SMART AGRICULTURE BASED ON INTERNET OF THINGS

C.Kailash, Student, B.Sc. IT Data Science

Rathinam College of Arts and Science, Coimbatore.

1. INTRODUCTION:

1.1 Overview:

In modern world, every appliances humans use are connected to internet. Controlling machines and other useful devices via internet facilitates less time consumption and ease in control over the appliances. This project focuses on automation of the agricultural works carried out in the field by live monitoring and access to control by various IOT devices (like raspberry pie) by internet.

1.2 Purpose:

The purpose of this project is to develop a user friendly Mobile application interface that enables the farmer to monitor the temperature, climatic conditions, water level etc of the cultivation field even from long distances. The application must also enable the farmer to control the irrigation facilities to the land. The mobile handset must have access to internet services wirelessly for the full pledged working of the model.

2. LITERATURE SURVEY:

2.1 Existing problem:

Human beings witness extreme climatic changes, deteriorating soil fertility, dry lands and collapsed ecosystem these days. The increasing population and decreasing cultivable lands and farmers make will surely make people’s future worse in case of healthy and sufficient food.
To encourage agriculture even when engaged in a full profession other than field work, and to facilitate the farmers an easy manipulation of the manual works that needs to be done crucially.

2.2 Proposed solution:

Internet of Things concept of automation and remote access needs to be applied in the scene. Providing the farmer a user friendly easy to use mobile application is the goal.

The android based Farming system is an automatic irrigation system which performs multiple operations in the field of agriculture; this project uses a centralized controller which is programmed to receive the input signal of multiple sensors of the field. Once the controller receives this signal, it generates an output that drives a relay for operating the water pump and other circuitry which provides automatic control action on field. If the user sees the moisture level of ever cannel has sufficient amount then user can switch off the motor easily using GUI.

The goal is to create a application that has control over the farm via internet. On theory basis this requires data like temperature, water level, Soil type, Humidity from the farm are required. Also to control the mechanisms an reliable central control unit from which data to user from the farm and commands from user to the farm will be maintained.

An ANDROID mobile operating system is interfaced with the microcontroller to control the parameters of the field. The soil moisture sensing arrangement is made by using two aluminum coated metallic rods inserted into the field at a distance. Connections from the metallic rods are interfaced to the control unit. This signal is sensed to mobile handset, which provides Graphical User Interface (GUI).

