Project Report

ON

Smart Agriculture System

based on IoT

Name: Anup Sharma

Internship: Smart Agriculture system based on IoT by SmartInternz

CONTENTS

1. Introduction

1.1 Overview

Basic factors affecting plant growth are sunlight, water content in soil, temperature, humidity, etc. India is a vast country where the atmospheric conditions keep on varying as we move along it's longitude or latitude & with these varying conditions the agricultural practices also change. Some areas require more water & water distribution cycles as compared to others depending upon the soil moisture as well as it's retention capacity. So what we propose to do through this project is to provide the user with a real time monitoring service measuring the current weather, humidity & soil moisture which ultimately helps the farmer to be fully aware of the conditions in his field & handling these can help him boost agriculture yields.

I have been assigned with the project "Smart Agriculture System Based On IOT" which as the name of the project signifies is built upon the IOT platform. An exception to the normal IOT projects, this project has been solely developed on the IBM cloud platform in collaboration with the smartinternz team for the RSIP 2020 event. This project derives an inspiration from the agricultural practices that are widely practiced throughout the country.

1.2 Proposed Solution

IoT is based on SMART AGRICULTURE SYSTEM is regarded as the IoT gadget focusing on Live Monitoring of Environmental data in terms of Temperature, Moisture and Humidity of atmosphere and the plant/crop. The system provides the concept of “Plug and Sense” in which farmers can directly implement smart farming by such as putting the System on the field and getting Live Data feeds on various electronic devices using Web Application. Moreover, the data generated via sensors can be easily shared and viewed by agriculture consultants anywhere remotely via Cloud Computing technology integration. The system allows manually to turn the pumping motor ON and OFF on sensing the moisture content of the soil.

Need of smart agriculture irrigation:

- Saving energy and resources, so that it can be utilized in proper way.
- Farmers would be able to smear to right amount of water at the right time by

automatic irrigation.

- Automated irrigation system uses values to turn motor ON and OFF. Motors can be automated easily by using controllers and no need of labor to turn motor ON and OFF.
- it measures accurate soil moistures, temperature, humidity.
- It is time saving, human error elimination in adjusting soil moisture levels.
- High reliability, security, compatibility of technical requirement, scalability.

LITERATURE SURVEY

2.1 Existing problem

Experts have analyzed collected data for finding correlation between environment work and yield for the standard work they are concentrated on crop monitoring, information of temperature and rainfall is collected as initial spatial data and analyzed to reduce the crop losses and to improve the crop production.
An IOT based agriculture monitoring system explains to monitor a crop field. A system is developed by using sensors and according to the decision from a server based on sensed data. By using the wireless transmission, the sensed data forwarded to web server database. The user can monitor and control the system remotely with the help of application which provides the web user. IOT based agriculture monitoring system develops various features like monitoring, temperature, humidity, moisture sensing.

Another problem that was identified is that for the farmers whose houses are farther from the fields find trouble in regularly getting up to the fields only to find out that conditions on that day are not particularly suitable & they are not equipped with any system to remotely monitor the weather conditions as well as no remote control for controlling the irrigation motors remotely.

