7/4/2020

Smart Agriculture System Based On IoT

Done By:
Dhaval Mehta
Submitted To:
Smartbridge

Index

Introduction

Overview

Agriculture is the main backbone of India. Agriculture required the dedication of various normal resource counting land, water, and natural condition.The quality what's more, measure of trademark resource has spoiled all through the a long time in view of fiscal issues related with extended expense of data and lessening farm compensation continually declining land, work, assets, and ecological issue, for instance, soil .What's more, water tainting putting the reasonableness without limits agriculture activity at possibility.

The most significant obstruction that emerges in customary cultivating is environmental change. The quantity of impacts of climate change incorporates overwhelming precipitation,most serious tempest and warmth waves, less precipitation and so on because of these the efficiency of cultivation declines to the significant degree. Environmental change likewise raises the natural outcomes, for example, the occasional change in the life cycle of the plant. To help the profitability and limit the hindrance in agricultural field we have to utilize inventive innovation and method and technologies called Internet of things.

The innovative advances in these field enables to develop momentum in agriculture and enables the farmers to easily cultivate there land without climatic hindrances using advanced machineries and technologies.This is called Smart Agriculture.The most important aspect of using smart agriculture is water management and environmental measurements both of which play a vital role in plant's lifecycle.Water management is done using actuators while environmental measurement is done using sensors/simulators.Based on the readings obtained from simulators farmer can control the actuators i.e. motors remotely.

The project aims at making agriculture smart using cloud and IOT technologies. The highlighting features of this project include smart agriculture with smart control based on real time field data. Secondly temperature maintenance, humidity maintenance and other environmental parameters. And finally the recommendation to farmer for smart agriculture

Purpose

Smart Agriculture with help of sensors and actuators benefits the society in many ways.Some of them are:

1. Conservation of Water
2. Adequate utilization of Resources
3. Better Yield
4. Less Tedious For Farmers
5. Diminishes human errors
6. Prevents Pollution
7. Remote Access
8. Time Efficient

In short,Smart Agriculture involves four main steps:
1)Data Gathering
2)Data Processing
3)Data Analysis
4)Data Automation i.e. Output

Literature Survey

Existing Solution

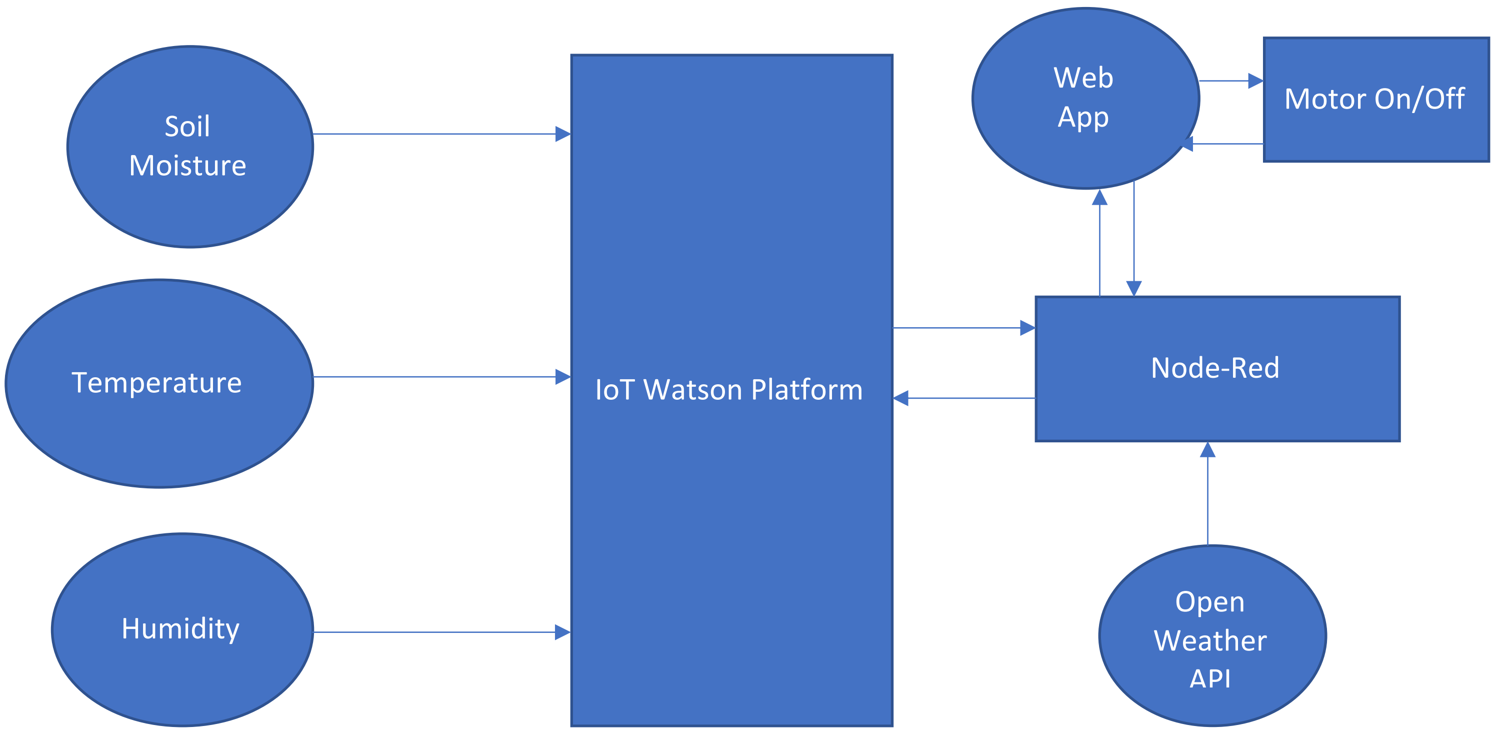
The situation of diminishing water tables, evaporating of waterways and tanks, flighty condition present an earnest need of appropriate use of water. To adapt up to this utilization of temperature and moisture sensors at reasonable areas for checking of yields is executed. A calculation created with edge estimations of temperature and soil moisture can be modified into smaller scale controller based entryway to control water amount. The framework can be fueled by Photo voltaic boards(Arduino or Raspberry Pi) and can have duplex correspondence connect dependent on cell – Internet interface that permit information review and water system booking to be modified through web page.The innovative advancement in open source programming and equipment make it simple to build up the gadget which can improve observing