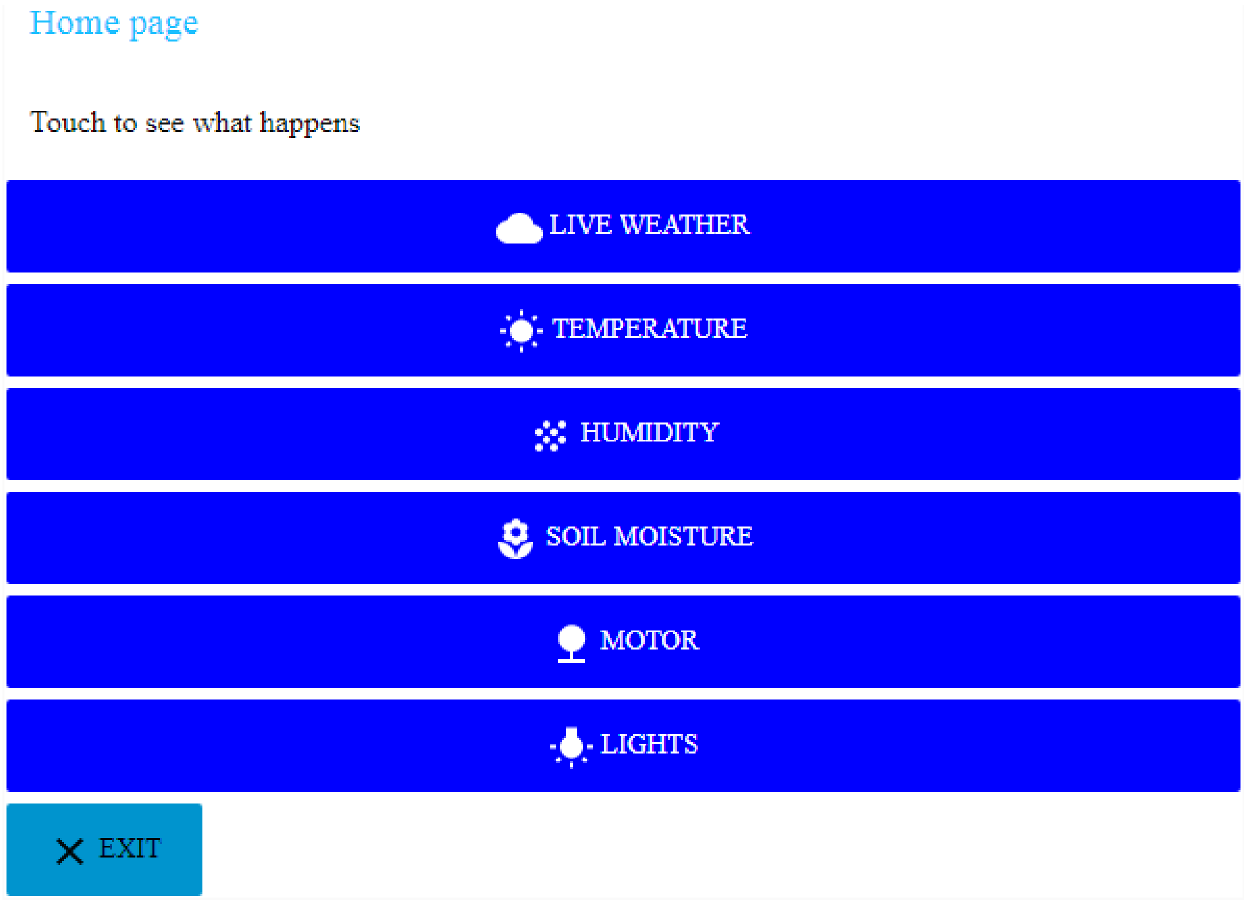


Fig 1.1(Snap shot from my UI stating all features as proposed in solution)

The application User Interface given above was created for the accomplishment of easy access to farm details and farm appliances such as lights and motor control to the user.

3. THEORITICAL ANALYSIS:

3.1 Block Diagram:

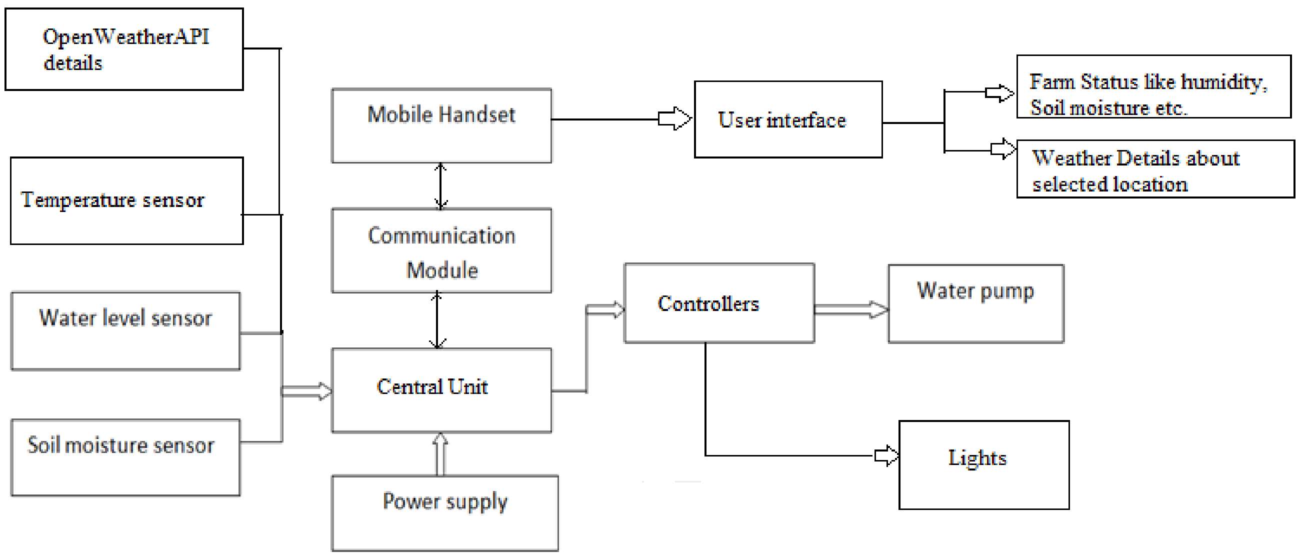


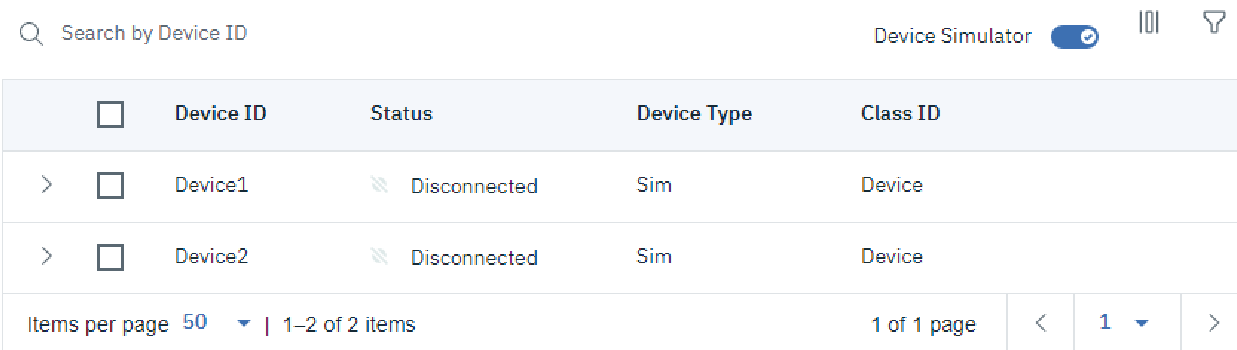
Fig 1.2(Block diagram)

