Smart Agriculture System

Abstract

In every country agriculture is done from ages which are considered to be science and also art of cultivating plants. In day today life, technology is updating and it is also necessary to trend up agriculture too. IoT plays a key role in smart agriculture. Internets of Things (IoT) sensors are used to provide necessary information about agriculture fields. The main advantage of IoT is to monitor the agriculture by using the wireless sensor networks and collect the data from different sensors which are deployed at various no des and send by wireless protocol. By using IoT system the smart agriculture includes the humidity sensor, temperature sensor, moisture sensor and DC motor. This system starts to check the humidity and moisture level. The sensors are used to sense the level of water and if the level is below the range then the system automatically stars watering. According to the change in temperature level the sensor does its job. IoT also shows the information of humidity, moisture level by including date and time.

1.Introduction

One of the largest livelihood providers in India is Agriculture. Agriculture plays an essential role in supporting human life. The rise in population is proportional to the increase in agriculture production. Basically, Agriculture production depends upon the seasonal situations which do not have enough water sources. To get beneficial results in agriculture and to overcome the problems, IoT based smart agriculture system is employed.

Global and regional scale agricultural monitoring systems aim to provide up-to-date information regarding food production. In IoT-based smart farming, a system is built for monitoring the crop field with the help of sensors like light, humidity, temperature, soil moisture, etc. The farmers can monitor the field conditions from anywhere. IoT-based smart farming is highly efficient when compared with the conventional approach. It will not only automatically irrigate the water based on the moisture level in the soil but also send the Data to Server to keep track of the land condition.

1.1 Overview

This is a Smart Agriculture System project based on Internet Of Things (IoT), that can measure soil moisture and temperature conditions for agriculture using Watson IoT services. IoT is network that connects physical objects or things embedded with electronics, software and sensors through network connectivity that collects and transfers data using cloud for communication. Data is transferred through internet without human to human or human to computer interaction.

In this project we have not used any hardware. Instead of real soil and temperature conditions, sensors IBM IoT Simulator is used which can transmit soil moisture temperature as required.

○ **Project requirements:** Node-RED, IBM Cloud, IBM Watson IoT, Node.js, IBM Device, IBM IoT Simulator, Python 3.8, Open Weather API platform.

○ **Project Deliverables:** Application for IoT based Smart Agriculture System.

1.2 Purpose

IoT based farming improves the entire agriculture system by monitoring the field in real-time. With the help of IoT in agriculture not only saves the time but also reduces the extravagant use of resources such as water and electricity. Sometimes due to over or less supply of water in the agricultural field crops may not grow proper. Using IoT supply of water and growth of plants can be satisfied to a greater extent. The flow of water can be controlled from the application.

Literature Survey

2.1 Existing problem:

In India irrigation systems are inadequate, leading to crop failures in some parts of the country because of lack of water. In other areas regional floods, poor seed quality and inefficient farming practices also leads to crop failures. By providing proper technology to farmers we can increase the crop yield which further increases the farmer's income and provides more contribution to country's GDP. The challenge is to develop a cheap, but accurate system that will provide the farmer with the adequate amount of information that is needed to make an informed decision about whether or not the crops should be watered or not.

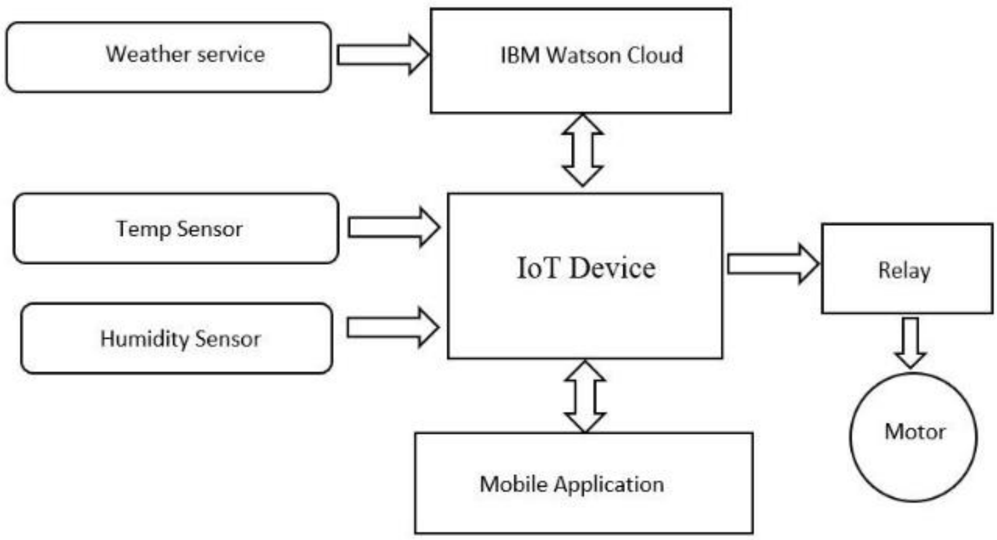
2.1 Proposed Solution:

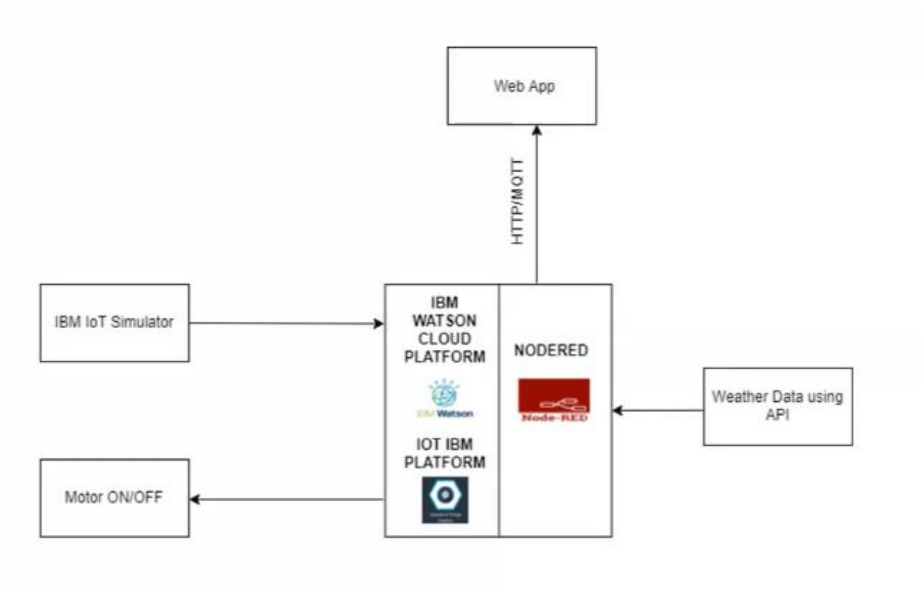
By adopting the latest technologies of IoT in agriculture practices, every aspect of traditional farming method can be changed from roots. Currently, integration of sensors and the IoT in smart agriculture can raise agriculture to levels which were previously unimaginable. It has also been proposed that by constantly measuring soil parameters like soil type, nutrient presence, flow of irrigation, pest resistance, etc. we can prepare a report, based on that report a farmer may crop seeds which are suitable to that land. Furthermore, the use of unmanned aerial vehicles for crop surveillance are favourable for optimizing crop yield. In our proposed system the IoT device derives the values from humidity and temperature sensors buried in the ground. The device reports these values to the user. With the help of mobile application or web-app user can see these values in the dashboard. The IoT device is also integrated with weather forecast services so that user, farmer in our case always updated with the weather in that location. By using state-of-art sensors we can precisely measure soil parameters and report these back to farmer, with his experience he may plant a crop which is suitable to that land or we can use artificial intelligence to suggest the farmer for a suitable crop. We can also control flow of water into the crops by using IoT device. With help of relays we can control the motor from mobile application with a single click from anywhere. The data collected can be processed using Machine Learning algorithms and the farmer can be provided with AI inputs so that the decision making process becomes more efficient.

3. Theoretical Analysis

3.1 Block Diagram:

The IBM Watson Cloud receives the temperature and humidity data from sensors and reports these values to the user through web-based apps. With the help of weather service API like OpenWeather's API, the IBM Watson gets the weather forecasting data from weather service providers and updates the user with weather of the farm location. Based on these data farmer can either turn on or turn off his motor via the web app itself over the internet.





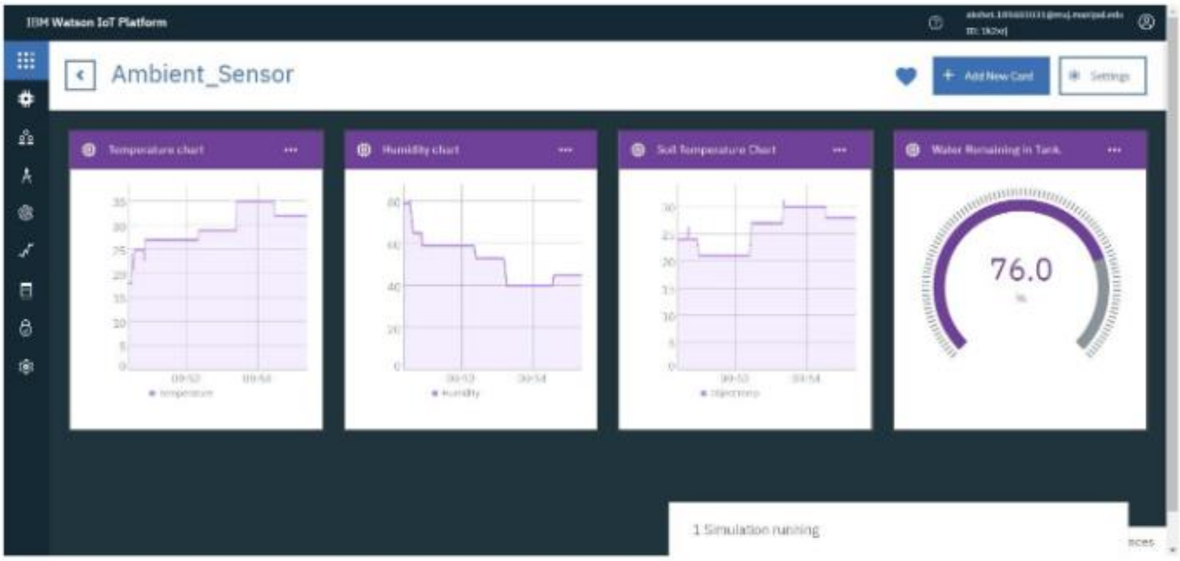
3.2 Hardware/Software Designing:

The hardware part consists of very basic components. The temperature and humidity sensors viz. a DHT11, are connected to the analog input pins of IoT device like NodeMcu. In case of RaspberryPi we need to use a Digital to Analog converter to get values from analog sensor. The digital out pin of IoT device is connected to water motor through relay. In software part we need to configure our IoT device with IBM Watson cloud service so that we can visualize the sensor data to the user.