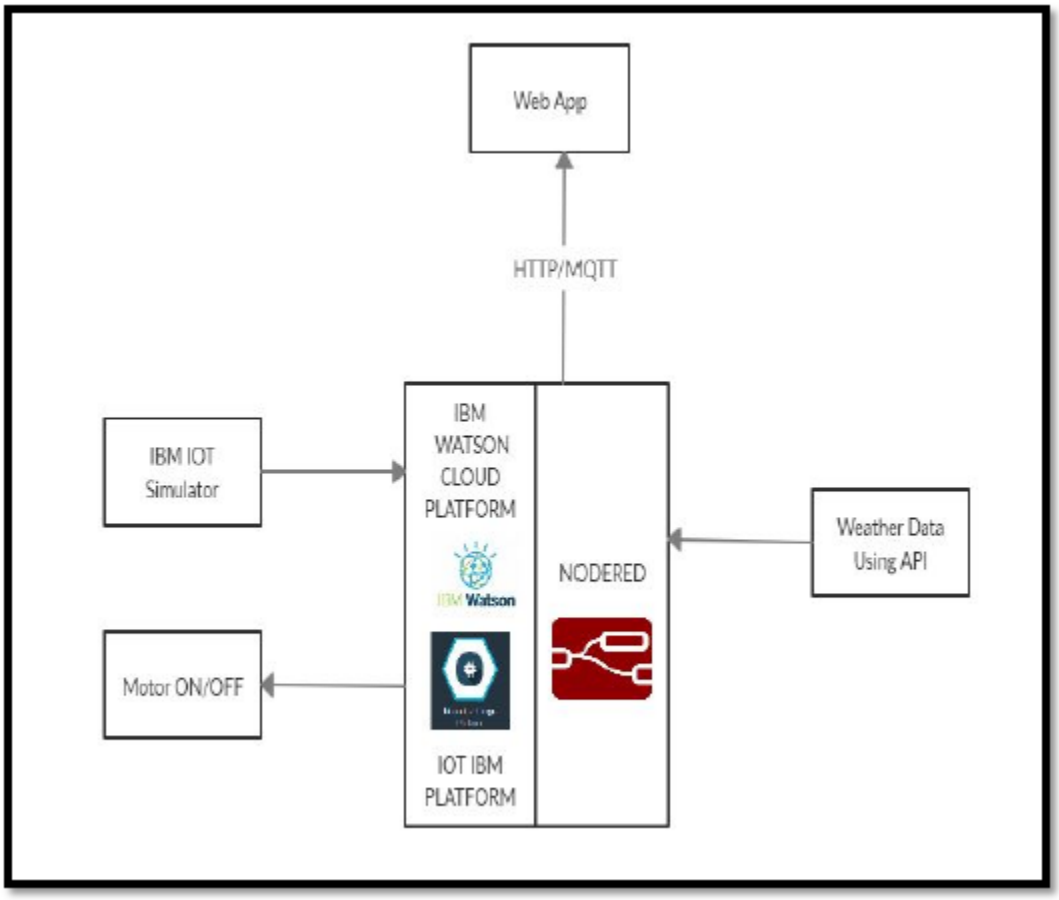
2.2 Proposed Solution

The solution is that using IOT for monitoring the agriculture through the webserver. The system development composes parts: the server, PC client, open weather API, sensor, IBM cloud platform. The mobile application will be developed in android. It helps to monitor an controlled field from anywhere.

● The supply of water can be controlled from anywhere by controlling the motor state (ON/OFF), using web application.
● Real time weather conditions can also be checked using the open weather API's from different websites and it is displayed on our web app.
● soil moisture and other parameters can be checked by using the Watson IoT sensor and send the data to cloud services& to the web app.
● our surrounding temperature can be checked by the sensor &controlled easily.