3.2 Software designing:

For the project node.js software is used. Node-red an application of node.js is used in software designing part. For input data simulator from IBM IOT Watson is utilized. IOT Devices are created from IOT platform of IBM. **Device1** and **Device2** of the type "Sim" are the two devices experimented for this project. Node red is a programming tool for wiring hardware devices, online APIs. It facilitates browser based editor which makes connecting the flows much easier.

Nodes of node-red such as IBM IOT in/out, Open weather, plays crucial roles in this project as they are directly concerned with the input and output of the entire project. JSON language is used to send messages as payload to the devices in and out in this project.

IBM Watson IOT Platform is a foundational cloud offering that can connect and control IOT sensors, appliances, homes, and industries. Built on IBM Cloud, Watson IOT Platform provides an extensive set of built-in and add-on tools. Use these tools to process IOT data with real-time and historical analytics, extract key performance indicators (KPIs) from your data, add "smarts" in the cloud for non-smart products, and securely connect your own apps and existing tools to the Watson IOT Platform infrastructure.



Search by Device ID | Device Simulator

	Device ID	Status	Device Type	Class ID
	Device1	Disconnected	Sim	Device
	Device2	Disconnected	Sim	Device

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

Fig 1.3(Devices made in IBM IoT platform)

These two devices are connected to IBM in and IBM out node of the node-red editor respectively. Using generated api from the platform this measure to connect the devices were done.

The IBM in node was given event type and the IBM out node was given Command type which makes them ideal in the full pledged working of this application.