4. Experimental Investigations:

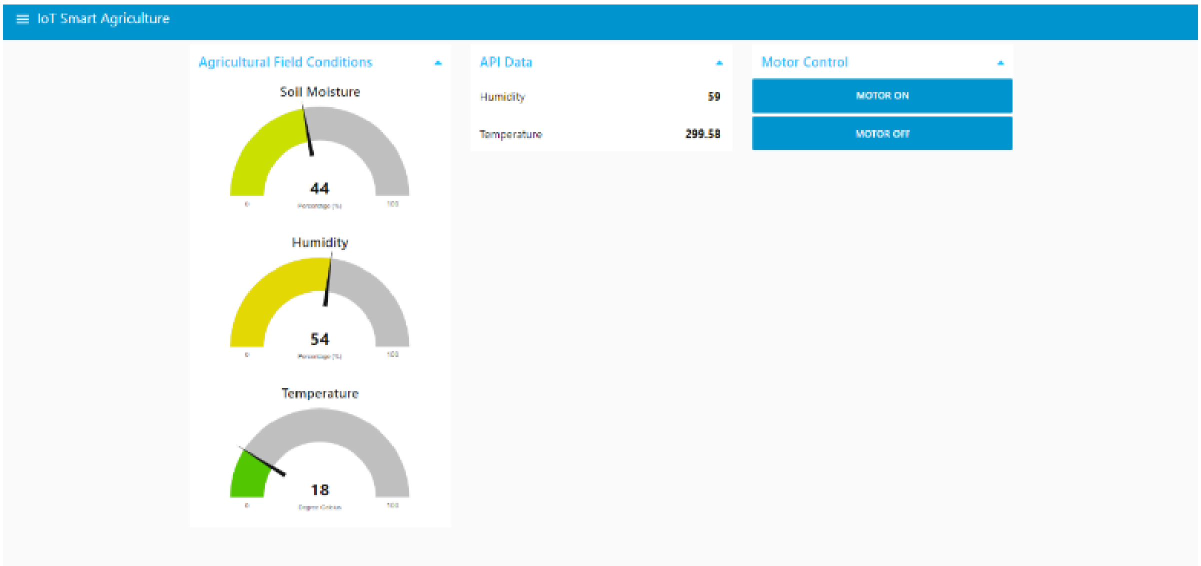
By using IBM Bluemix, we can simulate working of temperature and humidity sensors. Once sensors are simulated, we can connect those sensors to the device that we have
created in IBM Watson platform. We can also add different data sets including the height of
the water level in a water tank or the amount of fodder remaining by running the Simulations
in IBM Cloud Platform. Sensor simulator is shown below:

IBM Sensor.

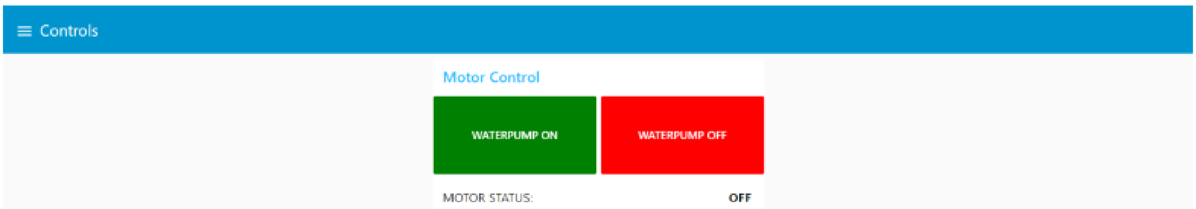


Using the Simulator.

By using Node-Red palette we can visualize those sensors values in a dashboard. We can also configure a weather service provider by using http request node. By connecting all the nodes in node-red and deploying the flow we will get reading as shown below. We control the motor by sending digital signals to relay from dashboard by using IBM out and button nodes in node-red once all these flows are deployed the system starts showing weather forecast and sensor data.