THEORITICAL ANALYSIS

3.1 Block diagram



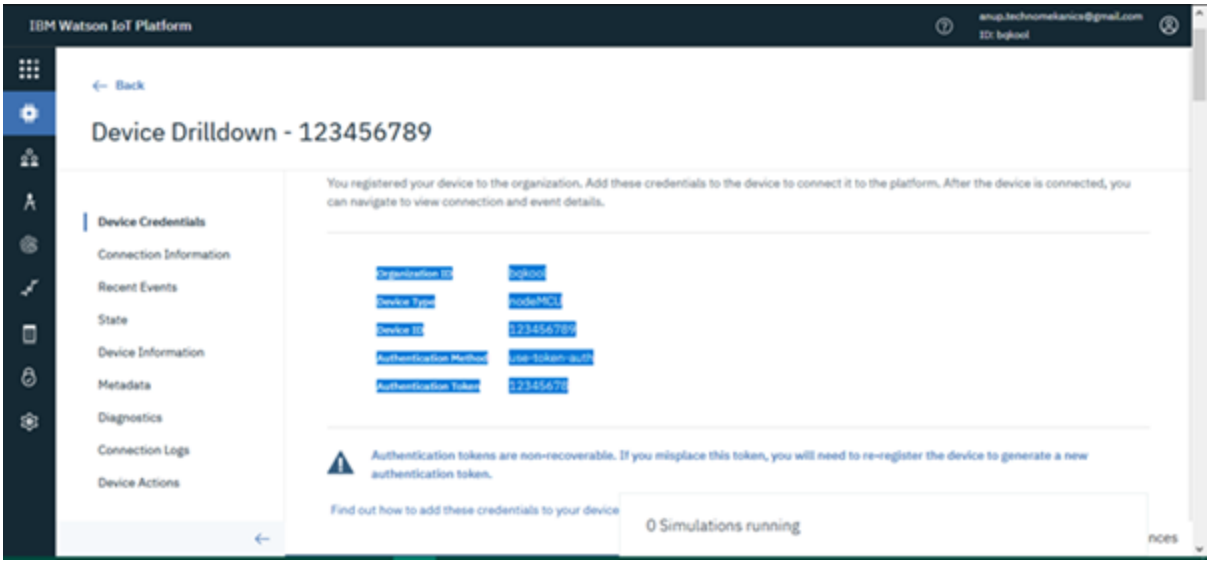
3.2 Hardware / Software designing

This project does not involve any kind of hardware & is solely based upon software designing. The software for this project is designed on the IBM cloud service namely the IBM IOT Watson platform apart from some other helper services from the IBM cloud itself. The project involves

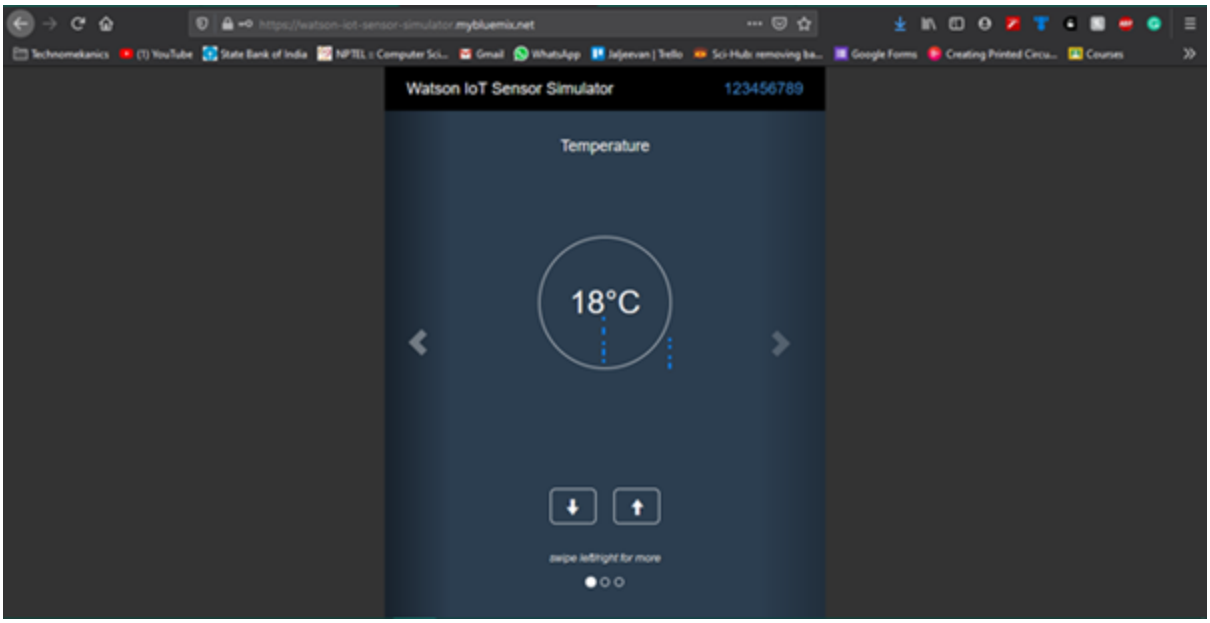
creating two virtual devices one on the user's end other on the receiver end. The user end device is then connected to the IOT simulator which sends the weather conditions info. back to the device. It is further connected through the openweather api to connect the city concerned with the user. This setup ensures regular transfer of weather data at regular intervals to the device. After this node red service from IBM cloud came into use & proper connections were layed out for building as well as debugging the web app. After this the second device is configured to work with the python code for controlling the motors & the nodered app as a whole is configured to work in one single flow & respond to motor on & motor off controls.

