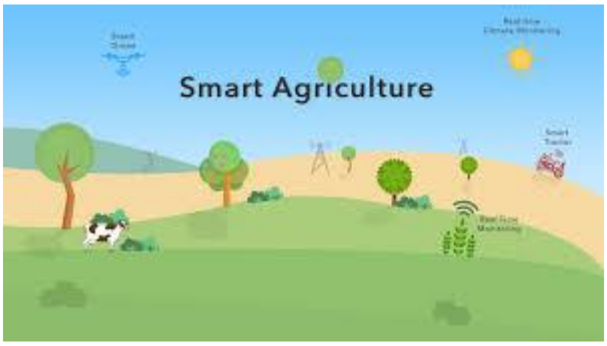
Smart Agriculture system based on IoT

A PROJECT REPORT

BY

SUNEJA MOSALI



INTRODUCTION

Overview:

The objectives of this report is to proposed IoT based Smart Agriculture System which will enable the farmers to get live updates of the temperature, humidity and soil temperature at a very low cost. The structure of this report covers IoT Technology and agriculture concepts, enabling technologies, IoT based smart agriculture system, the components and modules used in it and its working process, flow of the process in node-red and the final output of the overall process.

Purpose:

Internet of Things consists of two words- Internet and Things. The term things refers to various IoT devices having unique identities and have capabilities to perform remote sensing, actuating and live monitoring of certain sort of data. IoT devices also enable to have live exchange of data with other connected devices and application either directly or indirectly. The other term Internet is defined as Global Communication network connecting trillions of computers across the planets enabling sharing of information. We all know that agriculture is being done in every village since ages. Agriculture is nothing but of cultivating plants and is the main development needed other than any sector. Without agriculture we cannot survive. As the world is trending with new technologies now-a-days, Agriculture should also be collaborated with the new technologies to get real time benefits like cost savings, less man power, getting faster information about the weather forecasts , soil moisture, temperature, humidity, etc. All these can be fulfilled through sensors. But here, for this web application we are using Online IoT Simulator for getting temperature, humidity and soil moisture values. Using this app, a farmer can monitor his/her crop even if he/she is not physically present near the crops. As a result a farmer can yield profitable crops. This project is built using IBM cloud watson service, Node-RED application.

LITERATURE SURVEY

Existing Problem:

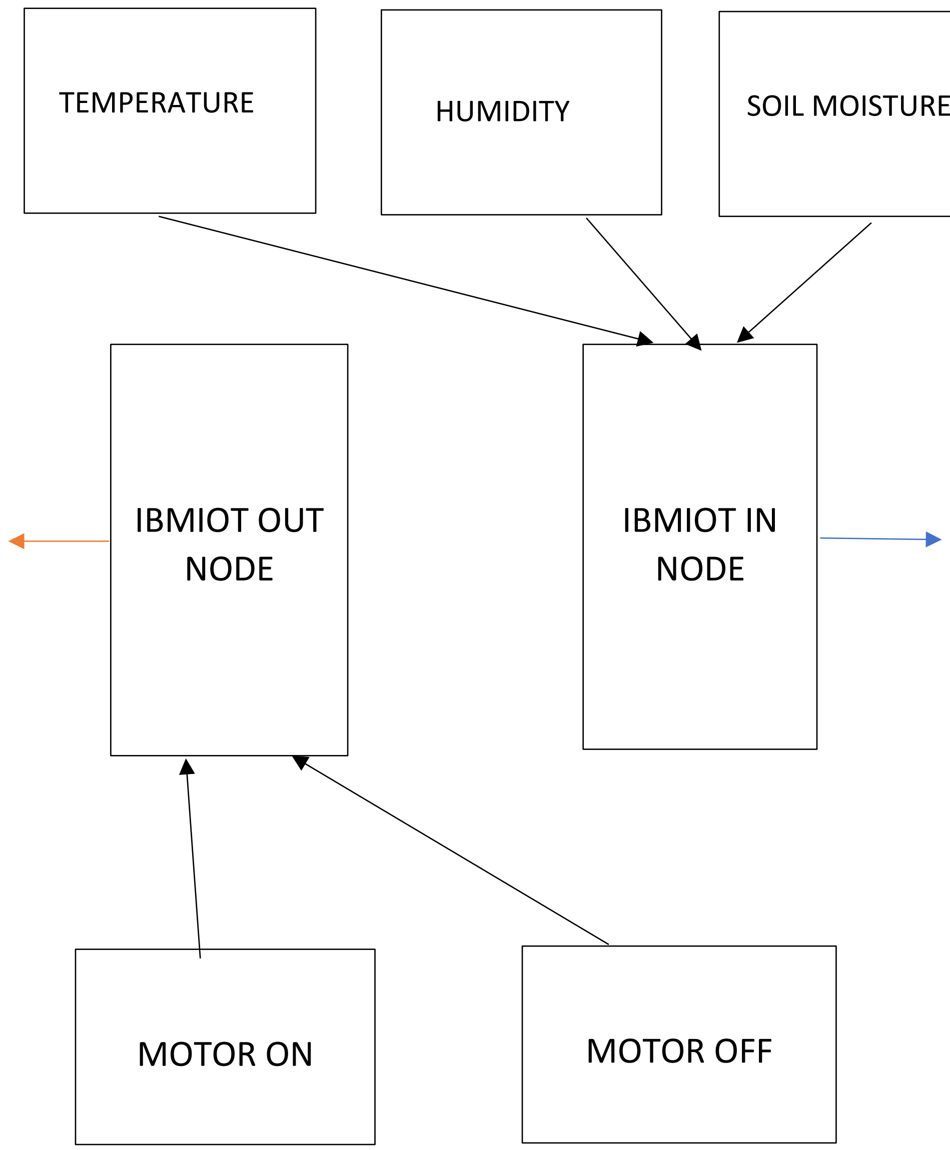
Agriculture is the foundation of our nation. In past days, agriculturists used to continuously figure out everything about their crops like soil fertility levels, which type of soil suits which crops, which season suits which crops, soil dampness, climate conditions, quantity and level of water required, etc. In long time, farmers used to water the crops manually, because there was no availability of the motors to pump water and ploughing also they used to with the help of Ox and bulls. Farmers used to do everything according to their estimations and presumptions. Anyway of negligence would cost them their entire crops. Agriculture in olden days is entirely filled with full of hard word and difficulty.

Proposed System:

To improve the efficiency of crops we are utilizing the innovation of new technologies. IoT advancement aids in social affair information on conditions like atmosphere, temperature, humidity and soil temperature around the weeds through sensors. IoT utilize farmers to get related with his crops from wherever and at whatever point. By using these technologies we can control motor operations and take care of our crops timely even if we are not physically present near them.

THOERITICAL ANALYSIS

Block Diagram:



Hardware Requirements:

1. Two devices, one for getting the data of the temperature, humidity and soil temperature, and the other device is to get the status of motor whether it is on or off.
2. IBM Watson IoT sensor to get the values of the conditions near our crops.
3. Motor to on and off.

Software Requirements:

1. IBM Cloud
2. Node-red
3. Python IDLE
4. IBM Watson

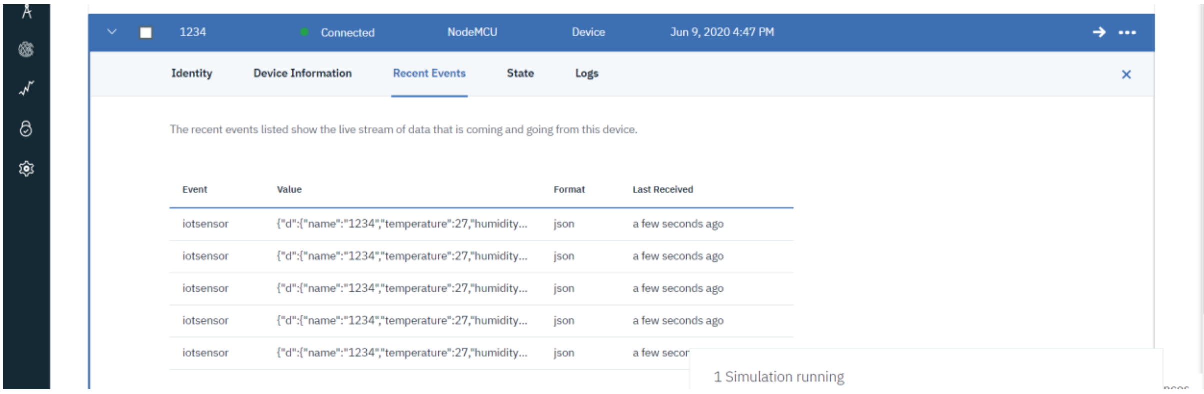
Functional Requirements:

1. Providing fast information
2. Security
3. Safety
4. Timely performance
5. Rapid reactions

EXPERIMENTAL INVESTIGATIONS

1. Sending data to the cloud:

When we are connecting the devices created with the Watson IoT sensor, we are sending the data to the IBM Cloud. They are stored in the json format. As in the UI we can only see the parameter values at that particular instance only, we are sending the data to the cloud and storing it. So it stores each and every second's data of the parameters in the cloud, which can be used as a reference to check at the later times.



2. Get the live updates of weather of our city:

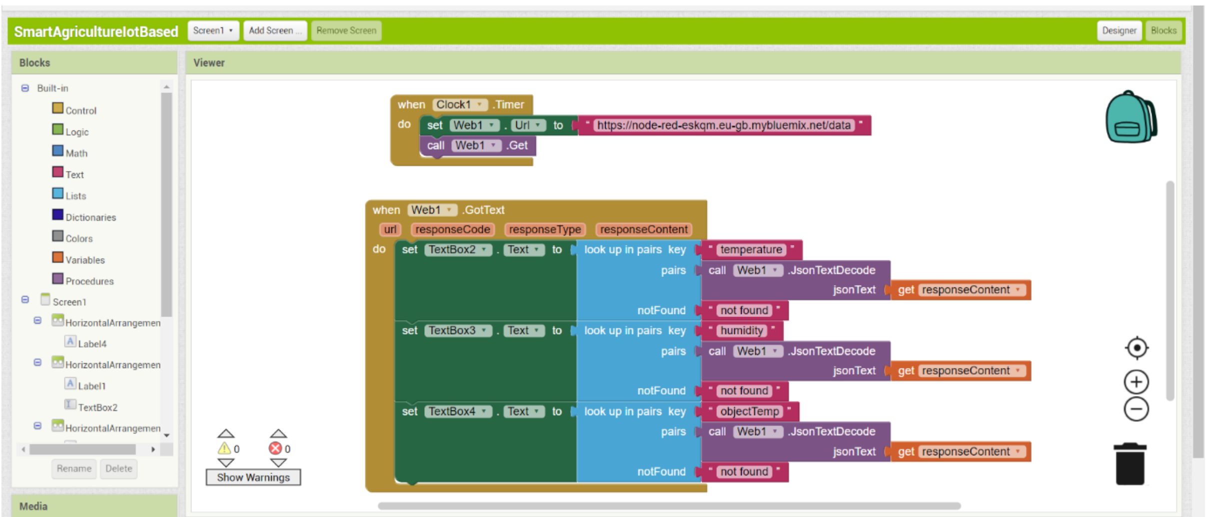
After creating a account in openweathermap.org we can generate a default API key for ourselves and find our city in the "find my city" search bar and giving our city name API key in the place mentioned in the API Call and pasting it in the URL bar, gives us the current temperature, pressure, humidity, wind speed and degrees, location, city, sunrise degrees and sunset degrees, etc in the json format. When we pass this information through a http request node in the node-red and connecting it to the gauge, displays us the values in the UI.