Proposed Solution

In the proposed solution,we are going to use Watson iot simulators that are going to act as sensors that would measure the temperature,moisture and humidity.Based on the data received from this device,the data would be sent over cloud so that the user can view the readings.Then based on this data the user would decide to switch on the motor or switch off the motor so as to remotely irrigate the farm. This communication flow would be controlled by Node-Red.

Theoretical Analysis

Block Diagram



Hardware/Software Requirements

We would be requiring following software for working on our projects:

1)IBM Cloud Platform – the account is must for it would be used to exchange data between web app and sensors

2)IBM Watson IoT Platform – here the devices would be created for communication. The Watson IoT Device Simulator is a solution that that enables customers to create and simulate hundreds of virtual connected devices, without having to configure and manage physical devices, or develop time-consuming scripts

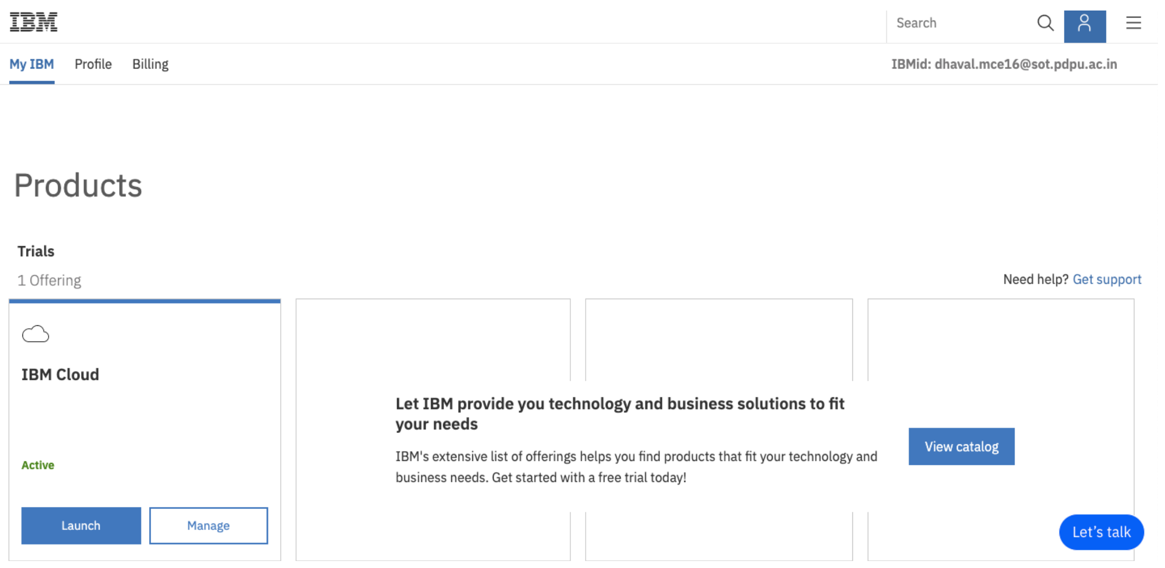
3)OpenWeatherPlatform API – this is the api that would provide the web app with real time weather conditions

4)Node-Red – this is the platform that would connect the web app and cloud.From here data would be transferred to web app and command from user would be received to monitor the working of actuators.

5)Python IDE – here the code would be written to monitor the commands sent by the user. Basically code would be to subscribe the services.

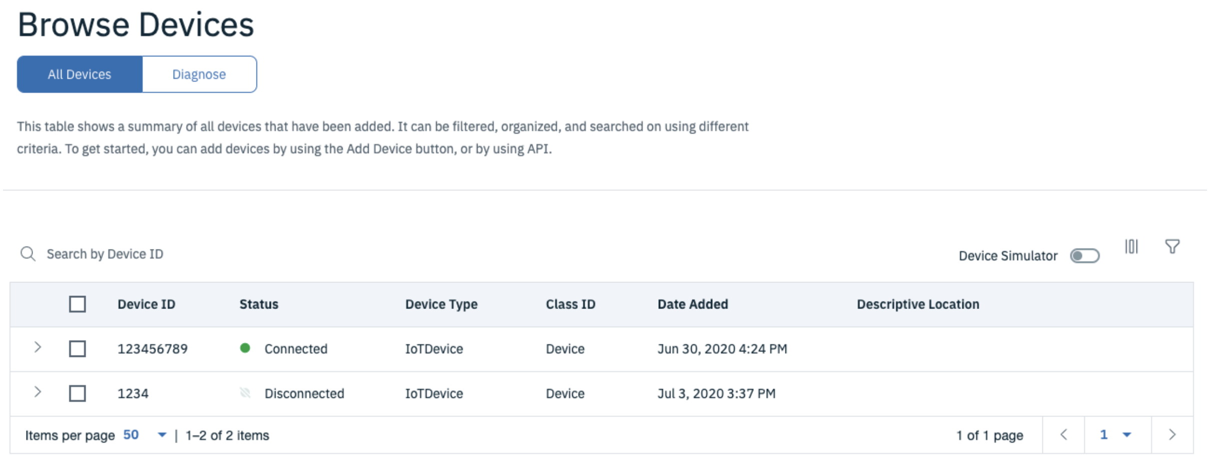
Experimental Investigations

First we create account in IBM cloud.After that we go to the IBM Watson IoT Platform.In IBM Watson IoT Platform we create the devices that would receive the data from the sensors i.e. simulators.This is the data that is going to be published in web app.