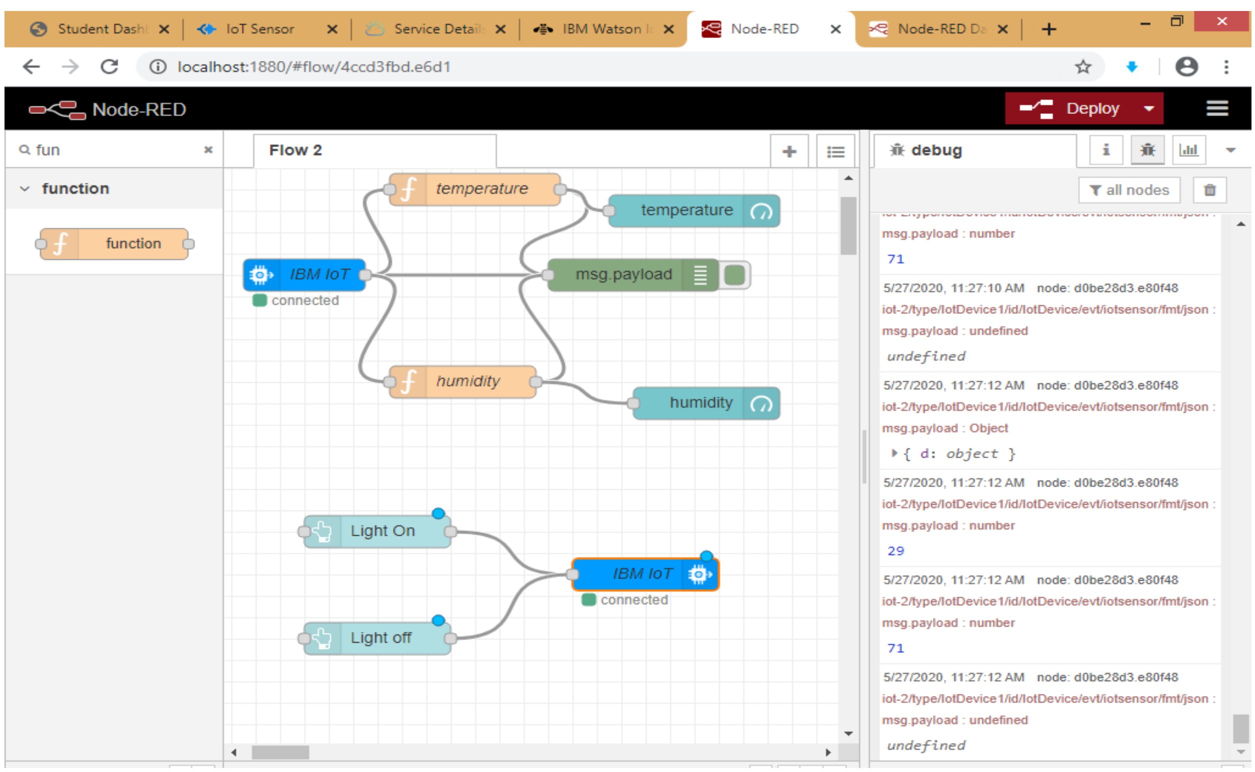


Fig 1.4(Snapshot of node-red platform- IBM devices)

