TEAM ID PNT2022TMID12813

IOT ENABLED SMART FARMINGAPPLICATION

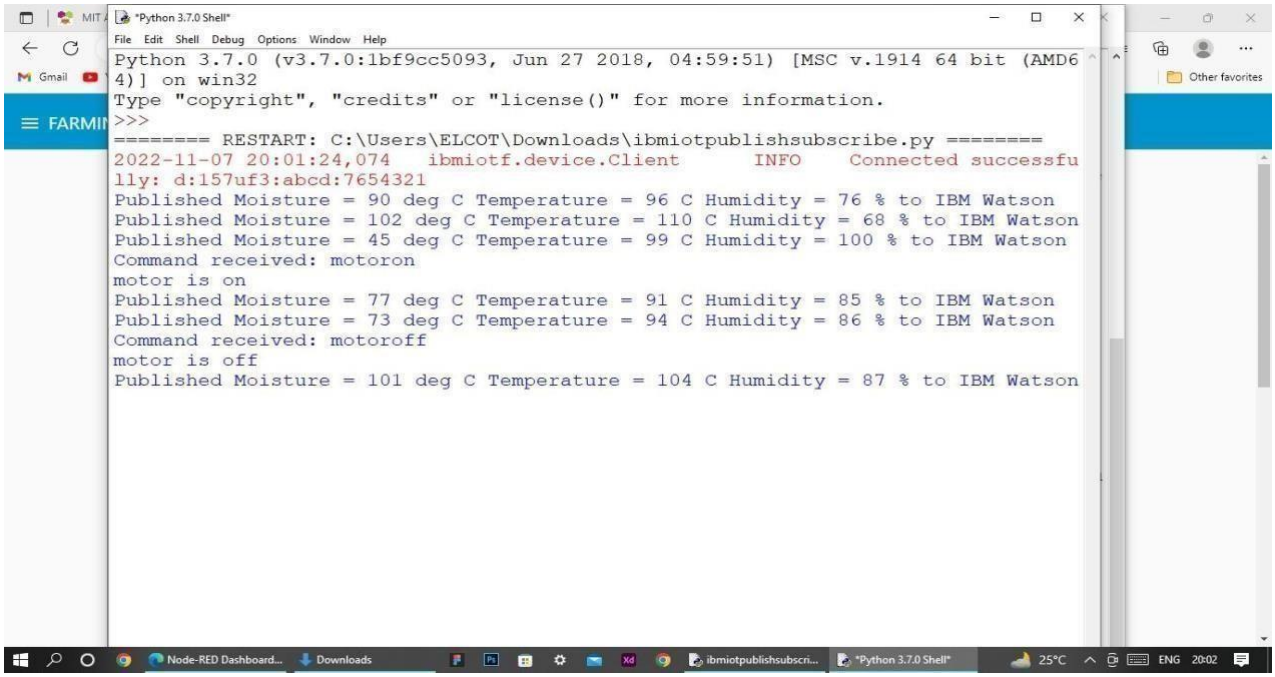
SPRINT DELIVERY – 4

Receiving commands from IBM cloud using Python program

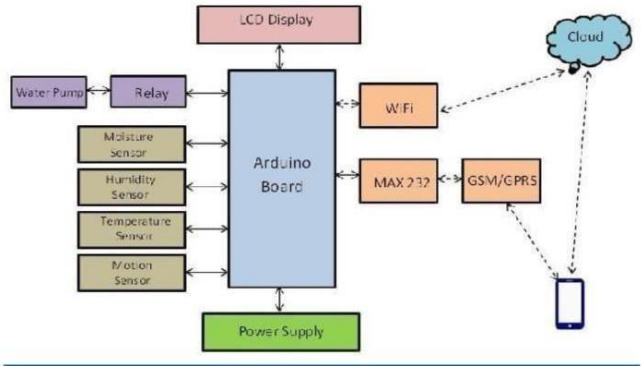
```
import timeimport sys
import ibmiotf.application import
ibmiotf.device import random


#Provide your IBM Watson Device
Credentialsorganization = "0hzydu"
 deviceType =
"NodeMCU"deviceId =
"12345" authMethod =
"token" authToken =
"12345678"


# Initialize GPIO  def myCommandCallback(cmd):
print("Command received: %s" %cmd.data['command'])
status=cmd.data['command']
 if status=="motoron":
print ("motor is on")
                                eli
fstatus == "motoroff":
 print ("motor 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)

Mois=random. Randint(20,120)

data={'temp':temp,'Humid':Humid,‘Mois’

:Mois}

      #print data

 def  myOnPublishCallback():

 print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %Humid, “Moisture =%s deg c” % Mois “to IBM Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)

 if not success:

       print("Not connected to IoTF")         time.sleep(10)

deviceCli.commandCallback = myCommandCallback

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

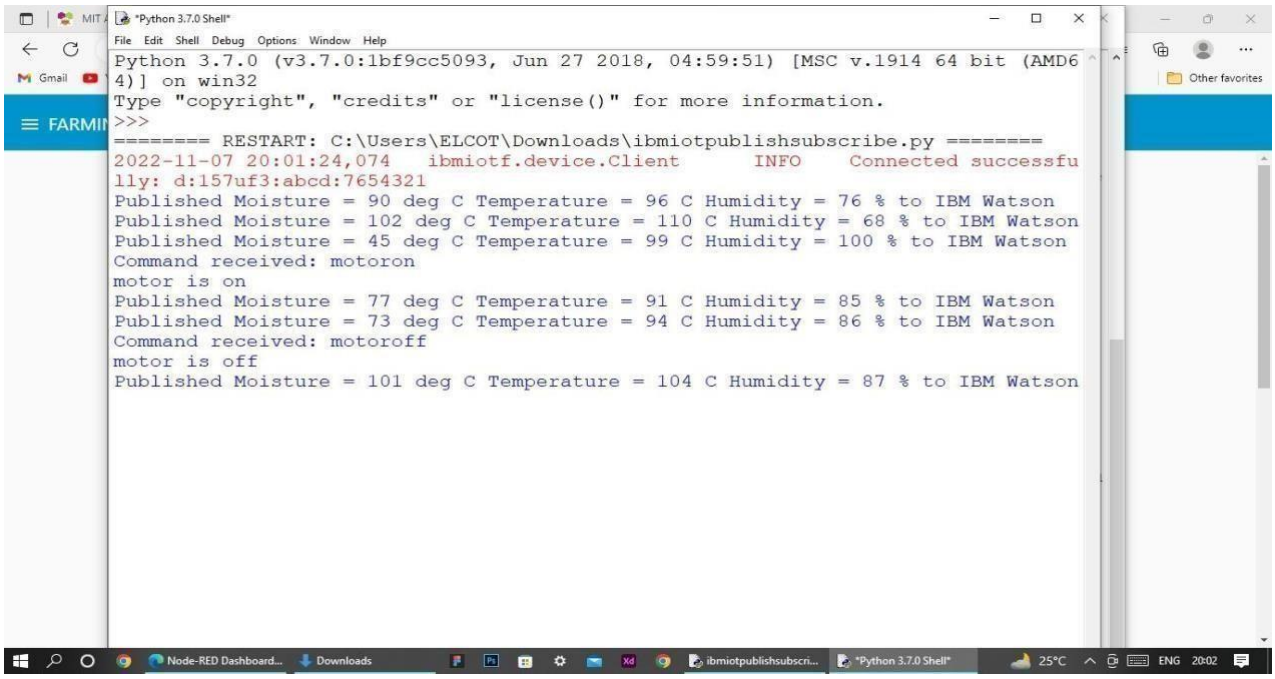
OUTPUT FROM PYTHON



Flow Chart



Observations & Results


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: C:\Users\ELCOT\Downloads\ibmiotpublishsubscribe.py ========