IBM
Search
My IBM
Profile
Billing
IBMid: dhaval.mce16@sot.pdpu.ac.in
Products
Trials
1 Offering
Need help? Get support
IBM Cloud
Active
Launch
Manage
Let IBM provide you technology and business solutions to fit your needs
IBM's extensive list of offerings helps you find products that fit your technology and business needs. Get started with a free trial today!
View catalog
Let's talk

IBM ACCOUNT CREATION

After creation of device,we get it's metadata like it's authentication,device id,device type, organization which are necessary for it's validation in node-red.



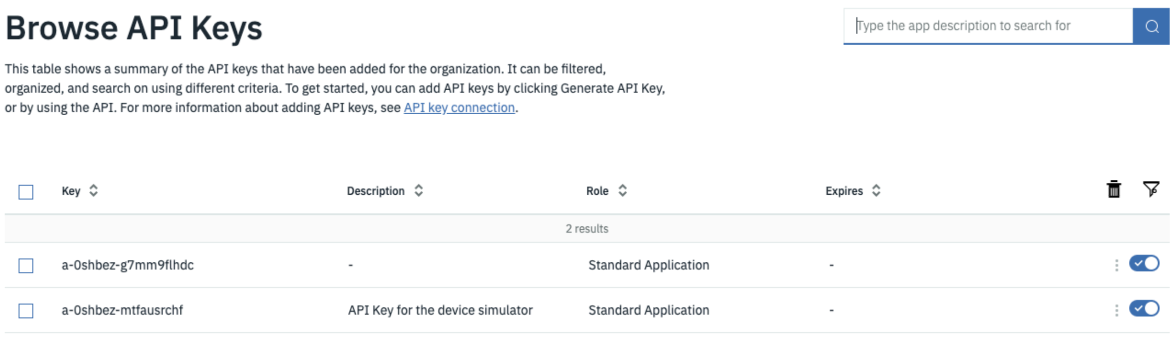
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
123456789	● Connected	IoTDevice	Device	Jun 30, 2020 4:24 PM	
1234	Disconnected	IoTDevice	Device	Jul 3, 2020 3:37 PM	

Items per page 50 | 1–2 of 2 items 1 of 1 page 1

Device Manager

After creating the device we create the API key so that the simulator and IBM IoT Platform can communicate



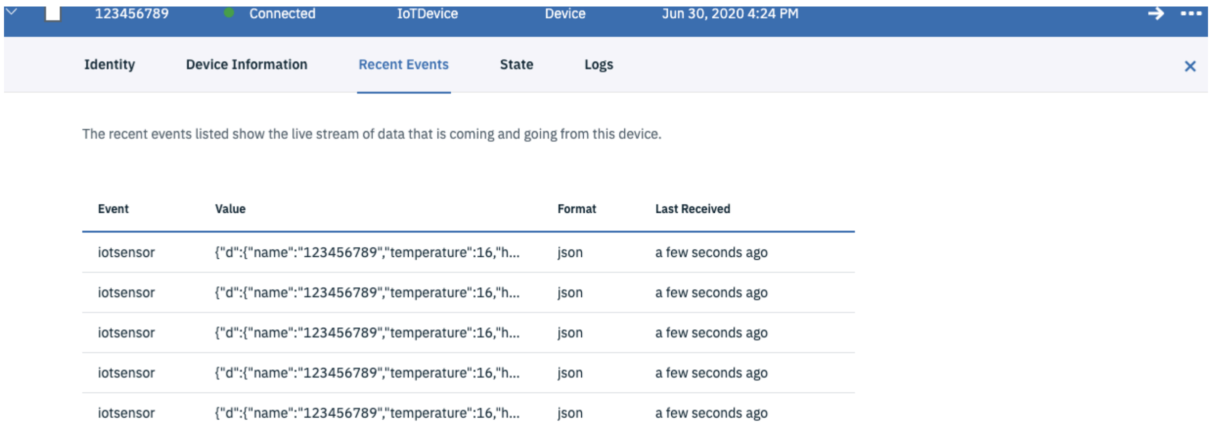
Browse API Keys

This table shows a summary of the API keys that have been added for the organization. It can be filtered, organized, and search on using different criteria. To get started, you can add API keys by clicking Generate API Key, or by using the API. For more information about adding API keys, see API key connection.

Key	Description	Role	Expires
a-0shbez-g7mm9flhdc	-	Standard Application	-
a-0shbez-mtfausrchf	API Key for the device simulator	Standard Application	-

API Keys

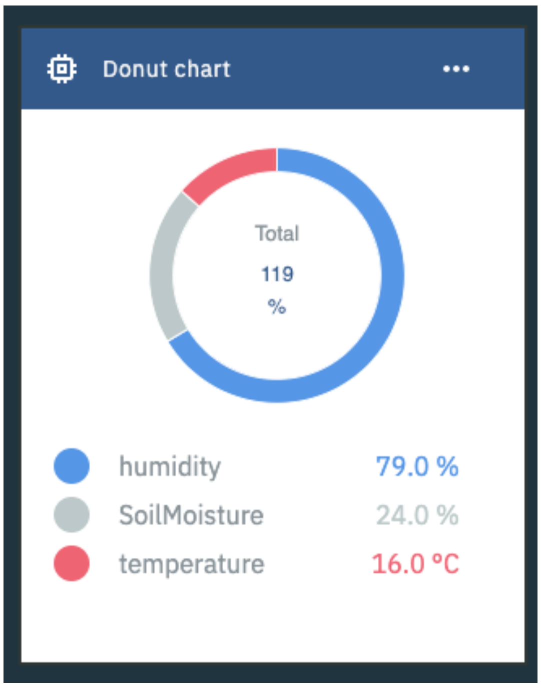
After this, we go to Watson IoT Simulator Platform which consists of temperature,humidity and soil moisture simulators which would send the data to the IBM IoT Platform.The metadata that we received while creating the device would be used here so that the device gets connected to the platform and starts sending it's realtime data.The data can be viewed in boards we create in the IBM IoT Platform.

