IOT ENABLED SMART FARMINGAPPLICATION

SPRINT DELIVERY – 4

Date	17.11.2022
Team ID	PNT2022TMID06965
Project Name	SMART FARMER - IOT ENABLED SMART FARMINGAPPLICATION SYSTEM

Receiving commands from IBM cloud using Python program

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device
organization = "wj54qd"
deviceType="abc"
deviceId = "123"
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")
    elif status == "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()
deviceCli.connect()
while True:
   temp=random.randint(90,110)
   Humid=random.randint(60,100)
   Mois=random.randint(20,120)
   data = { 'temp' : temp, 'hum': 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("event", "json", data,qos=0,
on_publish=myOnPublishCallback)
   if not success:
      print("Not connected to IoTF")
   time.sleep(3)
   deviceCli.commandCallback = myCommandCallback
#Disconnect the device and application from the cloud
deviceCli.disconnect()
```

iot.py - C:\Users\ELCOT\Desktop\iot.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
organization = "wj54qd"
deviceType="abc"
deviceId = "123"
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")
    elif status == "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()
deviceCli.connect()
while True:
   temp=random.randint(90,110)
   Humid=random.randint(60,100)
   Mois=random.randint(20,120)
   data = { 'temp' : temp, 'hum': 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("event", "json", data,qos=0, on_publish=myOnPublishCallback)
   if not success:
```

Ln: 17 Col: 0

Type here to search ENG 01:10 18-11-2022

iot.py - C:\Users\ELCOT\Desktop\iot.py (3.7.0)

File Edit Format Run Options Window Help

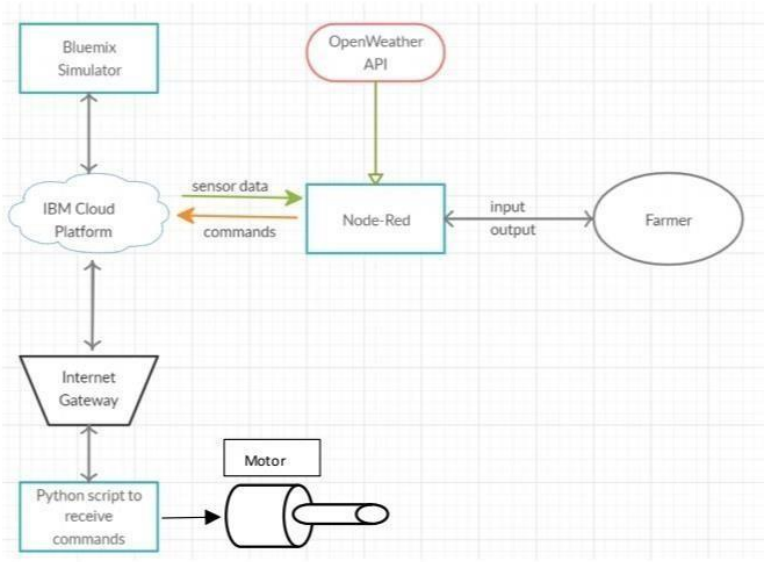
```
deviceType="abc"
deviceId = "123"
authMethod = "token"
authToken= "12345678"
# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    elif status == "motoroff":
        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()
deviceCli.connect()
while True:
   temp=random.randint(90,110)
   Humid=random.randint(60,100)
   Mois=random.randint(20,120)
   data = { 'temp' : temp, 'hum': 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("event", "json", data,qos=0, on_publish=myOnPublishCallback)
   if not success:
       print("Not connected to IoTF")
   time.sleep(3)
   deviceCli.commandCallback = myCommandCallback
#Disconnect the device and application from the cloud
deviceCli.disconnect()
```