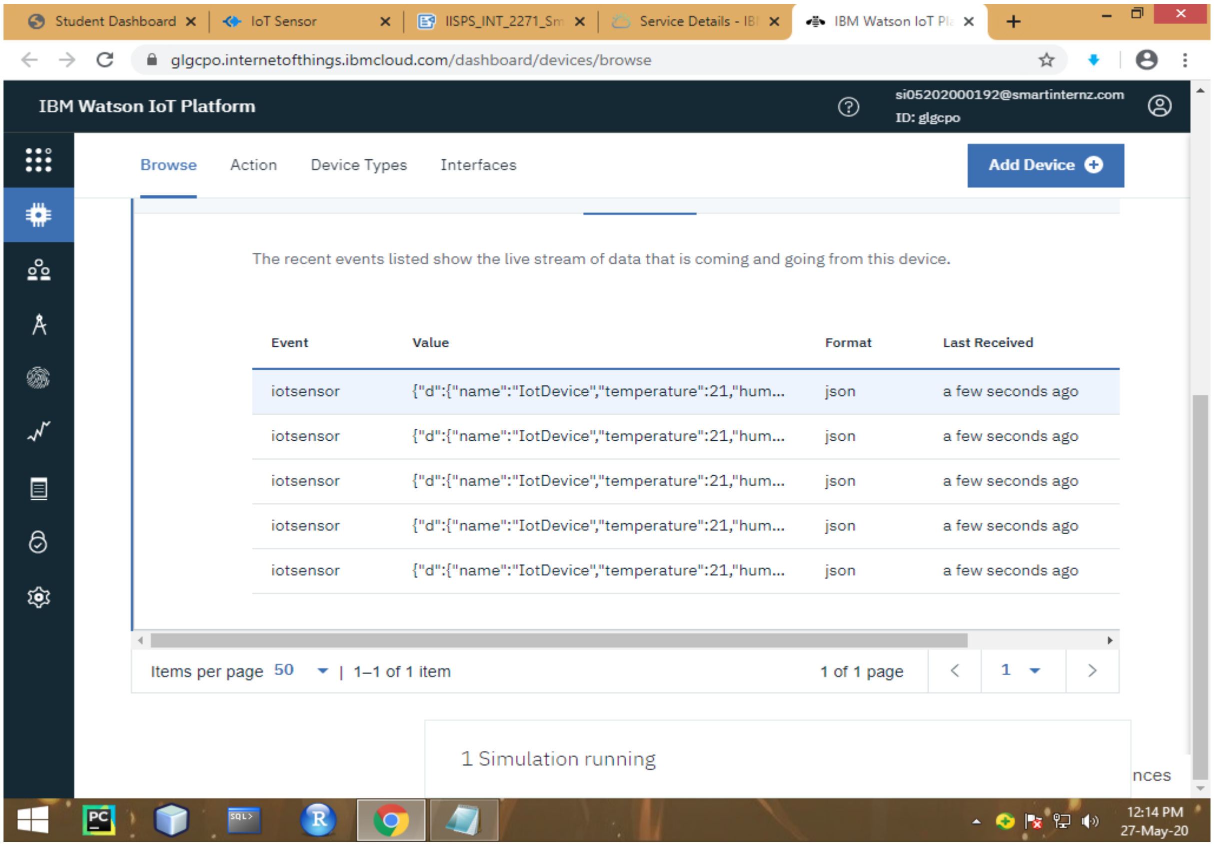


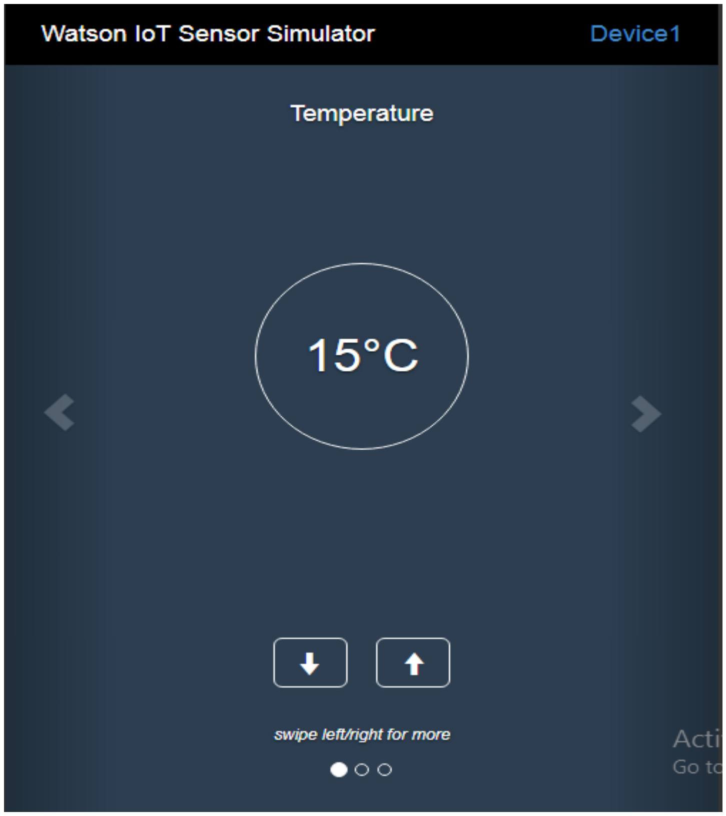
Fig 1.5(IBM IOT Platform-Messages from simulator)

4. EXPERIMENTAL INVESTIGATIONS:

The experiments were carried out in node-red dashboard using IBM Watson simulator which provided the parameters temperature, humidity and object temperature (soil moisture). The motor and light controls are set to turn on and off by the use of buttons. When triggered a command, the python code which was set to catch information from the device's output responded correctly. This shows that the command from the online dashboard controls the motors and lights on time.

The connection to IOT simulator from blue mix requires a Organization ID, which can be found at the top right corner of Watson IOT platform. Further the ID of the device which is going to receive the data from simulator and access token are necessary to connect to the simulator.

As soon as the device is connected the name appears at top right corner of the online simulator. Also in the IOT platform, the device status shows connected. After a few seconds' data as payload messages in JSON format will be sent to the device. These messages can be viewed at the recent events column.



(Fig1.6 Picture of online simulator)

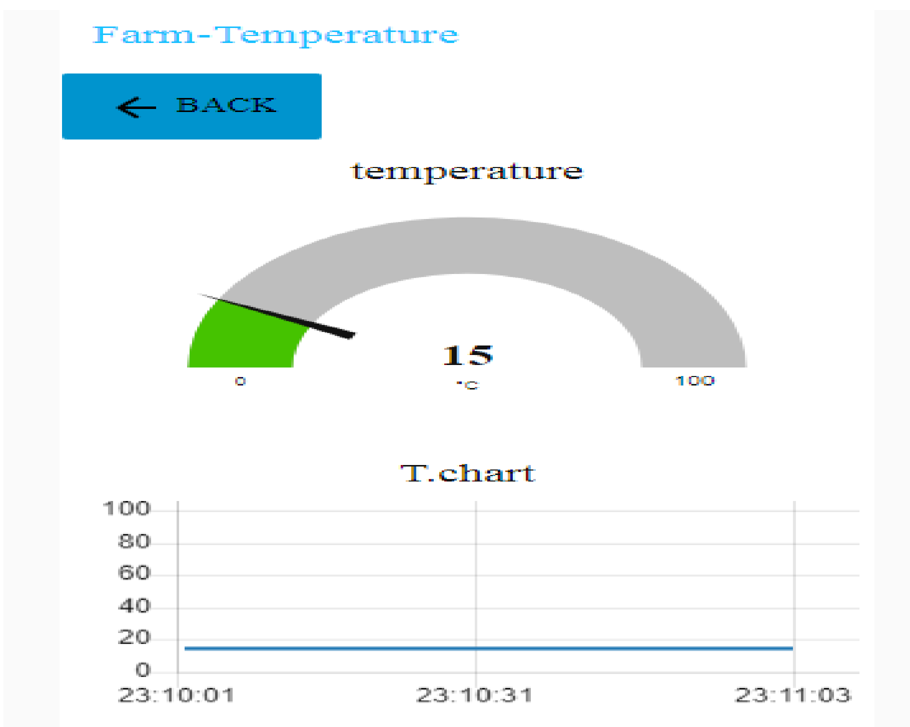


Fig 1.7 (Temperature details in UI)