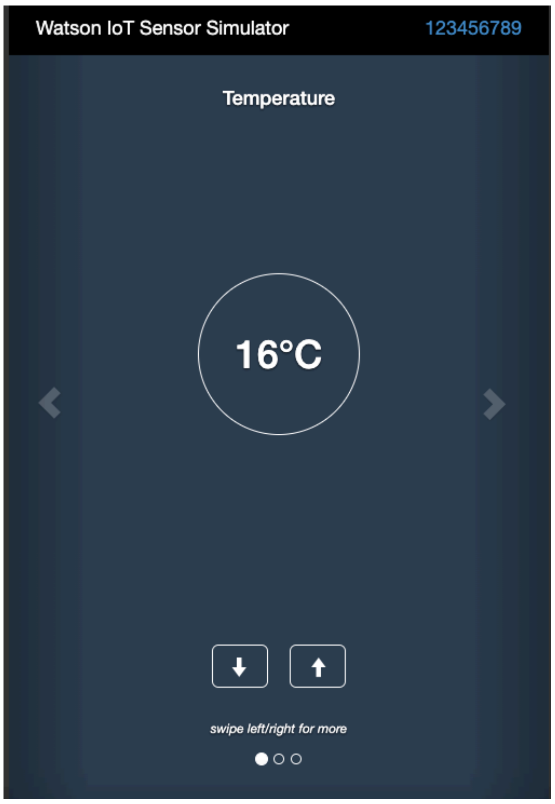


The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
iotsensor	{"d":{"name":"123456789","temperature":16,"h...	json	a few seconds ago
iotsensor	{"d":{"name":"123456789","temperature":16,"h...	json	a few seconds ago
iotsensor	{"d":{"name":"123456789","temperature":16,"h...	json	a few seconds ago
iotsensor	{"d":{"name":"123456789","temperature":16,"h...	json	a few seconds ago
iotsensor	{"d":{"name":"123456789","temperature":16,"h...	json	a few seconds ago