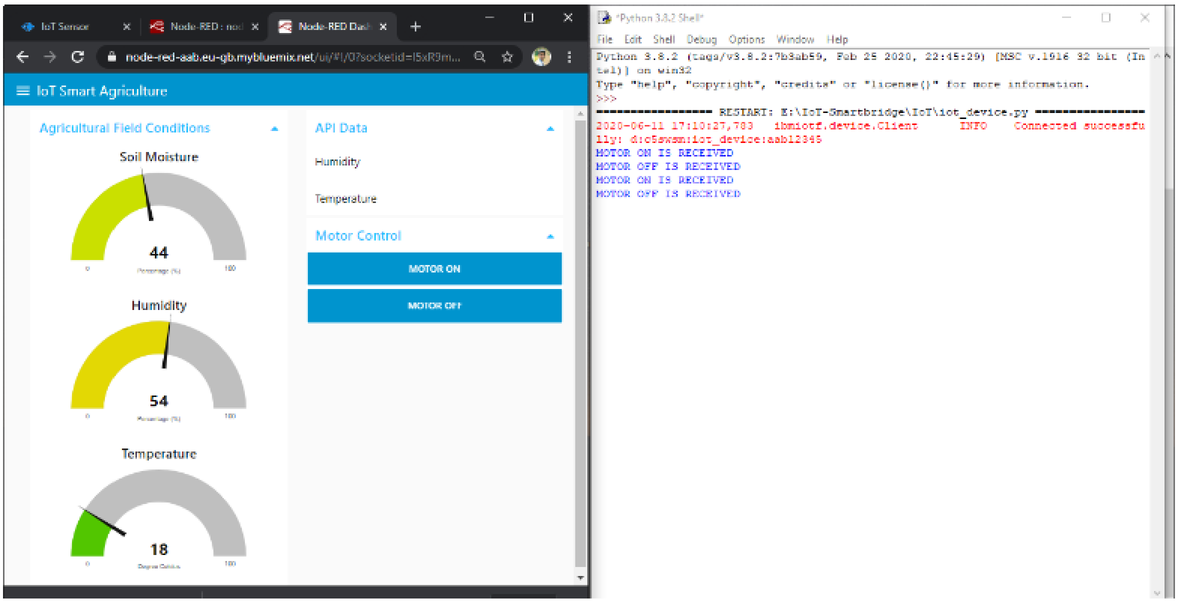


fig(b1) Web Application



Controls.

fig(b2) Web Application - Tab 2



fig(c) Device Control Action

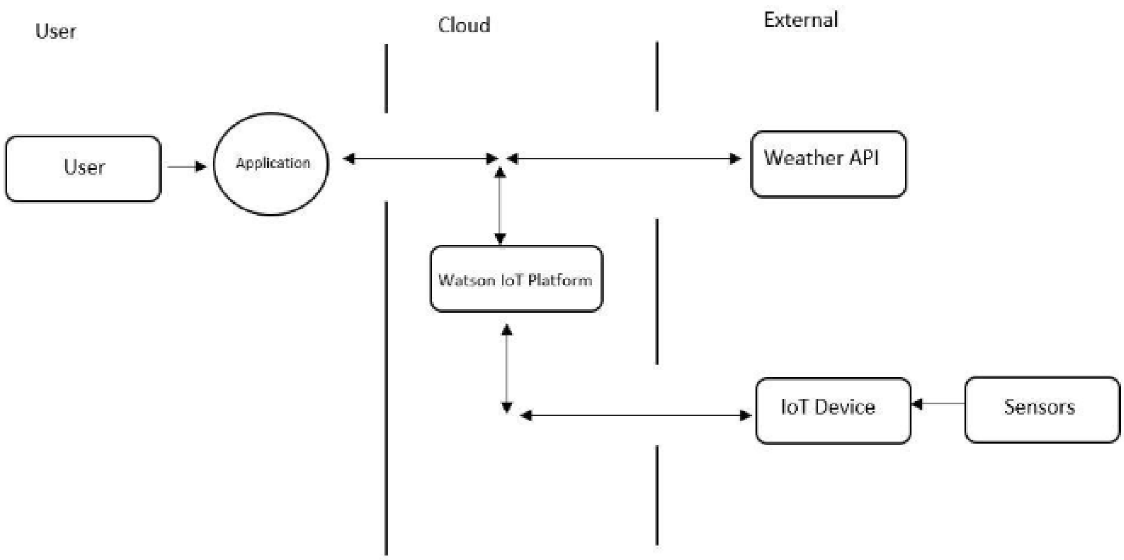
Simulator passes the data through IBM Colud to the web application. The data is displayed on the ashboard show in fig(b1 & b2). Web Application is build using Node-RED. We have created 2 tabs:

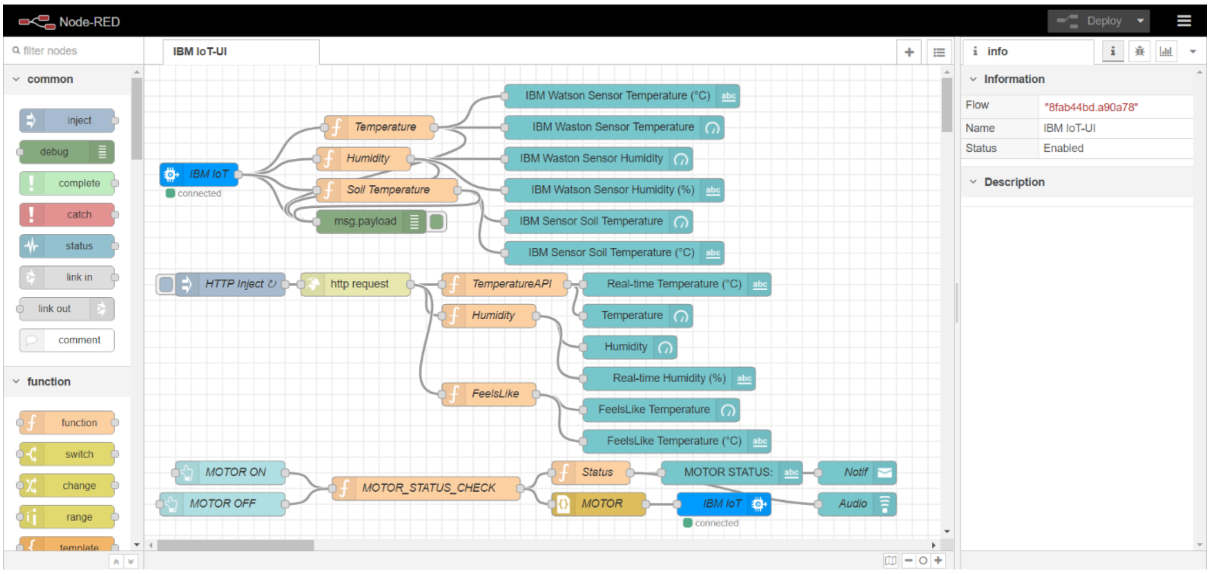
1. IoT Smart Agriculture.
2. Graphical Representation.