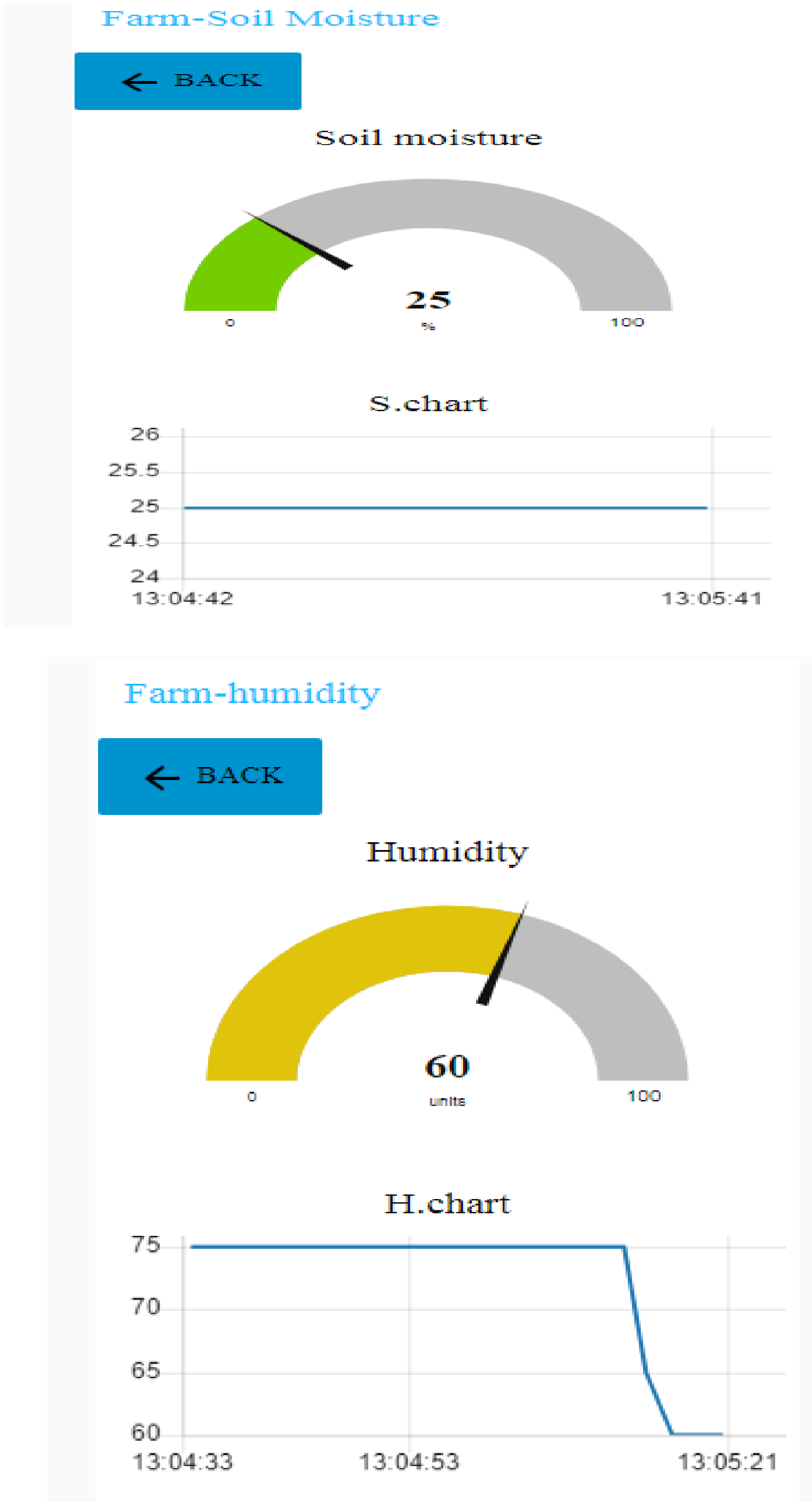


Fig 1.8, 1.9(Humidity & Soil Moisture details at UI)

The controls are given as buttons which sends command messages to the device central unit and access the irrigation motors, light facilities.