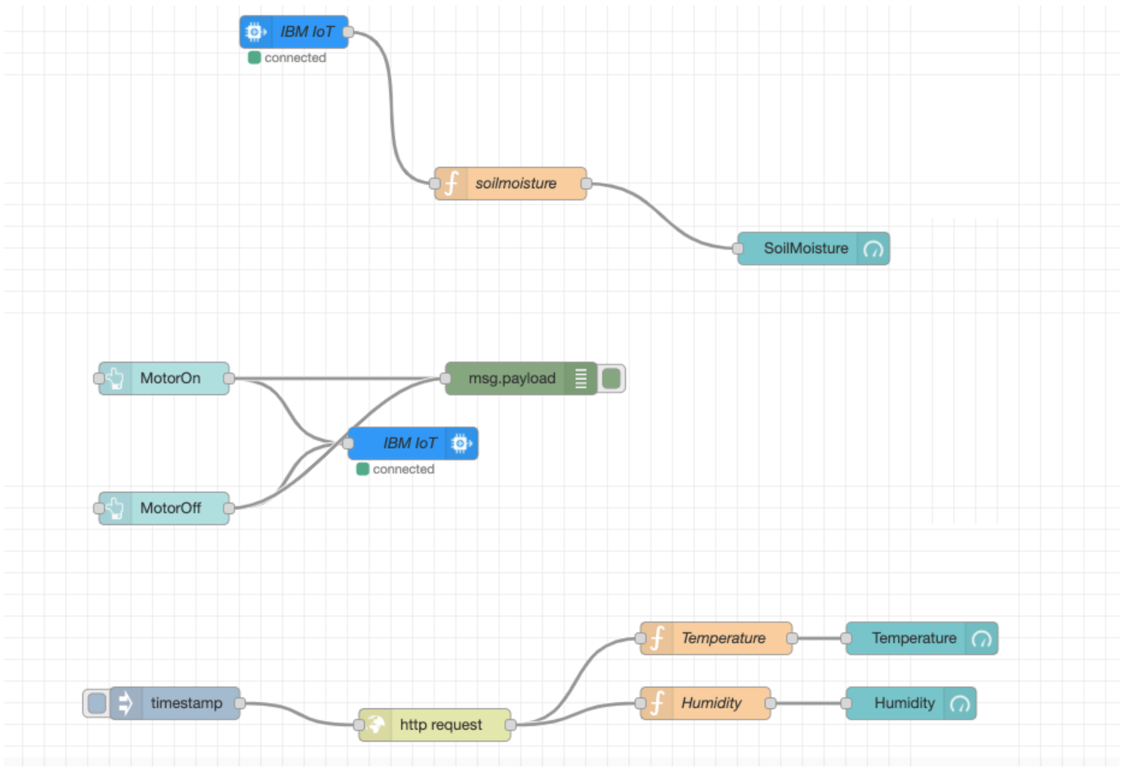
Data received from simulator



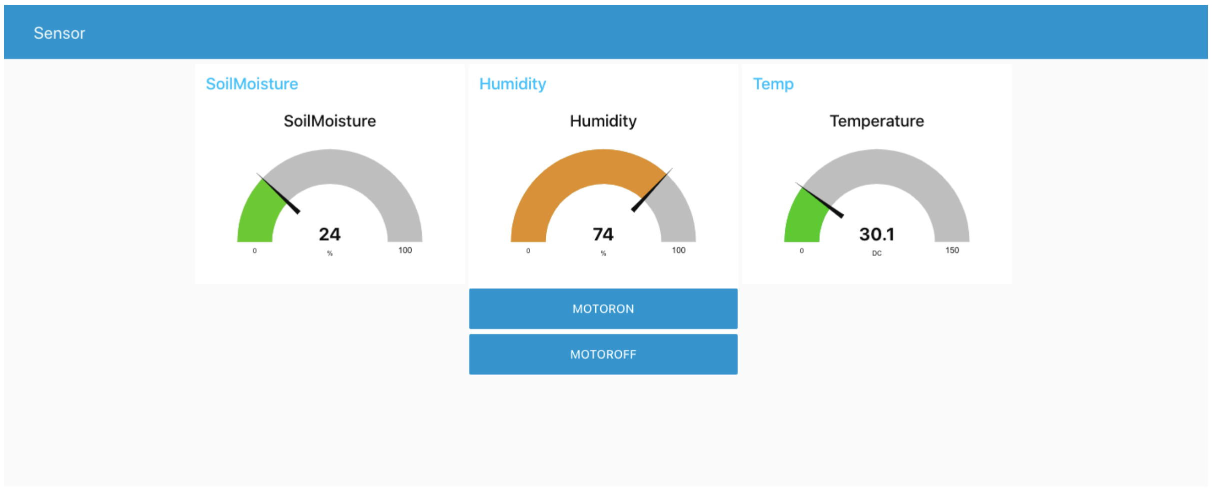


Watson IoT Sensors

After that in node-red,we install the required nodes like IBMIOT nodes and dashboard nodes to design the User Interface.In Node-red we create the flow that would read the sensor data from IBM IoT Platform and send to the web app user interface that would contain the appropriate ui to view the environmental data and based on which the farmer would either switch on/off the motor remotely.The decision of the farmer would then be send back using the node-red flow to the IBM IoT Platform.Thus this way using sensors and actuators the farmer could use advanced and modern technologies thus saving his time and efforts.The real weather conditions like temperature and humidity would be accessed by making http request to OpenWeatherAPI while the soil moisture conditions would be controlled by the soil sensor.

