

IOT ENABLED SMART FARMING APPLICATION.

The Python Code

TEAM ID : [PNT2022TMID37492](#)

Python IDE

- Install Python3 compiler
- Install any python IDE to execute python scripts, in my case I used Spyder to execute the code.



Code:

```
import time
```

```
import sys
```

```
import ibmiotf.application
```

```
import ibmiotf.device
```

```
import random
```

#Provide your IBM Watson Device Credentials

```
organization = "157uf3"
```

```
deviceType = "abcd"
```

```
deviceId = "7654321"
```

```
authMethod = "token"
```

```
authToken = "87654321"
```

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,"authmethod":authMethod,"authtoken":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 IoT")

time.sleep(10)

deviceCli.commandCallback = myCommandCallback


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