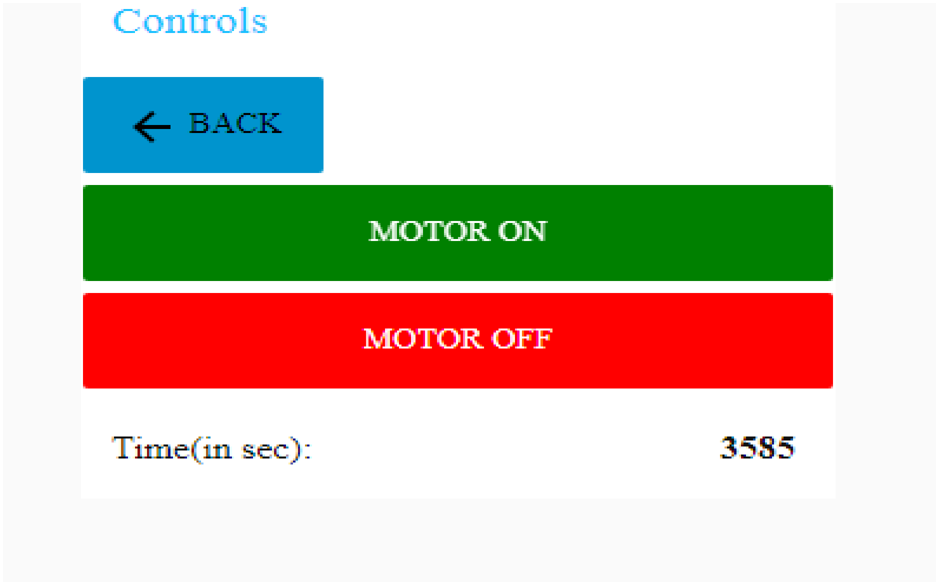


Fig 1.10 (Motor controls)

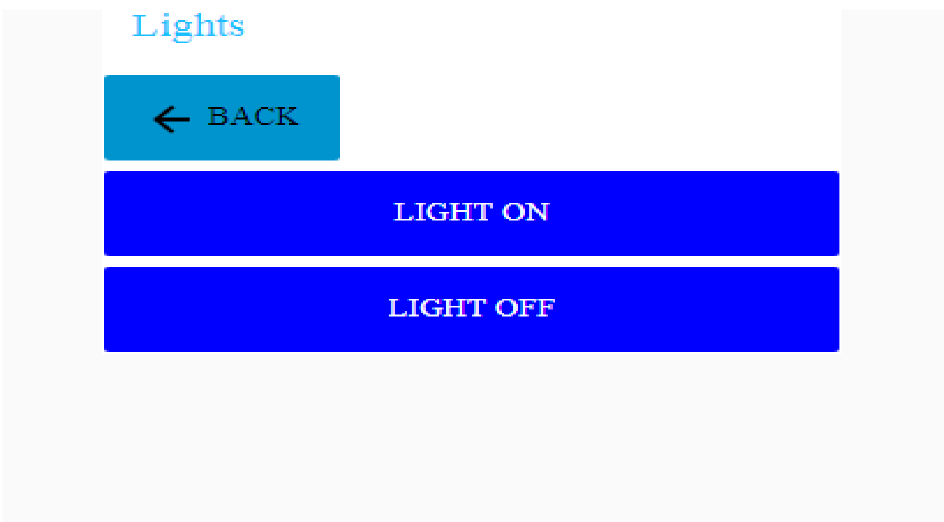


Fig 1.11(Light controls)

The data from the **ibm iot out** node from Node-red which is connected to the second device **Device2**, receives the commands from the User Interface Dashboard such as light on, off Motor on, off. At the python editor, coding was made in such a way it receives data from the second device by giving the device credentials. When run it showed device connected. And further changes were noted as the commands like light on, off; motor on, off were given.

The python terminal received desirable output “LIGHT ON IS RECEIVED”, “MOTOR ON RECEIVED”.

```
# connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greetin
deviceCli.connect()

while True:

        deviceCli.commandCallback = myCommandCallback

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

```
2020-06-13 12:58:00,553   ibmiotf.device.Client     INFO    Connected successfully: d:glgcpo:Sim:Dev
ice2
```