Web Application is also used to control the devices further like motor, pumps, lights, or any other devices in the agricultural field. In this project the output is passed using python code and the control action is displayed in python code console window in fig(c).

5. Flow Chart:

When the user opens the application, weather forecast from the weather service provider is shown in dash board. User can also see the sensor data in gauges with the help of IBM nodes in node-red. The Watson platform derives the sensors data from IoT device and visualizes that data to the user. The Watson platform continuously updates the weather of the location. Based on these data user can turn on or turn off his motor by pressing respective buttons in the application.





Flowchart.

Following are the nodes used in the project in the Web Application:

1. IBM IoT: IN and OUT Nodes.
2. function Nodes.
3. Gauge Nodes.
4. Chart Nodes.
5. Debug Node.
6. Button Nodes.

Following are the nodes used for the weather condition from openweathermap:

1. Timestamp Node.
2. http request Node
3. Function Nodes.
4. Text Nodes.
5. Debug Node.

6. Result:

By using the above technology, we can increase the crop yield and also helps the farmers to understand their land. Farmers can monitor their lands from anywhere with their smartphones. From all the sensor data and weather conditions he can come to a conclusion about watering the crop, this helps in saving crops from excessive or inadequate water.

We have successfully build a web based UI and integrated all the services using Node-RED.
Web Application : https://node-red-aab.eu-gb.mybluemix.net/ui/

7. Advantages and Disadvantages:

ADVANTAGES:

1. Enables the farmer with real-time necessary data of the farm conditions.
2. Real-time weather forecasting of crop location.
3. Farmers can control their motor from anywhere in the world at any given time.

DISADVANTAGES:

1. Network error can cause device to disconnect.
2. Weather forecast may not be accurate at all times.
3. There is no confirmation that the motors are really turned on and off physically.

8. Applications:

This system with some modifications can be used in different scenarios like healthcare industry and also production lines. It is cheap, easy to use, requires no technical knowledge and enable the user to control machines wirelessly from anywhere in the world.

- Precision Farming that is farming processes can be made more controlled and accurate.
- Live monitoring can be done of all the processes and the conditions on the agricultural
field.
- All the controls can be made just on the click.
- Quality can be maintained.

9. Conclusion

IoT will help to enhance smart farming. Using IoT the system can predict the soil moisture level and humidity so that the irrigation system can be monitored and controlled. IoT works in different domains of farming to improve time efficiency, water management, crop monitoring, soil management and control of insecticides and pesticides. This system also minimizes human efforts, simplifies techniques of farming and helps to gain smart farming. Besides the advantages provided by this system, smart farming can also help to grow the market for farmer with single touch and minimum effort.

10. Future Scope

The project has vast scope in developing the system and making it more user friendly and the additional features of the system like:

- By installing a webcam in the system, photos of the crops can be captured and the data can be sent to database.
- Speech based option can be implemented in the system for the people who are less literate.
- GPS (Global Positioning System) can be integrated to provide specific location of the farmer and more accurate weather reports of agriculture field and garden.
- Regional language feature can be implemented to make it easy for the farmers who are aware of only their regional language.

References:

[1]k.lakshmisudha, swathi hegde, neha cole, shruti iyer, "good information used for very widespread distribution", weekly status from microcomputer applications (0975-8887), number 146-no.11, july 2011

[2] nikesh gondchawar, Dr. r.complexion.kawitkar, "iot based agriculture", an almanac that includes all the analytics including the processing of the current smart minicomputer in addition to chat planning (ijarcce), vol.5, affair 6, June

2016. A comprehensive article on the latest Techniques and Technology in Computing and Communication ISSN: 2321- 8169 Volume: 5: 2 177 - 181 Credit
[3] M.K.Gayatri, J.Jayasakthi, Dr.G.S.Anandhamala, "Providing Smart Agriculture Farm Solutions to Better Farmers Using IoT", INEEE International Convention on Technological Innovations in ICT for Agriculture
[4] Fitness. r. nandurkar, slant. r. thool, r. vegetation. thool, global consultation on telemechanics, control, resilience and (aces), 2014. Development (TIAR 2015).
[5] Paparao Nalajala, D. Hemanth Kumar, P. Ramesh and Bhavana Godavarthi, 2017. Design and Functionality of Real-Time Farming Monitoring System Using Agriculture by the Internet of Things (IoT). Journal of Engineering and Applied Science, 12: 9389- 9393.
[6] Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra Nieto- Garibay, and Miguel Ángel PortaGándara, "Computerized Irrigation System Using Wireless Sensor Network and GPRS Module", IEEE Transaction on Instrumentation and Measurements, 0018-9456,2013d Rural Development

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
deviceId = "Device ID" # replace Device ID
authMethod = "token"
authToken = "authtoken" # Replace the authtoken
def myCommandCallback(cmd): # function for Callback
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()

deviceCli.connect()
while True:
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

