A project report on IoT based Smart Agriculture System

1. NTRODUCTION

The objectives of this report is to propose IoT based Smart Agriculture System which will enable farmers to have live data of soil moisture environment temperature at very

low cost so that live monitoring can be done.
The various services used in the development of this project are listed as follows:

DEFINITION IOT BASED SMART AGRICULTURE SYSTEM

IoT based SMART AGRICULTURE SYSTEM is regarded as IoT gadget focusing on Live Monitoring of Environmental data in terms of Temperature, Moisture and other types depending on the sensors integrated with it. The system provides the concept of "Plug & Sense" in which farmers can directly implement smart farming by as such putting the System on the field and getting Live Data feeds on various devices like Smart Phones, Tablets etc. and the data generated via sensors can be easily shared and viewed by agriculture consultants anywhere remotely via Cloud Computing technology integration.

PURPOSE

Smart agriculture involves integration of advanced technologies into already persisting agricultural practices with a view to boost production quality and efficiency for farming products. It helps in automated farming with collection of data for further analysis to provide the operator with accurate information for better decision making to gain high quality output of the product. A technically sound agriculture system rooted on observing, measuring and responding to inter and intra-field variableness in products. The global of smart agriculture research is to ground a decision making support system for farm management. A system that optimizes and examines how high-tech farming can aid to production output as well as focuses on the preservation of resources.

By providing them with the benefits of technological advancements, smart agriculture aims to reduce the heavy workload of the farm workers, hence improving their quality of life. Smart agriculture deems it necessary to address the issues of population growth, climate change and labour that gained a lot of technological attention, from planting and watering of crops to health and harvesting.

2. LITERARY SURVEY

EXISTING PROBLEM

The global population is predicted to touch 9.6 billion by 2050 – this poses a big problem for the agriculture industry. Despite combating challenges like extreme weather conditions, rising climate change, and farming's environmental impact, the demand for more food has to be met. To meet these increasing needs, agriculture has to turn to new technology. New smart farming applications based on IoT technologies will enable the agriculture industry to reduce waste and enhance productivity from optimizing fertilizer use to increasing the efficiency of farm vehicles' routes.

PROPOSED SOLUTION

Smart farming is a capital-intensive and hi-tech system of growing food cleanly and sustainable for the masses. It is the application of modern ICT (Information and Communication Technologies) into agriculture.

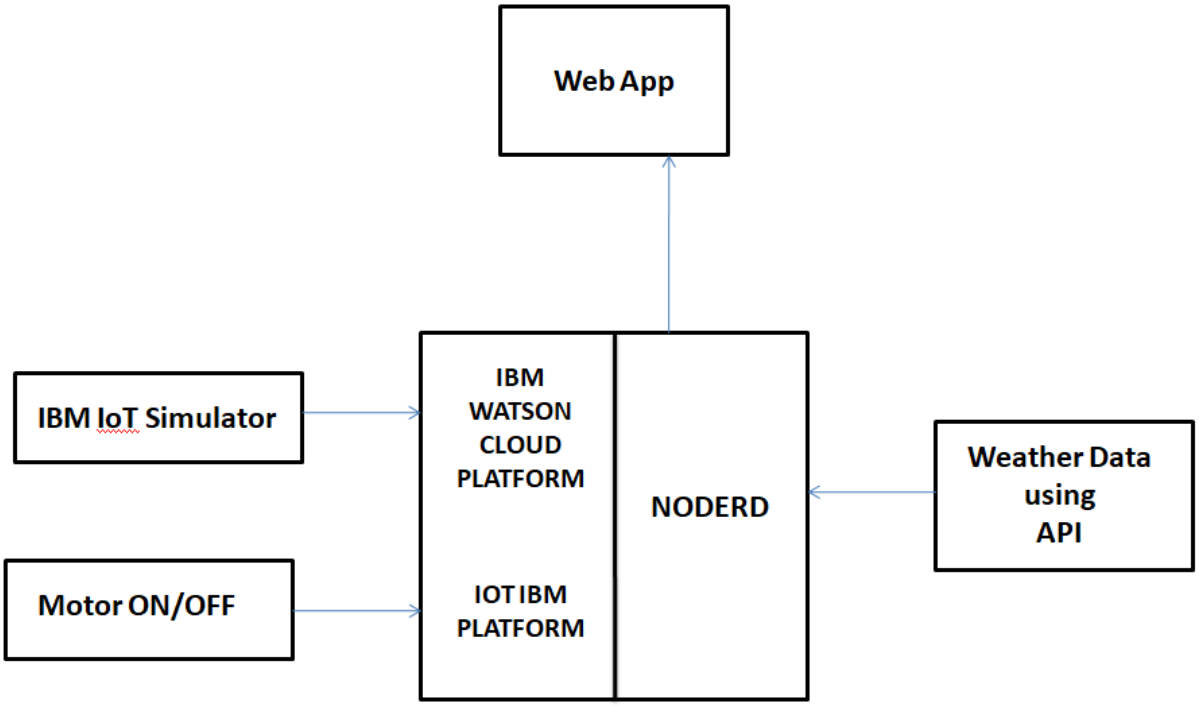
In IoT-based smart farming, a system is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, etc.) and automating the irrigation system. The farmers can monitor the field conditions from anywhere. IoT-based smart farming is highly efficient when compared with the conventional approach.