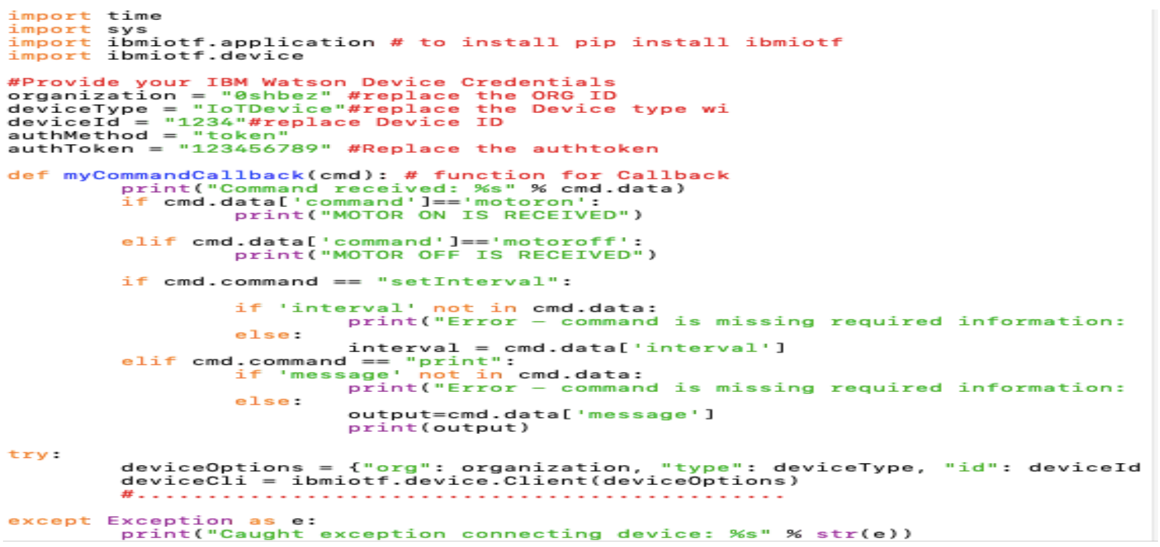


Node-Red Flow for the required smart agriculture



Web App UI

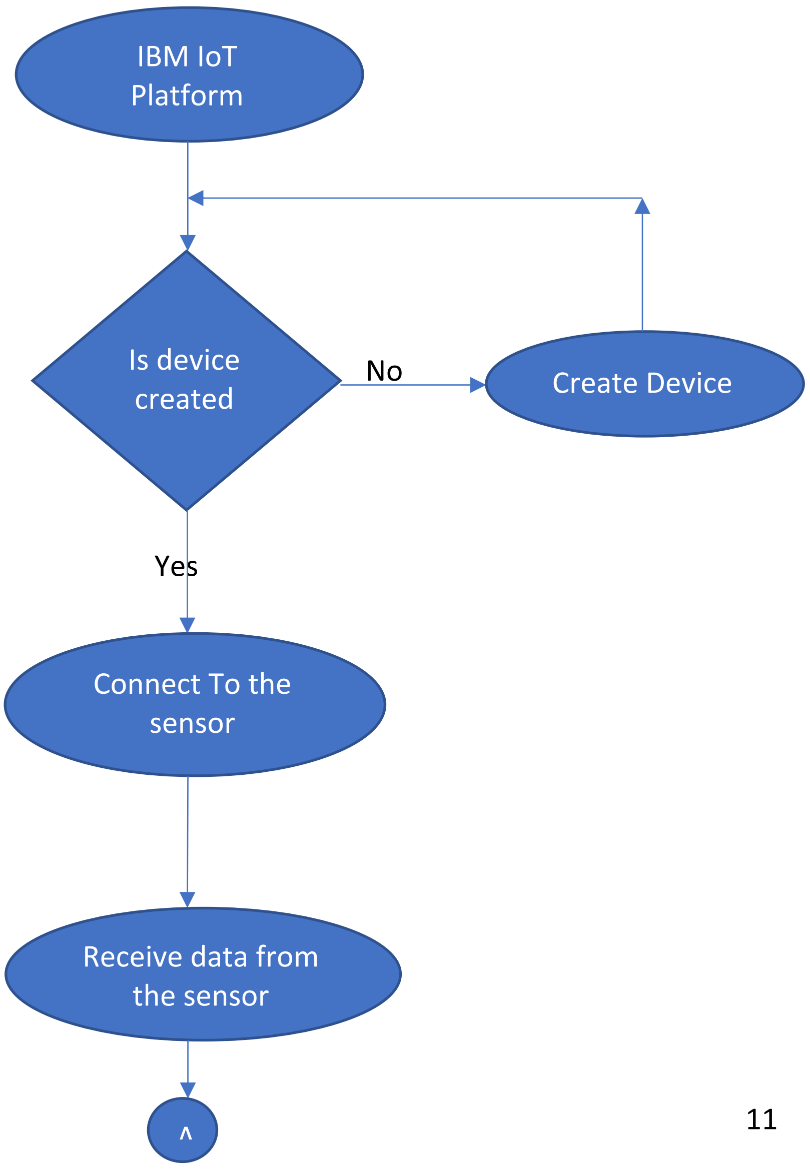
After that we use python code to connect to the device that receives the actuator command so that we know whether our actuator are working or not.What this code would do is it would tell us the decision of the farmer whether he has switched on the motor or closed it.We would need to add the metadata of the device in the code so that it gets connected to the device.

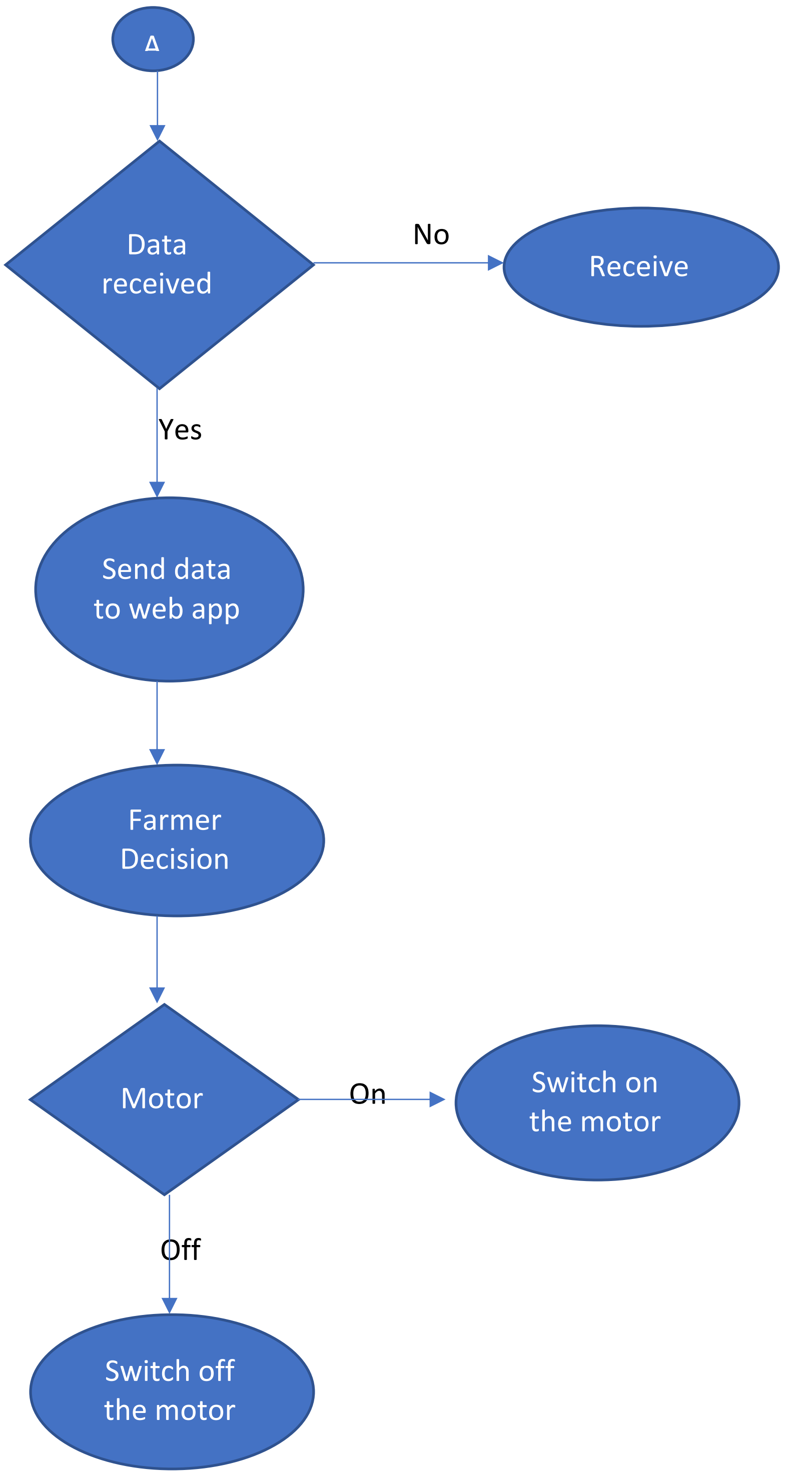


```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device

#Provide your IBM Watson Device Credentials
organization = "0shbez" #replace the ORG ID
deviceType = "IoTDevice"#replace the Device type wi
deviceId = "1234"#replace Device ID
authMethod = "token"
authToken = "123456789" #Replace the authtoken

def myCommandCallback(cmd): # function for Callback
        print("Command received: %s" % cmd.data)
        if cmd.data['command']=='motoron':
                print("MOTOR ON IS RECEIVED")

        elif cmd.data['command']=='motoroff':
                print("MOTOR OFF IS RECEIVED")

        if cmd.command == "setInterval":

                if 'interval' not in cmd.data:
                        print("Error - command is missing required information:
                else:
                        interval = cmd.data['interval']
        elif cmd.command == "print":
                if 'message' not in cmd.data:
                        print("Error - command is missing required information:
                else:
                        output=cmd.data['message']
                        print(output)

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))
```