4. EXPERIMENTAL INVESTIGATIONS

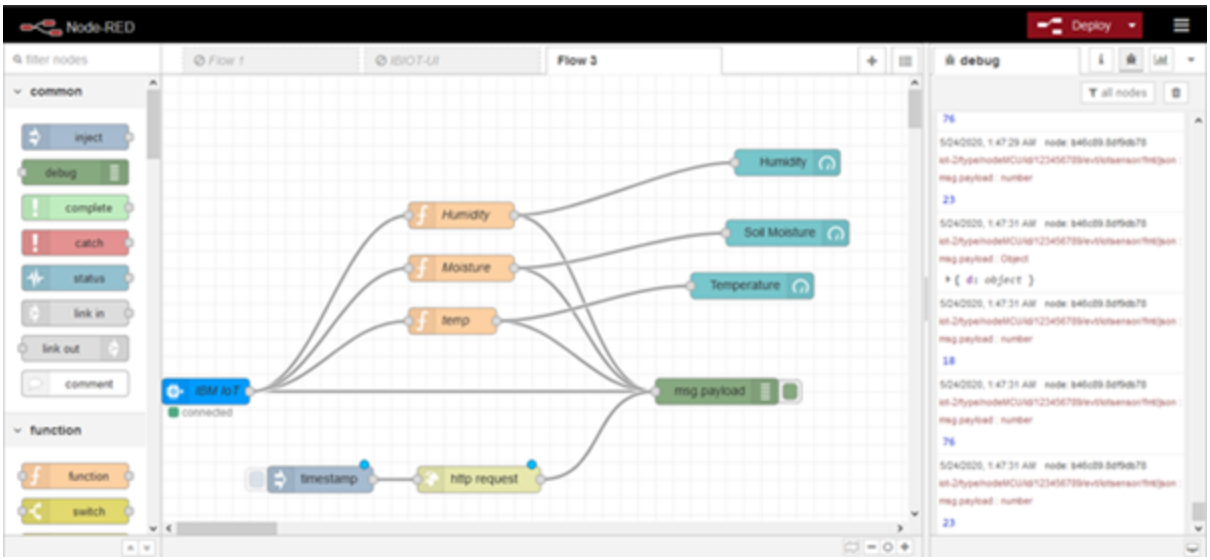
This project was developed on the IBM IOT platform that is a service provided by the IBM Cloud. Then the first device was created (which is the one going to be used by the farmer for sending commands).



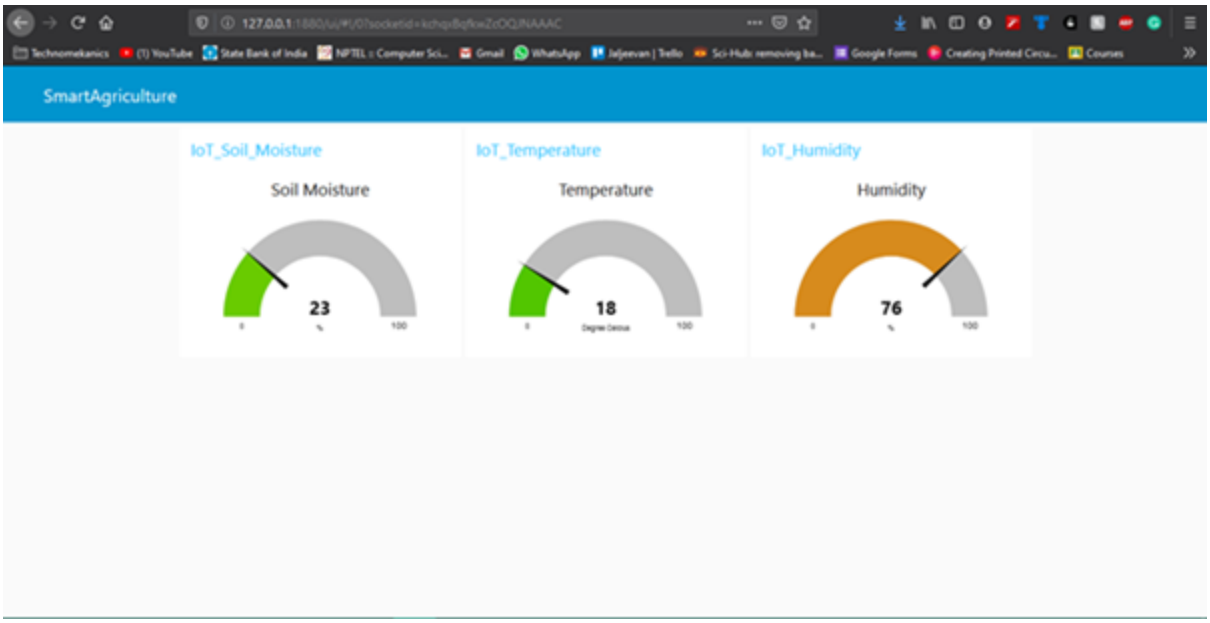
This device was then connected to the Watson IOT simulator by linking the device credentials. Through this the device was connected to the cloud receiving data remotely.