B. Node-RED Flow:


```
[
    {
        "id": "8fab44bd.a90a78",
        "type": "tab",
        "label": "IBM IoT-UI",
        "disabled": false,
        "info": ""
    },
    {
        "id": "ae13bd5c.c66c8",
        "type": "ibmiot in",
        "z": "8fab44bd.a90a78",
        "authentication": "apiKey",
        "apiKey": "7dd0c222.d9446c",
        "inputType": "evt",
        "logicalInterface": "",
        "ruleId": "",
        "deviceId": "1410a2000",
        "applicationId": "",
        "deviceType": "IoT_Sensor",
        "eventType": "+",
        "commandType": "Data",
        "format": "json",
        "name": "IBM IoT",
        "service": "registered",
        "allDevices": "",
        "allApplications": "",
        "allDeviceTypes": "",
        "allLogicalInterfaces": "",
```

```
        "allEvents": true,
        "allCommands": "",
        "allFormats": "",
        "qos": 0,
        "x": 70,
        "y": 140,
        "wires": [
            [
                "dde0e6ad.e51b68",
                "ace840b4.e9014",
                "811784ad.111378",
                "68abc0e2.cc756"
            ]
        ]
    },
    {
        "id": "dde0e6ad.e51b68",
        "type": "debug",
        "z": "8fab44bd.a90a78",
        "name": "",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "x": 290,
        "y": 200,
        "wires": []
    },
    {
        "id": "ace840b4.e9014",
        "type": "function",
        "z": "8fab44bd.a90a78",
        "name": "Temperature",
        "func": "msg.payload=msg.payload.d.temperature\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 300,
        "y": 80,
        "wires": [
            [
                "dde0e6ad.e51b68",
                "2c5a1cb7.66dab4",
                "18710b82.ee1814"
            ]
        ]
    },
    {
        "id": "811784ad.111378",
        "type": "function",
        "z": "8fab44bd.a90a78",
        "name": "Humidity",
        "func": "msg.payload=msg.payload.d.humidity\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 280,
        "y": 120,
        "wires": [
            [
                "dde0e6ad.e51b68",
```

```
            "2a63b2e5.7f752e",
            "7403cd54.192ce4"
        ]
    ]
},
{
    "id": "2c5a1cb7.66dab4",
    "type": "ui_gauge",
    "z": "8fab44bd.a90a78",
    "name": "",
    "group": "3c243370.98abdc",
    "order": 0,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": " IBM Waston Sensor Temperature",
    "label": "°C",
    "format": "{{value}}",
    "min": 0,
    "max": "40",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 600,
    "y": 80,
    "wires": []
},
{
    "id": "2a63b2e5.7f752e",
    "type": "ui_gauge",
    "z": "8fab44bd.a90a78",
    "name": "",
    "group": "3c243370.98abdc",
    "order": 0,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": "IBM Waston Sensor Humidity",
    "label": "%",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 580,
    "y": 120,
    "wires": []
},
{
    "id": "68abc0e2.cc756",
    "type": "function",
    "z": "8fab44bd.a90a78",
```

```
        "name": "Soil Temperature",
        "func": "msg.payload=msg.payload.d.objectTemp\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 310,
        "y": 160,
        "wires": [
            [
                "f31ac5a2.181198",
                "f1d8cb5e.6ae208",
                "dde0e6ad.e51b68"
            ]
        ]
    },
    {
        "id": "f31ac5a2.181198",
        "type": "ui_gauge",
        "z": "8fab44bd.a90a78",
        "name": "",
        "group": "3c243370.98abdc",
        "order": 7,
        "width": 0,
        "height": 0,
        "gtype": "gage",
        "title": "IBM Sensor Soil Temperature",
        "label": "°C",
        "format": "{{value}}",
        "min": 0,
        "max": "40",
        "colors": [
            "#00b500",
            "#e6e600",
            "#ca3838"
        ],
        "seg1": "",
        "seg2": "",
        "x": 580,
        "y": 200,
        "wires": []
    },
    {
        "id": "18710b82.ee1814",
        "type": "ui_text",
        "z": "8fab44bd.a90a78",
        "group": "3c243370.98abdc",
        "order": 5,
        "width": 0,
        "height": 0,
        "name": "",
        "label": "IBM Watson Sensor Temperature (°C)",
        "format": "{{msg.payload}}",
        "layout": "row-spread",
        "x": 610,
        "y": 40,
        "wires": []
    },
    {
        "id": "7403cd54.192ce4",
        "type": "ui_text",
        "z": "8fab44bd.a90a78",
        "group": "3c243370.98abdc",
```

```
        "order": 5,
        "width": 0,
        "height": 0,
        "name": "",
        "label": "IBM Watson Sensor Humidity (%)",
        "format": "{{msg.payload}}",
        "layout": "row-spread",
        "x": 600,
        "y": 160,
        "wires": []
    },
    {
        "id": "f1d8cb5e.6ae208",
        "type": "ui_text",
        "z": "8fab44bd.a90a78",
        "group": "3c243370.98abdc",
        "order": 5,
        "width": 0,
        "height": 0,
        "name": "",
        "label": "IBM Sensor Soil Temperature (°C)",
        "format": "{{msg.payload}}",
        "layout": "row-spread",
        "x": 600,
        "y": 240,