```
Command received: {u'command': u'lighton'}
LIGHT ON IS RECEIVED
Command received: {u'command': u'lightoff'}
LIGHT OFF IS RECEIVED
Command received: {u'command': u'motoron'}
MOTOR ON IS RECEIVED
Command received: {u'command': u'motoroff'}
MOTOR OFF IS RECEIVED
```


Fig 1.12(commands received at python terminal)

5. FLOWCHART:

Flow chart is explaining that, the data from Open weather API and sensors fitted in the farm are sent to the device of the user. In the device’s user interface details like weather details of the location, readings from the temperature, humidity, soil moisture as gauges, An control unit that has control over electrical appliances such as light bulbs and water irrigation motors.

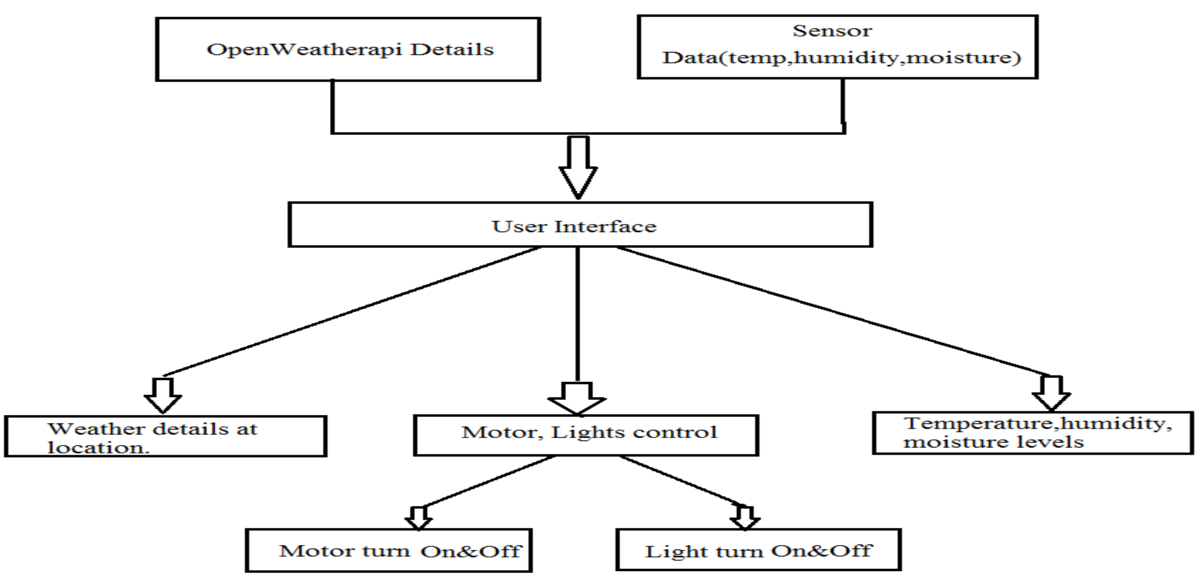


Fig 1.13(Flow chart)

6. RESULT:

The response from the dashboard to the simulator matched and changed the measuring gauges perfectly. Using the http request node and open weather node, weather details of the location from open weather api website were derived and displayed successfully in the dashboard.

Thus the UI perfectly shows variation from the simulator and also commands the control devices like motors and lights which facilitate the ease of access to irrigation and lighting to the field. Thus the cause of this project is met.

7. ADVANTAGES&DISADVANTAGES:

Merits:

1. The system increases the crop productivity and reduces farmer's workload.

2. The time consumed is less there by giving more throughputs.

3. Efficient water usage

4. Leads to development of a cost effective irrigation control system. Save electrical energy.

5. This automation system will be used for disabled and people at long distance and farms.

6. Farmer can set the time, when motor turns off automatically.

Demerits:

1. This architecture is based on the capabilities of current and next-generation controllers and their application requirements, which may lower favorable output.

2. Farmer has to keep a frequent watch over the application to check for how much time the motor has run. Though a warning notification is set for certain minutes after motor has turned on, monitoring needs to be active from the farmer's side.

8. APPLICATIONS:

1. Automation can be used in the fields of medical, home care, pet care etc.

2. Turning lights on and off by mobile is handy for old people.

3. In agriculture, building canals which directly lets the water to plant's surface can reduce water wastage. When irrigation controls is given via mobile, farmer's workload is reduced.

4. The same way as turning lights on and off, food containers and water containers placed above bowls of pets, can be controlled. This becomes helpful to the owner to feed the pet when he/she is away from home.

5. In Medical field a urgent message from the patient's heart rate, blood pressure can sent to the concerned doctor's mobile phone which makes the rate of sudden demise much less.

9. CONCLUSION:

The main concern of this project was to facilitate a farmer with modern methods of controlling the farm land with ease by the use of mobile application. The purpose was met satisfiable manner and, in future by further development of technologies a much easier task handling mobile application, can be created.

10. FUTURE SCOPE:

We can further develop interface screen in order to display the current status of the soil moisture content levels, percentage of water utilized to water the plant, duration of time for which the water pump is ON, etc. We can also show the graphical representation of the moisture content levels in the soil. **IOT** sensors are capable of providing farmers with information about crop yields, rainfall, pest infestation, and soil nutrition are invaluable to production and offer precise data which can be used to improve **farming** techniques over time. With a future of efficient, data-driven, highly-precise farming methods, it is definitely safe to call this type of farming smart. We can expect IOT will forever change the way we grow food.

11. Bibliography:

[1] www.digiteum.com/iot-agriculture.

[2] Smart Irrigation System Using GSM Module and Controller - R.Ashok, G.Jeyameena, T.Shobana, K.Lakshmi Priya.

[3] Smart Agriculture System using IoT Technology- Muthunoori Naresh, P Munaswamy.

[4] Green House Automation using Zigbee and Smart Phone YR Dhumal, JS Chitode - International Journal of Advanced Research in …, 2013.

APPENDIX

A. Source code:

https://github.com/rachuriharish23/ibmsubscribe- python source code from Harish Rachuri.