I then configured the node red app enabling to receive data from the iot platform. It was linked with the open weather api key to get live weather details for the desire. with this the app got its first functionality of monitoring ready & working.

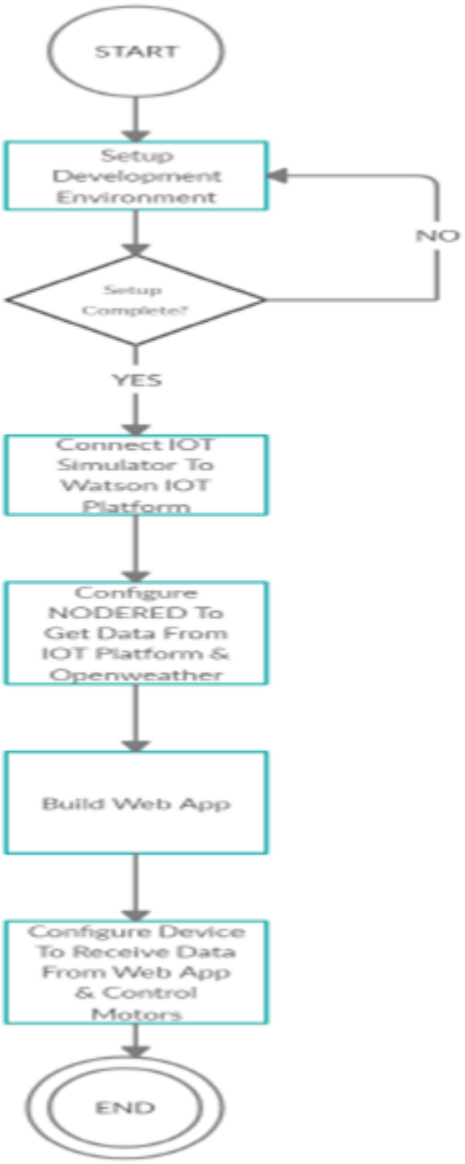


After this to add the second functional the nodes were configured as well as button added to control the motors remotely. The dashboard functionality was added finally finishing the app.



The final step was to write the python code to integrate all the mid products into single streamlined final working product.

5. FLOWCHART



6. RESULT

By following the above-mentioned steps, we create a basic smart agriculture which can help us the farmer to know the temperature, Moisture and humidity also can turn on and off the

MOTOR from his house. We have successfully created the smart agriculture system using IBM Watson Assistant, IBM Cloud Function, IBM Watson and Node-Red.

7. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. User can monitor the weather conditions remotely as well.
2. User can easily irrigate his fields with a single tap.
3. Save time
4. Save Money

DISADVANTAGES:

1. The app is not a standalone product and requires an external physical smart device to run it as well as various sensors as well.
2. Apart from its own cost the app may incur other charges to the farmer which may not be affordable to him (data charges, device charges)
3. The app may not show accurate data when the atmospheric conditions keep changing rapidly (like in the monsoons)

8 APPLICATIONS:

1. Smart Agriculture System based on IOT app can monitor soil moisture & climatic conditions to grow & yield a good crop
2. Farmer can get real time weather forecasting data using this app
3. Based on all the parameters he can water his crop by controlling the motors using the app
4. He can also water the fields remotely using this app.

9 CONCLUSION

The proposed model explores the use of IOT in agriculture sector. This model aims at increasing the crop yield by helping in predicting better crop sequence for a particular soil. Data on cloud helps the agriculturists in improving the yields. This system is cost effective and feasible. This system will sense all environmental parameters and send the data to the user via cloud. It leads to higher crop yield, better quality, and less use of protective chemicals.