Ln: 17 Col: 0

Type here to search ENG 01:11 18-11-2022

Flow Chart



Observations & Results

Python 3.7.0 Shell

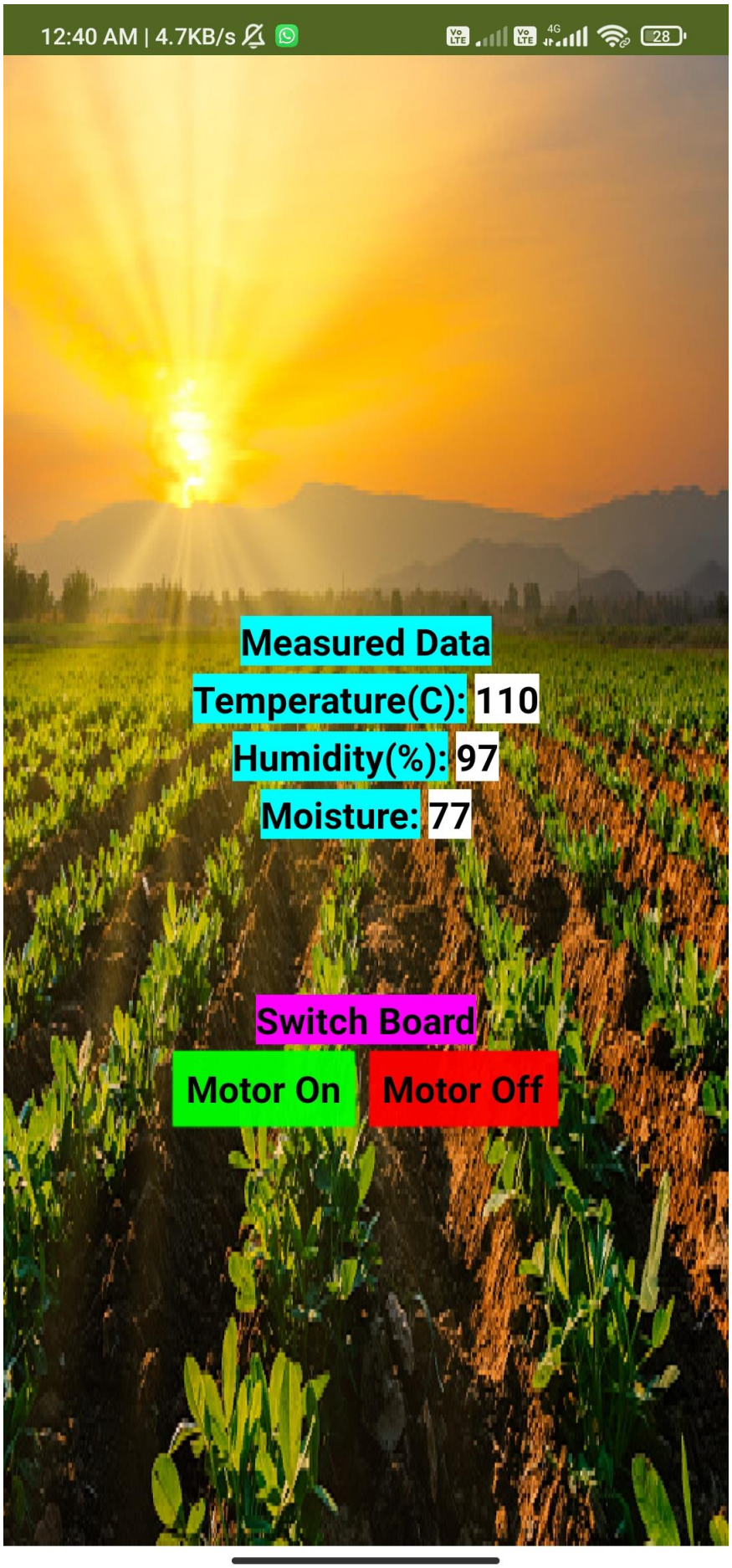
File Edit Shell Debug Options Window Help