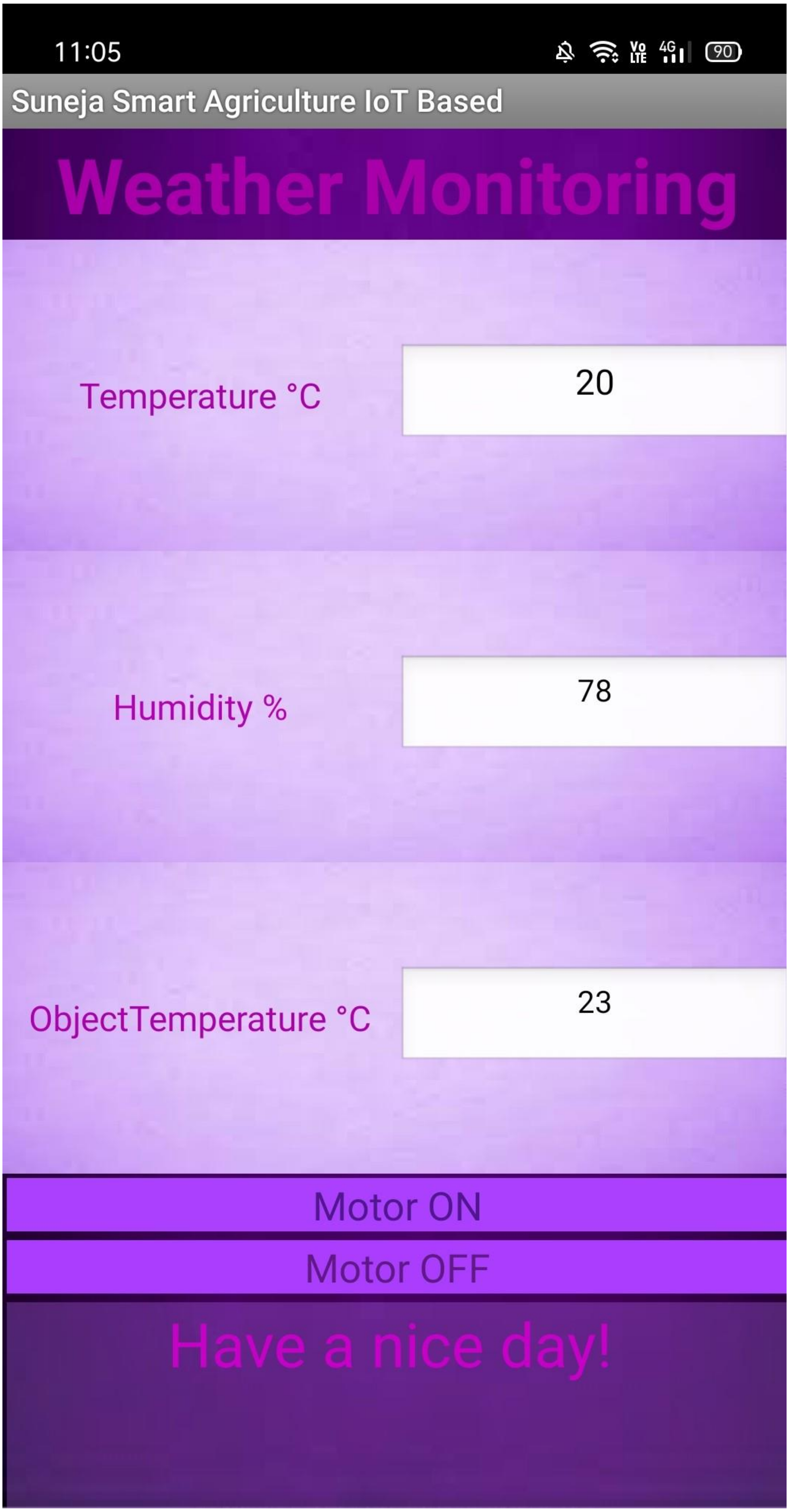
3. Creating a mobile application for the portable use for the farmers to check the weather conditions near their crops:

We will set the values of the temperature, humidity and soil moisture receiving from the sensor as global so that we can get the data anytime anywhere through the internet in fraction of seconds. We will be getting the values and setting them as global and pass the information to the http response node in the node-red.