2022-11-07 20:01:24,074 ibmiotf.device.Client INFO Connected successfully: d:157uf3:abcd:7654321
Published Moisture = 90 deg C Temperature = 96 C Humidity = 76 % to IBM Watson
Published Moisture = 102 deg C Temperature = 110 C Humidity = 68 % to IBM Watson
Published Moisture = 45 deg C Temperature = 99 C Humidity = 100 % to IBM Watson
Command received: motoron
motor is on
Published Moisture = 77 deg C Temperature = 91 C Humidity = 85 % to IBM Watson
Published Moisture = 73 deg C Temperature = 94 C Humidity = 86 % to IBM Watson
Command received: motoroff
motor is off
Published Moisture = 101 deg C Temperature = 104 C Humidity = 87 % to IBM Watson
Gmail
Other favorites
Node-RED Dashboard...
Downloads
ibmiotpublishsubscri...
Python 3.7.0 Shell
25°C
ENG
20:02


MIT App Inventor
Service Details - IBM Cloud
Node-RED Dashboard
https://node-red-hdyfv-2022-10-01.eu-gb.mybluemix.net/ui/#!/1?socketid=ZgBLuLWNfw0sYpeTAAAZ
Gmail
YouTube
Maps
New Tab
Other favorites
FARMING MEASURE DATA
Farming Measure Data
Temperature
110
108
106
104
102
100
98
96
94
92
90
20:01:33
20:01:53
20:02:13
20:02:33
20:02:53
20:03:13
20:03:34
Humidity
91
%
0
100
Moisture
110
100
90
80
70
60
50
40
30
20
20:01:33
20:02:03
20:02:33
20:03:03
20:03:54
Switchboard
MOTOR SWITCH ON
MOTOR SWITCH OFF
Node-RED Dashboard...
Downloads
ibmiotpublishsubscri...
Python 3.7.0 Shell
Pollut...
ENG
20:03

Conclusion

Thus the objective of the project to implement an IoT system in order to help farmers to control and monitor their farms has been implemented successfully.