10 FUTURE SCOPE:

The future scope of this project could be including variety of soil sensors like pH sensor, rain sensor and then collecting and storing the data on server. This would make the predicting and analyzing processes more accurate.

11. BIBLIOGRAPHY:

● IBM Cloud:
https://cloud.ibm.com/docs/overview?topic=overview-whatis-platform

● Watson IOT:
https://www.iotone.com/software/ibm-watson-iot-platform/s62

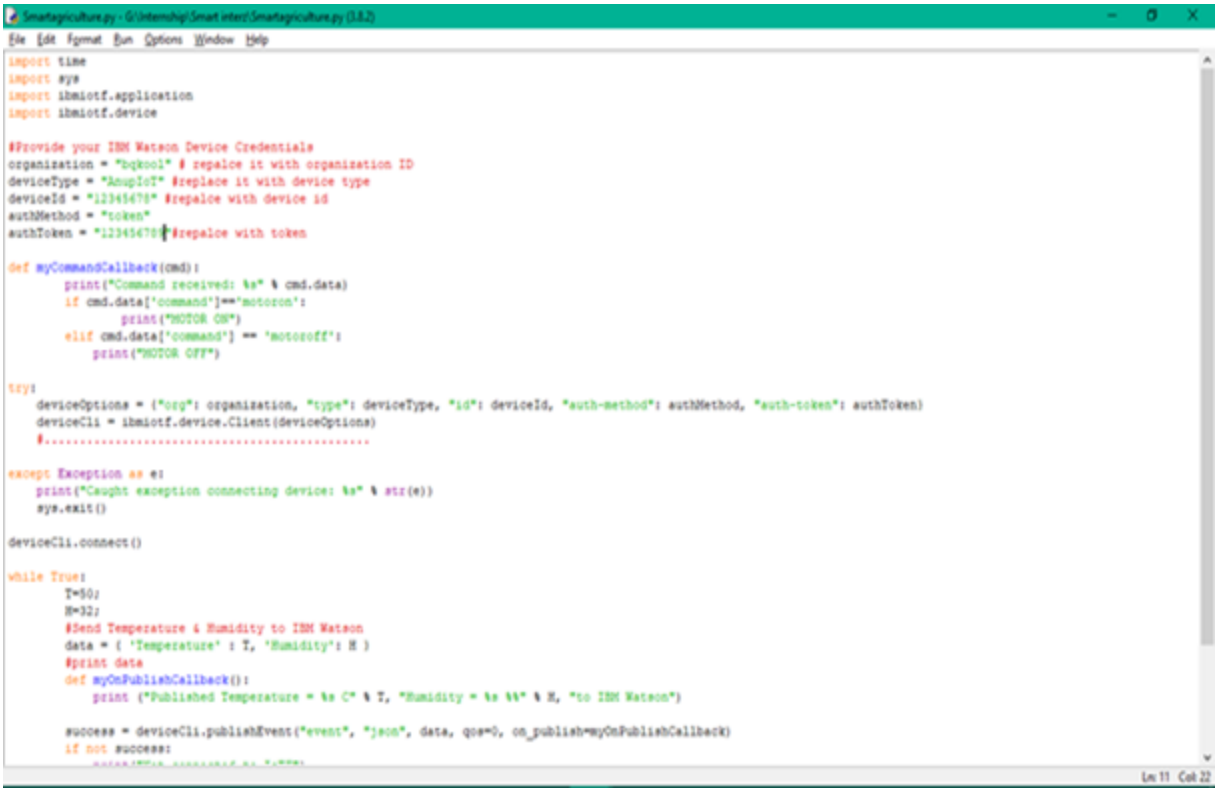
● Node-RED:
https://nodered.org/docs/getting-started/windows#3-run-node-red