For creating an app we must go to the MIT APP inventor and design our front and back end as per our requirements to make is easy for the users.

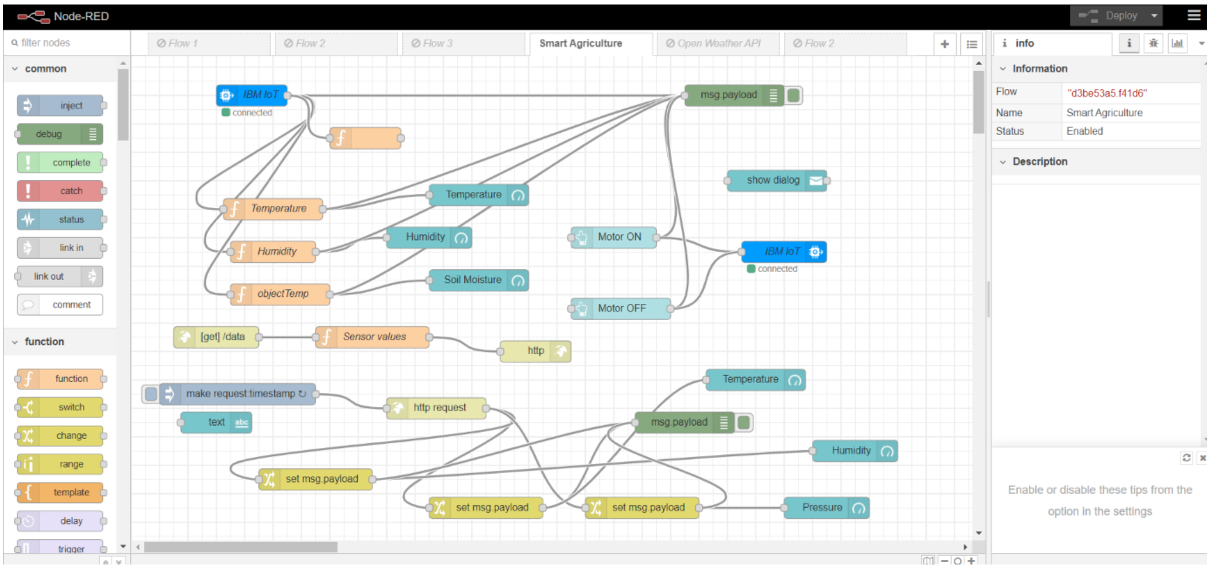
APP's backend design and front end design:




11:05
Suneja Smart Agriculture IoT Based
Weather Monitoring
Temperature °C
20
Humidity %
78
ObjectTemperature °C
23
Motor ON
Motor OFF
Have a nice day!

