

Project Delivery Sprint 4

Date	17 Nov 2022
Team ID	PNT2022TMID04699
Project Name	Smart Farmer - IoT Enabled Smart FarmingApplication

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 Credentials
organization = "95a96q"
deviceType = "NodeMCu"
deviceId = "123456"
authMethod = "use-token-auth"
authToken = "P123@456"

# 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()

# 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)
    data = { 'temp' : temp, 'Humid': Humid }

    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %% " % Humid, "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()

```

- DATA SEND FROM PYTHON PROGRAM :

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 = "i3869j"
deviceType = "abod"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led is off")
    else :
        print ("please send proper command")

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId
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 e
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11

    temp=random.randint(90,110)
```

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\hpl\Downloads\lmsl.py =====
2022-11-19 10:35:42.016 ibmiotf.device.Client INFO Connected successfully: d:95a96q:ModeMCu:123456
Published Temperature = 108 C Humidity = 91 % to IBM Watson
Published Temperature = 99 C Humidity = 68 % to IBM Watson
Published Temperature = 95 C Humidity = 97 % to IBM Watson
Published Temperature = 107 C Humidity = 87 % to IBM Watson
Published Temperature = 104 C Humidity = 82 % to IBM Watson
Published Temperature = 103 C Humidity = 90 % to IBM Watson

- DATA RECEIVED IN IBM CLOUD :

IBM Watson IoT Platform

praveenb.19eca@kongu.edu
ID: 95a96q

Browse Action Device Types Interfaces

Add Device

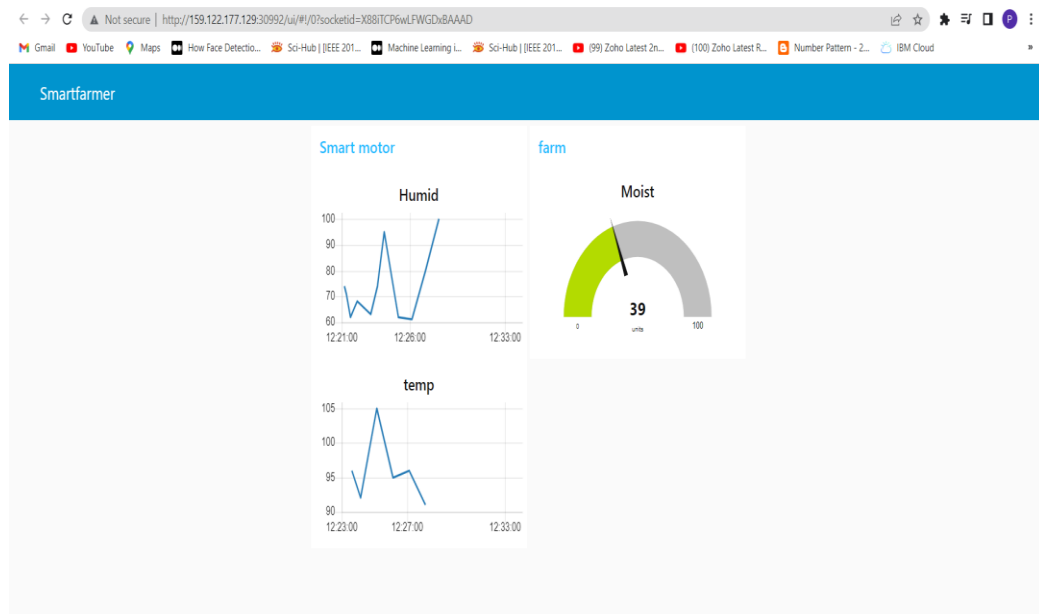
123456 Disconnected NodeMCu Device Nov 15, 2022 9:11 PM

Identity Device Information **Recent Events** State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IoTSensor	{"temp":96,"Humid":79}	json	a few seconds ago
IoTSensor	{"temp":103,"Humid":98}	json	a few seconds ago
IoTSensor	{"temp":104,"Humid":82}	json	a few seconds ago
eventflow	{"randomNumber":17,"temperature":102,"humid":100}	json	a few seconds ago
IoTSensor	{"temp":107,"Humid":87}	json	a few seconds ago

- DATA RECEIVED IN NODE – RED DASHBOARD (WEB UI)



- DATA RECEIVED IN MOBILE APP

The screenshot shows the login screen of the Smart Farmer mobile app. The background is a vibrant green with a large, detailed leaf graphic. The app's logo, 'Smart FARMER', is prominently displayed in the upper center. Below the logo, there are two input fields: 'USER NAME:' with the text 'praveen_b' entered, and 'PASSWORD:' with masked characters '.....'. A 'SUMBIT' button is located at the bottom center of the form area.

- COMMAND RECEIVED FROM WEB UI AND MOBILE APP
 - MOTOR ON/OFF COMMAND



```

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: D:\IBM PROJECT\python 3.7\ibmiotpython.py =====
2022-11-14 14:22:24,419 ibmiotf.device.Client INFO Connected successfully: d:p2cfk6:SMART:15
Published Temperature = 68 C Humidity = 66 % Soil Moisture = 78 % to IBM Watson
Published Temperature = 16 C Humidity = 85 % Soil Moisture = 39 % to IBM Watson
Command received: motoron
motor is on
Published Temperature = 39 C Humidity = 32 % Soil Moisture = 75 % to IBM Watson
Command received: motoron
motor is on
Published Temperature = 48 C Humidity = 21 % Soil Moisture = 5 % to IBM Watson
  
```

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: D:\IBM PROJECT\python 3.7\ibmiotpython.py =====
2022-11-14 14:22:24,419 ibmiotf.device.Client INFO Connected successfully: d:p2cfk6:SMART:15
Published Temperature = 68 C Humidity = 66 % Soil Moisture = 78 % to IBM Watson
Published Temperature = 16 C Humidity = 85 % Soil Moisture = 39 % to IBM Watson
Command received: motoron
motor is on
Published Temperature = 39 C Humidity = 32 % Soil Moisture = 75 % to IBM Watson
Command received: motoron
motor is on
Published Temperature = 48 C Humidity = 21 % Soil Moisture = 5 % to IBM Watson
Published Temperature = 9 C Humidity = 29 % Soil Moisture = 44 % to IBM Watson
Published Temperature = 85 C Humidity = 64 % Soil Moisture = 17 % to IBM Watson
Command received: motoroff
motor is off
Published Temperature = 12 C Humidity = 43 % Soil Moisture = 94 % to IBM Watson
Command received: motoroff
motor is off
Published Temperature = 72 C Humidity = 86 % Soil Moisture = 0 % to IBM Watson
Published Temperature = 100 C Humidity = 95 % Soil Moisture = 90 % to IBM Watson
|
```

ADVANTAGES

- Less labour cost.
- Field can be monitored the environment parameters and controlled the motor remotely.
- Better standards of living.
- Farmers can also monitor and control the farm field by Web UI.
- Increase in convenience to farmers.

DISADVANTAGES

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

CONCLUSION

Thus, the objective of the project is to implement an IOT system in order to help farmers to control the motor function and monitor the environment parameters like temperature, humidity and soil moisture of their farms has been implemented successfully.