Published data Successfully: %s ({'temperature': -7, 'humidity': 58, 'moisture': 77})
Published data Successfully: %s ({'temperature': 97, 'humidity': 90, 'moisture': 23})
Published data Successfully: %s ({'temperature': 58, 'humidity': 100, 'moisture': 21})
```

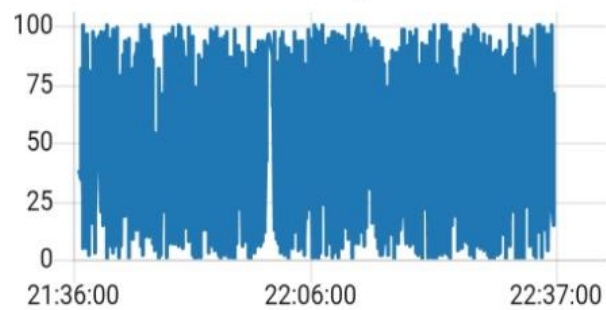
In the bottom right corner of the shell window, there is a watermark that reads "Activate Windows Go to Settings to activate Windows." The bottom of the screen shows the Windows taskbar with the system clock indicating 6:06 PM on 11/18/2022.



Motor

sensor data

Humidity



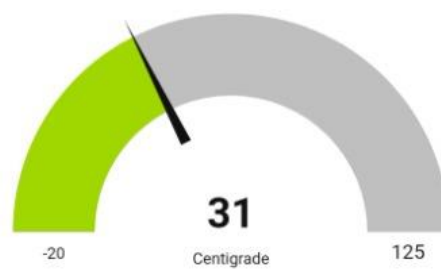
MOTOR ON

Moisture



MOTOR OFF

Temperature



ADVANTAGES AND DISADVANTGES

ADVANTAGES:

- ✓ A remote control system can help in working irrigation system valves dependent on schedule. Irrigating remote farm properties can be exceptionally troublesome and labor-intensive. It gets hard to comprehend when the valves were started and whether the ideal measure of water was distributed.
- ✓ For situations where a quick reaction is required, manual valve actuation may not be conceivable constantly. Thus, remote observing and control of irrigation systems, generators or wind machines or some other motor-driven hardware become the next logical step.
- ✓ Various solutions are available to monitor engine statistics and starting or stopping the engine. When the client chooses to begin or stop the motor, the program transmits a sign to the unit within seconds by means of a mobile phone system.

DISADVANTAGES:

- ✓ The smart agriculture needs availability of internet continuously. In Rural part most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.
- ✓ The smart farming based equipment requires farmers to understand and learn the use of technology. This is major challenge in adopting smart agriculture farming at large scale across the countries.

CONCLUSION

The project concludes that this system is easy to implement and time, money, and manpower saving solution for irrigating fields. It will be comfortable for farmers to operate the irrigation at remote locations i.e. from home. A farmer should visualize his agricultural land's moisture content from time to time and whether the water level of the source is sufficient or not. The IOT based irrigation system displays the values of the sensors continuously in smart phones and farmers can operate them anytime from and anywhere. This will save time and avoid the problem of continuous vigilance. Not only this, it will also control the consumption of water for the irrigation of the field, thus preventing the water wastage and would help in sustaining productivity, increasing the yield.