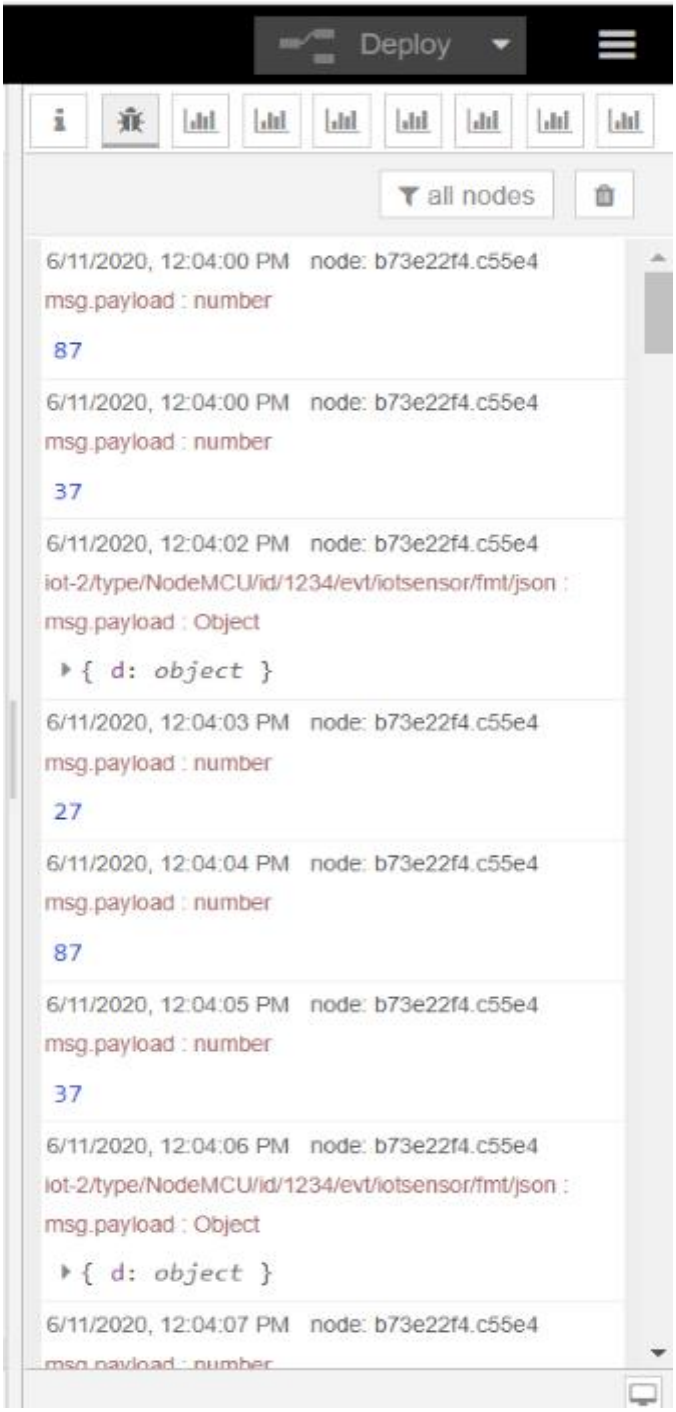
FLOW CHART

Flow chart of the nodes connected in Node-red:



RESULT

Getting values of the temperature, humidity and soil moisture in node-red:

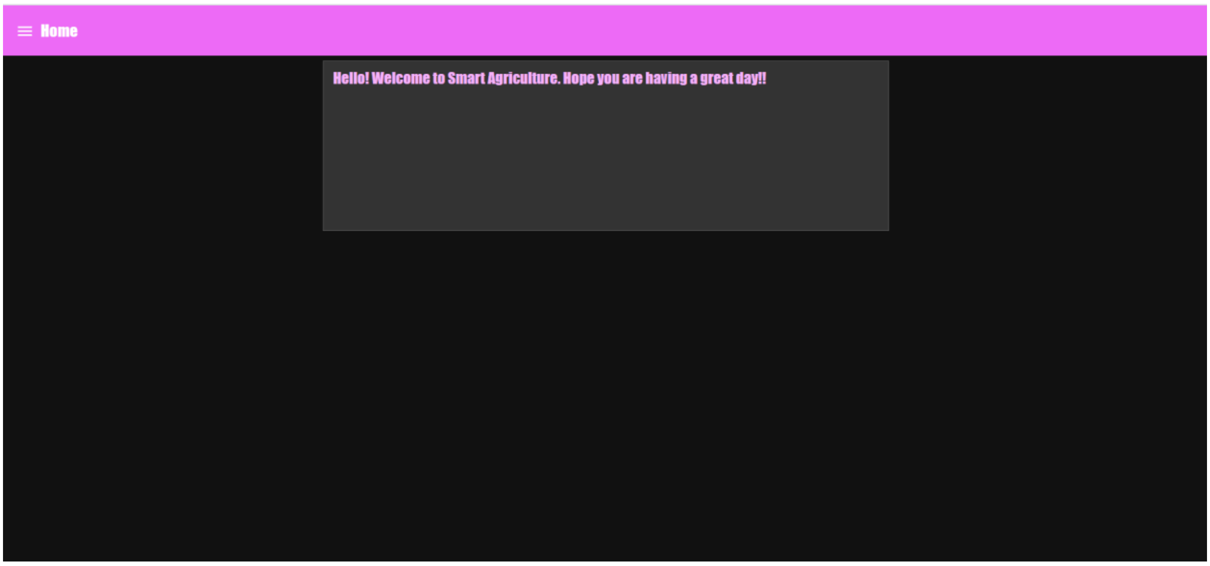


Getting status of the motor in node-red:

6/11/2020, 11:50:58 AM node: b73e22f4.c55e4
msg.payload : Object
{ command: "motoron" }

6/11/2020, 11:51:54 AM node: b73e22f4.c55e4
msg.payload : Object
{ command: "motoroff" }

Getting values in the created User Interface:



Getting OPEN Weather API Values of our city:

ADVANTAGES AND DISADVANTAGES

Advantages:

1. Allows the farmers to maximize the yields using minimum resources.
2. Remote monitoring.
3. Cost effective.
4. Easy data collection and helps in wireless monitoring and control.
5. Delivers high quality crop productions.
6. Time saving.
7. Less man power.

Disadvantages:

1. Smart Agriculture needs the availability of internet continuously.
2. May face internet connections problems.
3. Rural areas which are still developing may not get all these requirements.
4. Smart Agriculture techniques require farmers to get educated and learn all these new technologies in order to operate things. This is the most challenging task of adopting smart agriculture.
5. Any problem arises with the devices, can not operate or control the operations of the crop.
6. If the device gives any wrong data that may leads to wrong decision at times.

APPLICATIONS

The common applications of the Smart Agriculture are:

1. Sensor based systems for monitoring crops, soil, fields, livestock, storage facilities, or basically any important factor that influences the production.
2. Smart agriculture vehicles, drones, autonomous robots and actuators.
3. Connected agriculture spaces such as smart greenhouse or hydroponics.
4. Data analytics, visualization and management systems.
5. Crop water management.
6. Pest management and control works.
7. Precision agriculture.
8. Food production and safety, etc.

Conclusion

IoT based Smart Agriculture system for live monitoring of weather conditions has been proposed using IBM Cloud and IBM Watson. With this Smart Agriculture system we can predict the temperature, humidity, soil moisture and operate the motor. This system has high efficiency and accuracy in fetching the live data of temperature, soil moisture, and humidity. Irrigation system can be monitored time to time. Damage caused by the weather conditions can be reduced. We can conserve water by knowing the soil moisture if water not needed much for the crops and if temperature is high and the crops need more water frequently. Increase in the productivity of the crops. Harvesting at proper time.

FUTURE SCOPE

Future development will be focused more on increasing sensors on this system to fetch more live data regard to pest control, food production, etc also by integrating with the GPS to enhance the Agriculture IoT technology to full fill Agriculture Precision ready product. We can use it as a home automation controller. We can remotely operate or perform the jobs. Also combining with solar panels to conserve power. So the entire system is going to be eco-friendly.

BIOGRAPHY

Appendix:

Reference links:

1. https://www.ibm.com/cloud/get-started
2. https://nodered.org/docs/getting-started/windows#3-run-node-red
3. https://watson-iot-sensor-simulator.mybluemix.net/
4. https://www.ics.uci.edu/
5. https://openweathermap.org/
6. 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 = "Org ID" #replace the ORG ID
deviceType = "Device Type"#replace the Device type wi
deviceId = "Device ID"#replace Device ID
authMethod = "token"
authToken = "authtoken" #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()
```