        "wires": []
    },
    {
        "id": "c00e705c.cffb3",
        "type": "inject",
        "z": "8fab44bd.a90a78",
        "name": "HTTP Inject",
        "topic": "",
        "payload": "",
        "payloadType": "date",
        "repeat": "10",
        "crontab": "",
        "once": true,
        "onceDelay": "0.1",
        "x": 110,
        "y": 280,
        "wires": [
            [
                "cd4ebeaf.9f68a"
            ]
        ]
    },
    {
        "id": "cd4ebeaf.9f68a",
        "type": "http request",
        "z": "8fab44bd.a90a78",
        "name": "",
        "method": "GET",
        "ret": "obj",
        "paytoqs": false,
        "url":
"http://api.openweathermap.org/data/2.5/weather?q=Mumbai&units=metric&appid
=577a1cbdcb2a11d51cc74f5f5962791a",
        "tls": "",
        "persist": false,
        "proxy": "",
```

```
    "authType": "",
    "x": 270,
    "y": 280,
    "wires": [
        [
            "b64ad60d.4bb4b8",
            "432a3d64.d27124",
            "38059f89.05df"
        ]
    ]
},
{
    "id": "b64ad60d.4bb4b8",
    "type": "function",
    "z": "8fab44bd.a90a78",
    "name": "TemperatureAPI",
    "func": "msg.payload=msg.payload.main.temp\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 460,
    "y": 280,
    "wires": [
        [
            "34ddf39f.c565dc",
            "79eca53d.2c641c"
        ]
    ]
},
{
    "id": "34ddf39f.c565dc",
    "type": "ui_gauge",
    "z": "8fab44bd.a90a78",
    "name": "",
    "group": "5c872dc1.21f164",
    "order": 8,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": "Temperature",
    "label": "°C",
    "format": "{{value}}",
    "min": 0,
    "max": "40",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 630,
    "y": 320,
    "wires": []
},
{
    "id": "4957277d.6043c8",
    "type": "ui_gauge",
    "z": "8fab44bd.a90a78",
    "name": "",
    "group": "5c872dc1.21f164",
    "order": 9,
```

```
        "width": 0,
        "height": 0,
        "gtype": "gage",
        "title": "Humidity",
        "label": "%",
        "format": "{{value}}",
        "min": 0,
        "max": "100",
        "colors": [
            "#00b500",
            "#e6e600",
            "#ca3838"
        ],
        "seg1": "",
        "seg2": "",
        "x": 620,
        "y": 360,
        "wires": []
    },
    {
        "id": "432a3d64.d27124",
        "type": "function",
        "z": "8fab44bd.a90a78",
        "name": "Humidity",
        "func": "msg.payload=msg.payload.main.humidity\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 440,
        "y": 320,
        "wires": [
            [
                "4957277d.6043c8",
                "1a959367.7575fd"
            ]
        ]
    },
    {
        "id": "79eca53d.2c641c",
        "type": "ui_text",
        "z": "8fab44bd.a90a78",
        "group": "5c872dc1.21f164",
        "order": 5,
        "width": 0,
        "height": 0,
        "name": "",
        "label": "Real-time Temperature (°C)",
        "format": "{{msg.payload}}",
        "layout": "row-spread",
        "x": 680,
        "y": 280,
        "wires": []
    },
    {
        "id": "1a959367.7575fd",
        "type": "ui_text",
        "z": "8fab44bd.a90a78",
        "group": "5c872dc1.21f164",
        "order": 5,
        "width": 0,
        "height": 0,
        "name": "",
```

```
        "label": "Real-time Humidity (%)",
        "format": "{{msg.payload}}",
        "layout": "row-spread",
        "x": 670,
        "y": 400,
        "wires": []
    },
    {
        "id": "38059f89.05df",
        "type": "function",
        "z": "8fab44bd.a90a78",
        "name": "FeelsLike",
        "func": "msg.payload=msg.payload.main.feels_like\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 440,
        "y": 420,
        "wires": [
            [
                "b83b554b.f16538",
                "84fd029d.60bce"
            ]
        ]
    },
    {
        "id": "b83b554b.f16538",
        "type": "ui_gauge",
        "z": "8fab44bd.a90a78",
        "name": "",
        "group": "5c872dc1.21f164",
        "order": 8,
        "width": 0,
        "height": 0,
        "gtype": "gage",
        "title": "FeelsLike Temperature",
        "label": "°C",
        "format": "{{value}}",
        "min": 0,
        "max": "40",
        "colors": [
            "#00b500",
            "#e6e600",
            "#ca3838"
        ],
        "seg1": "",
        "seg2": "",
        "x": 660,
        "y": 440,
        "wires": []
    },
    {
        "id": "84fd029d.60bce",
        "type": "ui_text",
        "z": "8fab44bd.a90a78",
        "group": "5c872dc1.21f164",
        "order": 5,
        "width": 0,
        "height": 0,
        "name": "",
        "label": "FeelsLike Temperature (°C)",
        "format": "{{msg.payload}}",
```

```
        "layout": "row-spread",
        "x": 680,
        "y": 480,
        "wires": []
    },
    {
        "id": "d8ff56c3.89cc68",
        "type": "ibmiot out",
        "z": "8fab44bd.a90a78",
        "authentication": "apiKey",
        "apiKey": "7dd0c222.d9446c",
        "outputType": "cmd",
        "deviceId": "1234",
        "deviceType": "Data",
        "eventCommandType": "blink",