Metadata In The Code

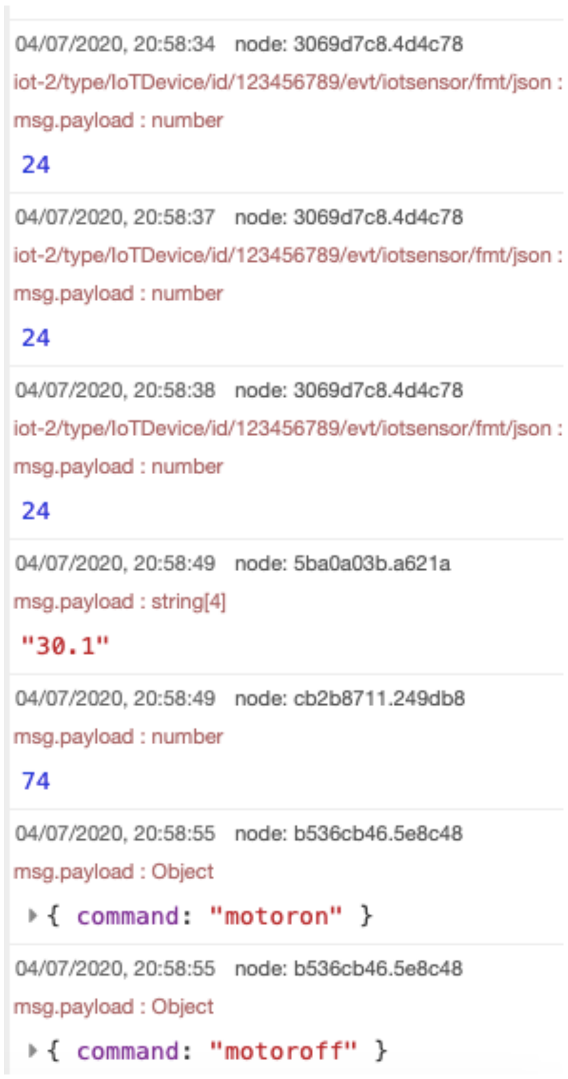
Flowchart




A
Data received
No
Receive
Yes
Send data to web app
Farmer Decision
Motor
On
Switch on the motor
Off
Switch off the motor