● Openweathermap:
https://openweathermap.org/

● Github:
https://github.com/rachuriharish23/ibmsubscribe

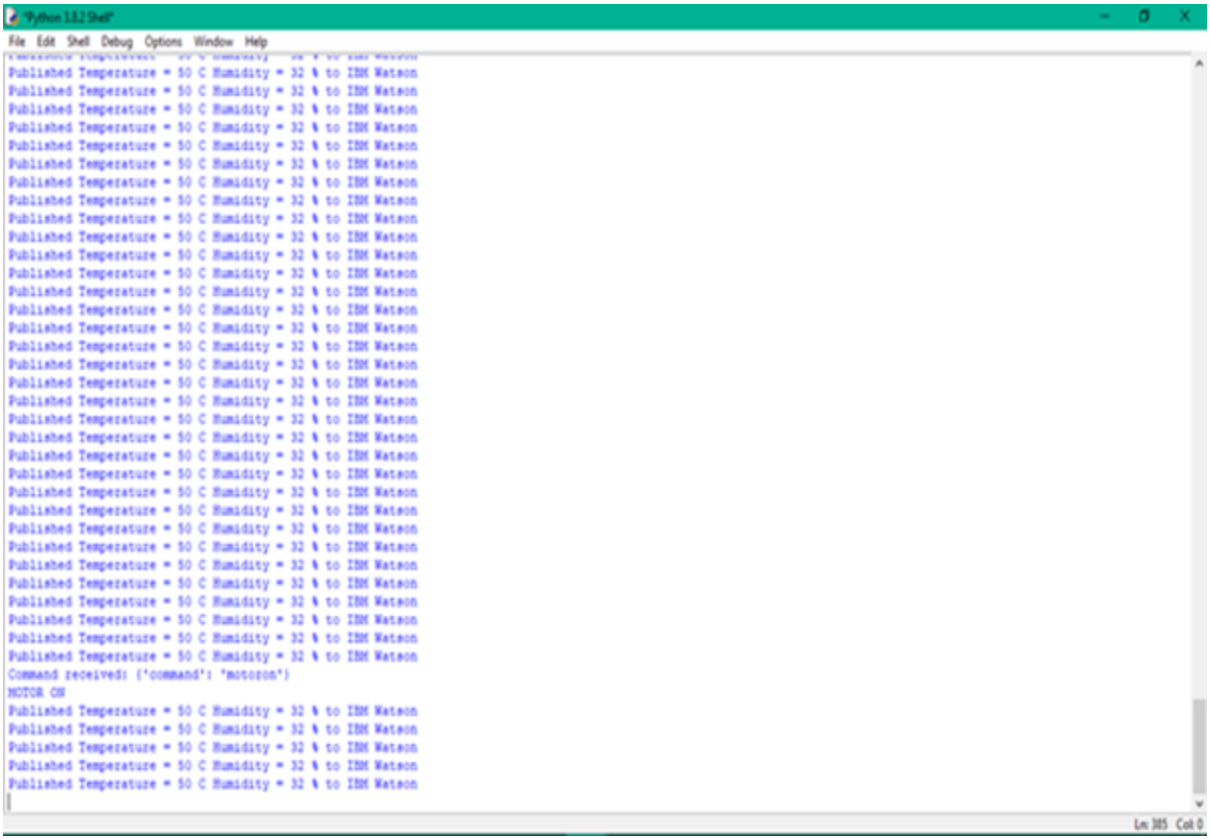
● Watson iot simulator :
https://watson-iot-sensor-simulator.mybluemix.net

12 SOURCE CODE



```
Smartagriculture.py - G:\Internship\Smart intern\Smartagriculture.py (3.8.2)
File  Edit  Format  Run  Options  Window  Help
import time
import sys
import ibmiotf.application
import ibmiotf.device

#Provide your IBM Watson Device Credentials
organization = "bgkool" # repalce it with organization ID
deviceType = "AnupIoT" #replace it with device type
deviceId = "12345678" #repalce with device id
authMethod = "token"
authToken = "12345678" #repalce with token

def myCommandCallback(cmd):
        print("Command received: %s" % cmd.data)
        if cmd.data['command']=='motoron':
                print("MOTOR ON")
        elif cmd.data['command'] == 'motoroff':
            print("MOTOR OFF")

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:
        T=50;
        H=32;
        #Send Temperature & Humidity to IBM Watson
        data = { 'Temperature' : T, 'Humidity': H }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % T, "Humidity = %s %%" % H, "to IBM Watson")

        success = deviceCli.publishEvent("event", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
                                                                                    Ln: 11  Col: 22
```

13. FINAL OUTPUT



THANK YOU