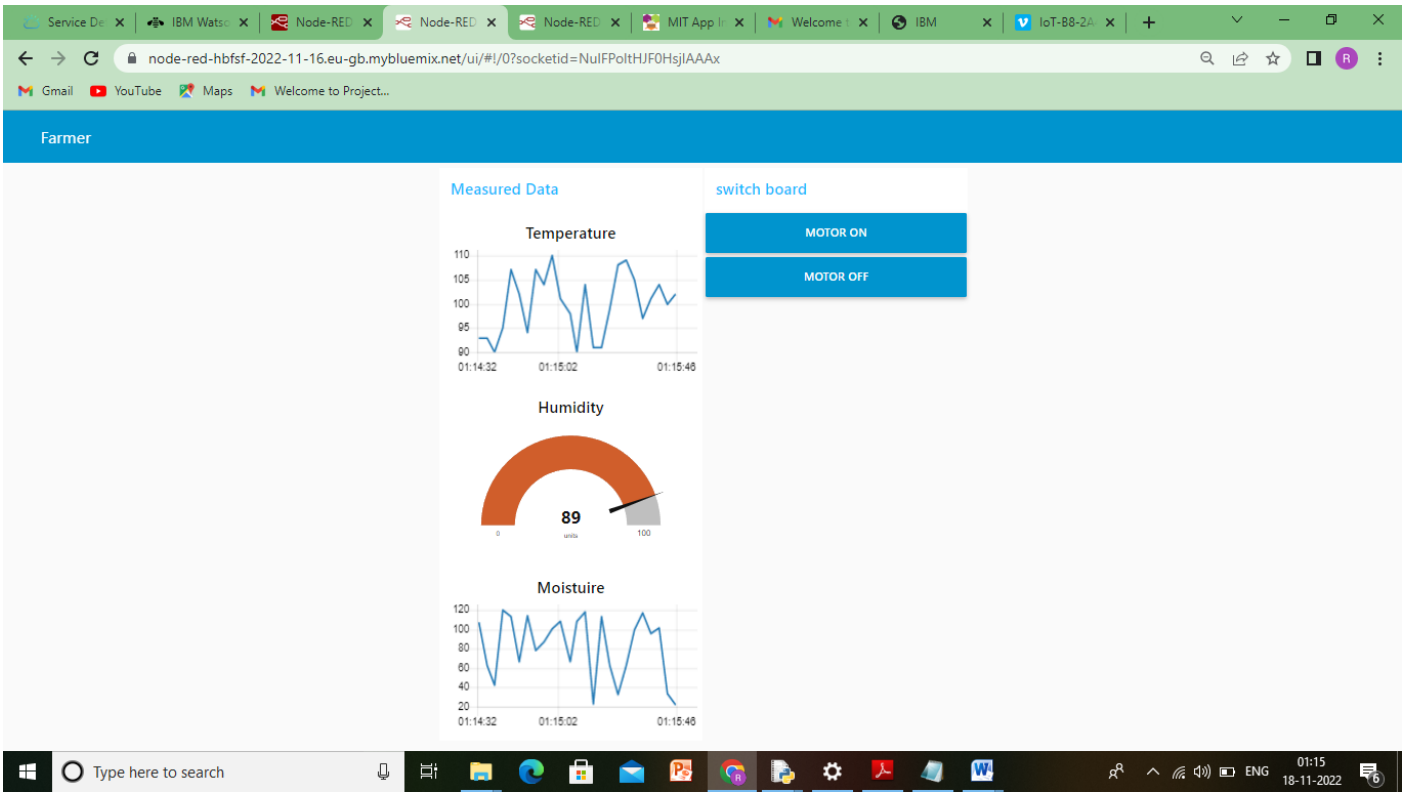
```
========== RESTART: C:\Users\ELCOT\Desktop\iot.py ==========
2022-11-18 01:11:15,198   ibmiotf.device.Client      INFO    Connected successfully: d:wj54qd:abc:123
published Temperature = 106 C Humidity = 91 % Moisture =52 deg c to IBM Watson
published Temperature = 109 C Humidity = 82 % Moisture =29 deg c to IBM Watson
published Temperature = 104 C Humidity = 74 % Moisture =116 deg c to IBM Watson
published Temperature = 94 C Humidity = 74 % Moisture =85 deg c to IBM Watson
published Temperature = 108 C Humidity = 91 % Moisture =84 deg c to IBM Watson
published Temperature = 97 C Humidity = 66 % Moisture =76 deg c to IBM Watson
published Temperature = 97 C Humidity = 77 % Moisture =103 deg c to IBM Watson
published Temperature = 90 C Humidity = 74 % Moisture =45 deg c to IBM Watson
published Temperature = 101 C Humidity = 71 % Moisture =38 deg c to IBM Watson
published Temperature = 100 C Humidity = 94 % Moisture =73 deg c to IBM Watson
Command received: motoron
motor is on
published Temperature = 92 C Humidity = 93 % Moisture =32 deg c to IBM Watson
published Temperature = 95 C Humidity = 88 % Moisture =59 deg c to IBM Watson
published Temperature = 102 C Humidity = 75 % Moisture =120 deg c to IBM Watson
Command received: motoroff
motor is off
published Temperature = 107 C Humidity = 97 % Moisture =105 deg c to IBM Watson
published Temperature = 94 C Humidity = 60 % Moisture =56 deg c to IBM Watson
published Temperature = 92 C Humidity = 80 % Moisture =113 deg c to IBM Watson
Command received: motoron
motor is on
published Temperature = 96 C Humidity = 89 % Moisture =59 deg c to IBM Watson
published Temperature = 100 C Humidity = 75 % Moisture =85 deg c to IBM Watson
published Temperature = 90 C Humidity = 100 % Moisture =82 deg c to IBM Watson
published Temperature = 97 C Humidity = 69 % Moisture =108 deg c to IBM Watson
published Temperature = 98 C Humidity = 75 % Moisture =114 deg c to IBM Watson
Command received: motoroff
motor is off
published Temperature = 97 C Humidity = 85 % Moisture =92 deg c to IBM Watson
Command received: motoron
motor is on
published Temperature = 92 C Humidity = 91 % Moisture =36 deg c to IBM Watson
published Temperature = 108 C Humidity = 62 % Moisture =79 deg c to IBM Watson
published Temperature = 110 C Humidity = 99 % Moisture =103 deg c to IBM Watson
published Temperature = 95 C Humidity = 98 % Moisture =34 deg c to IBM Watson
Command received: motoroff
motor is off
```

Ln: 7502 Col: 0

Type here to search

ENG 01:12 18-11-2022


12:40 AM | 4.7KB/s
28
Measured Data
Temperature(C): 110
Humidity(%): 97
Moisture: 77
Switch Board
Motor On
Motor Off


node-red-hbfsf-2022-11-16.eu-gb.mybluemix.net/ui/#!/0?socketid=NulFPoltHJF0HsjIAAAx
Gmail
YouTube
Maps
Welcome to Project...
Farmer
Measured Data
Temperature
110
105
100
95
90
01:14:32
01:15:02
01:15:46
Humidity
89
0
100
Moistuire
120
100
80
60
40
20
01:14:32
01:15:02
01:15:46
switch board
MOTOR ON
MOTOR OFF
Type here to search
ENG
01:15
18-11-2022

Advantages

- Farms can be monitored and controlled remotely.
- Increase in convenience to farmers.
- Less labor cost.
- Better standards of living.

Disadvantages:

- Lack of internet/connectivity issues.
- Added cost of internet and internet gateway infrastructure.
- Farmers wanted to adapt the use of Mobile App.

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.