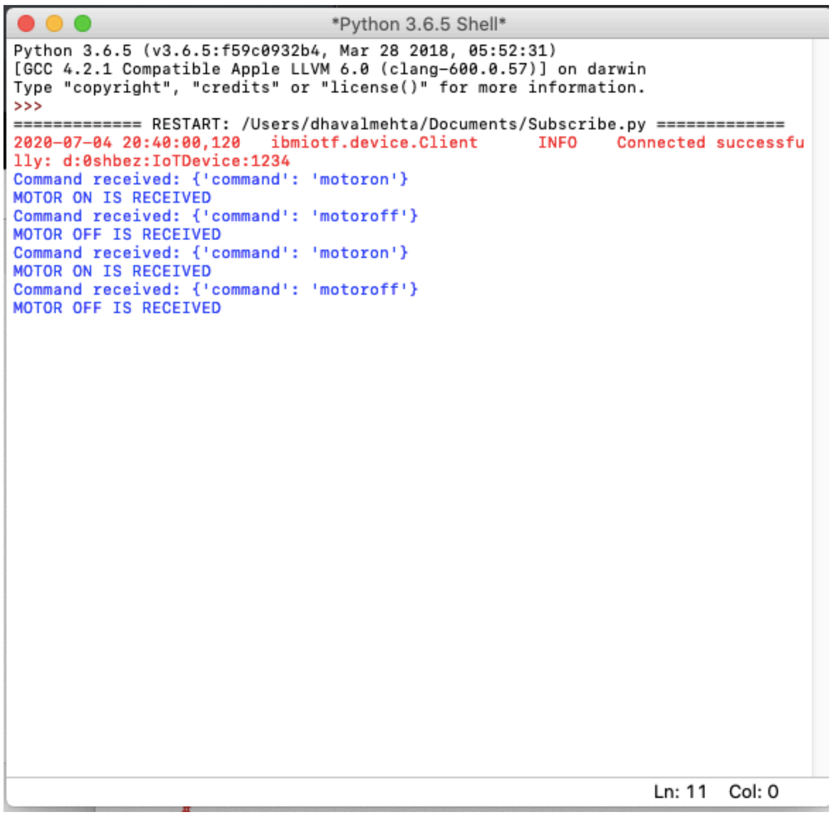
Result

Data received from the simulator is sent to the web app and based on that readings the farmer makes decision whether to switch on the motor or switch it off.That result is sent back to the cloud so that the used can view it's decision and appropriate action is taken.



```
04/07/2020, 20:58:34   node: 3069d7c8.4d4c78
iot-2/type/IoTDevice/id/123456789/evt/iotsensor/fmt/json :
msg.payload : number
24
04/07/2020, 20:58:37   node: 3069d7c8.4d4c78
iot-2/type/IoTDevice/id/123456789/evt/iotsensor/fmt/json :
msg.payload : number
24
04/07/2020, 20:58:38   node: 3069d7c8.4d4c78
iot-2/type/IoTDevice/id/123456789/evt/iotsensor/fmt/json :
msg.payload : number
24
04/07/2020, 20:58:49   node: 5ba0a03b.a621a
msg.payload : string[4]
"30.1"
04/07/2020, 20:58:49   node: cb2b8711.249db8
msg.payload : number
74
04/07/2020, 20:58:55   node: b536cb46.5e8c48
msg.payload : Object
▸{ command: "motoron" }
04/07/2020, 20:58:55   node: b536cb46.5e8c48
msg.payload : Object
▸{ command: "motoroff" }
```

Data received in node-red of farmer as well as simulator



Python 3.6.5 Shell

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 05:52:31)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: /Users/dhavalmehta/Documents/Subscribe.py =============
2020-07-04 20:40:00,120   ibmiotf.device.Client      INFO    Connected successfu
lly: d:0shbez:IoTDevice:1234
Command received: {'command': 'motoron'}
MOTOR ON IS RECEIVED
Command received: {'command': 'motoroff'}
MOTOR OFF IS RECEIVED
Command received: {'command': 'motoron'}
MOTOR ON IS RECEIVED
Command received: {'command': 'motoroff'}
MOTOR OFF IS RECEIVED
```

Ln: 11   Col: 0

Code Result

Advantages of Smart Agriculture

1)Data Collection – All the data is collected using sensors and is stored at one place so farmers can easily access it from that place.

2)Risk Reductions – Farmers can use up-to date data to predict the future yields and also predict the problems.They can use this data to improve their sales.

3)Better yields – Smart agriculture makes it possible to avoid challenges and remove all issues that may arise during farming processes. So the quality of the product is growing and consumers get a good product of high quality

4)Conserve Resources – Resources such as water and soil are preserved as they are used only when required.

5)Remote Monitoring – Local and commercial farmers can monitor multiple fields in multiple locations around the globe from an internet connection. Decisions can be made in real-time and from anywhere.

6)Lower Cost – Automating processes in planting, treatment and harvesting can reduce human error and overall cost

Disadvantages of Smart Agriculture

1)The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.

2)The smart farming based equipments require farmers to understand and learn the use of technology. This is major challange in adopting smart agriculture farming at large scale across the countries.

Applications of IoT in agriculture

1)Precision Farming – Precision farming can be thought of as anything that makes farming practice more controlled and accurate when it comes to raising livestock and growing crops

2)Livestock Monitoring – Cattles in the farm can be traced and used to monitor there health using sensors

3)Smart Greenhouse – Greenhouse farming is a methodology that helps in enhancing the yield of vegetables, fruits, crops, etc. Greenhouses control the environmental parameters through manual intervention or a proportional control mechanism. As manual intervention results in production loss, energy loss, and labor costs, these methods are less effective.So using smart greenhouse with sensors helps in overcoming this drawbacks.

Conclusion

This way with the help of IBM IoT Platform and node-red we can develop a certain aspect of smart agriculture.With the help of sensors the real time data is sent over the cloud to the web app where the user can remotely analyse the data. Based on the data received the user can decide whether it is favourable to switch on the motor or switch it off. This way user’s efforts as well as time are saved as well as unnecessary wastage of resources is reduced.

Future Scope

In future,due to reduced amount of resources and labor,it becomes necessary for a person to reduce the wastage of resources and also save time so that other works can be done. Also manual monitoring often leads to human errors and less efficient yields.Due to which the automation of farming with better equipments becomes a priority.Thus smart agriculture and use of sensors and cloud would be required to ease the workload.

Bibliography

1) https://smartinternz.com/Student/workspace/2876
2) https://www.youtube.com/watch?v=cicTw4SEdxk
3) https://openweathermap.org/
4) https://github.com/rachuriharish23/ibmsubscribe

Source Code

```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device

#Provide your IBM Watson Device Credentials
organization = "0shbez" #replace the ORG ID
deviceType = "IoTDevice"#replace the Device type wi
deviceId = "1234"#replace Device ID
authMethod = "token"
authToken = "123456789" #Replace the authtoken

def myCommandCallback(cmd): # function for Callback
    print("Command received: %s" % cmd.data)
    if cmd.data['command']=='motoron':
        print("MOTOR ON IS RECEIVED")

    elif cmd.data['command']=='motoroff':
        print("MOTOR OFF IS RECEIVED")

    if cmd.command == "setInterval":

        if 'interval' not in cmd.data:
            print("Error - command is missing required information: 'interval'")
        else:
            interval = cmd.data['interval']
    elif cmd.command == "print":
        if 'message' not in cmd.data:
            print("Error - command is missing required information: 'message'")
        else:
            output=cmd.data['message']
            print(output)

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:

        deviceCli.commandCallback = myCommandCallback

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