        "format": "json",
        "data": "Data",
        "qos": 0,
        "name": "IBM IoT",
        "service": "registered",
        "x": 740,
        "y": 560,
        "wires": []
    },
    {
        "id": "b4ee8e7b.1e364",
        "type": "ui_button",
        "z": "8fab44bd.a90a78",
        "name": "MOTOR ON",
        "group": "95ec373b.008dd8",
        "order": 1,
        "width": 4,
        "height": 2,
        "passthru": false,
        "label": "WaterPump ON",
        "tooltip": "",
        "color": "",
        "bgcolor": "#008000",
        "icon": "",
        "payload": "true",
        "payloadType": "bool",
        "topic": "",
        "x": 110,
        "y": 520,
        "wires": [
            [
                "832d7371.b7f0f"
            ]
        ]
    },
    {
        "id": "8f5e6fc7.0fa44",
        "type": "ui_button",
        "z": "8fab44bd.a90a78",
        "name": "MOTOR OFF",
        "group": "95ec373b.008dd8",
        "order": 2,
        "width": 4,
        "height": 2,
        "passthru": false,
        "label": "WaterPump OFF",
```

```
        "tooltip": "",
        "color": "",
        "bgcolor": "#FF0000",
        "icon": "",
        "payload": "false",
        "payloadType": "bool",
        "topic": "",
        "x": 100,
        "y": 560,
        "wires": [
            [
                "832d7371.b7f0f"
            ]
        ]
    },
    {
        "id": "832d7371.b7f0f",
        "type": "function",
        "z": "8fab44bd.a90a78",
        "name": "MOTOR_STATUS_CHECK",
        "func": "if(msg.payload === true)\n    msg.payload =
'{\"motor\":\"ON\"}'\nelse\n    msg.payload = '{\"motor\":\"OFF\"}'\nreturn
msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 360,
        "y": 540,
        "wires": [
            [
                "864d1114.f5ca6",
                "b4f33b63.71c3e8"
            ]
        ]
    },
    {
        "id": "864d1114.f5ca6",
        "type": "function",
        "z": "8fab44bd.a90a78",
        "name": "Status",
        "func": "var
data=JSON.parse(msg.payload);\nmsg.payload=data.motor;\nreturn msg;\n",
        "outputs": 1,
        "noerr": 0,
        "x": 570,
        "y": 520,
        "wires": [
            [
                "bf60451b.fd8be8",
                "f2e6273b.57ac58",
                "9087050c.0e4fd8"
            ]
        ]
    },
    {
        "id": "b4f33b63.71c3e8",
        "type": "json",
        "z": "8fab44bd.a90a78",
        "name": "MOTOR",
        "property": "payload",
        "action": "",
        "pretty": false,
```

```
        "x": 580,
        "y": 560,
        "wires": [
            [
                "d8ff56c3.89cc68"
            ]
        ]
    },
    {
        "id": "bf60451b.fd8be8",
        "type": "ui_text",
        "z": "8fab44bd.a90a78",
        "group": "95ec373b.008dd8",
        "order": 3,
        "width": 0,
        "height": 0,
        "name": "",
        "label": "MOTOR STATUS:",
        "format": "{{msg.payload}}",
        "layout": "row-spread",
        "x": 750,
        "y": 520,
        "wires": []
    },
    {
        "id": "f2e6273b.57ac58",
        "type": "ui_toast",
        "z": "8fab44bd.a90a78",
        "position": "top right",
        "displayTime": "3",
        "highlight": "",
        "sendall": true,
        "outputs": 0,
        "ok": "OK",
        "cancel": "",
        "raw": true,
        "topic": "",
        "name": "Notif",
        "x": 910,
        "y": 520,
        "wires": []
    },
    {
        "id": "9087050c.0e4fd8",
        "type": "ui_audio",
        "z": "8fab44bd.a90a78",
        "name": "Audio",
        "group": "95ec373b.008dd8",
        "voice": "en-US",
        "always": true,
        "x": 910,
        "y": 560,
        "wires": []
    },
    {
        "id": "7dd0c222.d9446c",
        "type": "ibmiot",
        "z": "",
        "name": "",
        "keepalive": "60",
        "serverName": "",
```

```
        "cleansession": true,
        "appId": "",
        "shared": false
    },
    {
        "id": "3c243370.98abdc",
        "type": "ui_group",
        "z": "",
        "name": "IBM Watson",
        "tab": "8314a149.d930b",
        "order": 1,
        "disp": true,
        "width": "6",
        "collapse": false
    },
    {
        "id": "5c872dc1.21f164",
        "type": "ui_group",
        "z": "",
        "name": "Realt-time Weather",
        "tab": "8314a149.d930b",
        "order": 1,
        "disp": true,
        "width": "6",
        "collapse": false
    },
    {
        "id": "95ec373b.008dd8",
        "type": "ui_group",
        "z": "",
        "name": "Motor Control",
        "tab": "e07a6d62.98f32",
        "order": 1,
        "disp": true,
        "width": 8,
        "collapse": false
    },
    {
        "id": "8314a149.d930b",
        "type": "ui_tab",
        "z": "",
        "name": "IBM Watson Sensor Data",
        "icon": "dashboard",
        "disabled": false,
        "hidden": false
    },
    {
        "id": "e07a6d62.98f32",
        "type": "ui_tab",
        "z": "",
        "name": "Controls",
        "icon": "dashboard",
        "disabled": false,
        "hidden": false
    }
]
```