The applications of IoT-based smart farming not only target conventional, large farming operations, but could also be new levers to uplift other growing or common trends in agricultural like organic farming, family farming (complex or small spaces, particular cattle and/or cultures, preservation of particular or high-quality varieties, etc.), and enhance highly transparent farming.

In terms of environmental issues, IoT-based smart farming can provide great benefits including more efficient water usage, or optimization of inputs and treatments. Now, let's discuss the major applications of IoT-based smart farming that are revolutionizing agriculture.

3. THEORETICAL ANALYSIS

BLOCK DIAGRAM



The development of the web application employs various platforms and service es that, work in sync to provide the user i.e. farmers with the accurate data and for the system to control the devices installed on the fields by taking a command from the farmer.

The services used are listed as below:

A. IBM IoT Platform & IBM Watson Cloud Platform

IBM IoT Platform is a service that helps creating and controlling various virtual services that can further control the devices, the services created for this project on the IBM IoT Platform are IBM Watson IoT Platform service, Continuous Delivery service and the Node Red service the enable and ensure the functioning of various sections of the project.

IBM Watson IoT Platform is a public or private cloud service that lets us add devices that are needed for the user to connect their system to the cloud server and, to collect and deploy data online. These devices require some information for the user to connect the system and access the data such as, it needs you to generate an API Key for linking with any applications if needed.

B. IBM IoT Simulator

IBM IoT Simulator/ Sensor is a simulating service that is designed to upload or collect data from the cloud. It demands various details from the user to connect the simulator/ sensor to the devices created on the IBM Watson IoT Platform such as Device Type, Device ID, Organization ID and Authentication Token, which are essential for establishing a link between the two. A few examples of the sensors are temperature sensor, humidity sensor etc.

C. NODE-RED

Node-RED is a programming tool for wiring together hardware devices, APIs and online services. Primarily, it is a visual tool designed for the Internet of Things, but it can also be used for other applications to very quickly assemble flows of various services.

Node-Red through the various nodes that it offers lets you design a system all on a single platform wherein one can connect all the different types of devices and services to then get a

desired result. Also Node-Red has nodes for designing a user interface (UI), for the web page which is to be accessed by the farmers.

Following are the various nodes used in the development of this Smart Agriculture System:

C.1. IBM IoT Nodes

C.1.1. ibmiot in

Input node that can be used with Watson IoT Platform to receive events sent from devices, receive commands sent to devices, or receive status updates concerning devices or applications. It produces an object called msg and sets **msg.payload** to be a String containing the payload of the incoming message.

C.1.2. ibmiot out

Output node that can be used with Watson IoT Platform to send a commands to a device or send an event on behalf of a device.

C.2. Function Nodes

C.2.1. Function Node

A JavaScript function block to run against the messages being received by the node. The messages are passed in as a JavaScript object called msg. By convention it will have a msg.payload property containing the body of the message. The function is expected to return a message object (or multiple message objects), but can choose to return nothing in order to halt a flow.

The function nodes used in the project are for the selection and display of the properties like Temperature, Humidity and Object Temperature, from the IBM IoT Simulator.

C.2.2. Change Node

Set, change, delete or move properties of a message, flow context or global context.

The node can specify multiple rules that will be applied in the order they are defined. The available operations are:

Set, Cange, Delete and Move.

The functionality of the node in this node-red flow is to change the entire msg.payload into a particular section or quantity of the msg.payload, for eg. the temperature received from the Open Weather Platform is to be displayed on the UI which is filtered from the msg.payload employing a change node as msg.payload.temp .

The other Function nodes that work as per other requirements are:

Switch, template, delay, trigger, execute nodes, etc.

C.3. Network Nodes

C.3.1. http request Node

The http request node sends HTTP requests and returns the response. This node is used for communication between the device and any website in order to read some set of useful date from

a website, in the project a http request node is specifically used for fetching the data i.e. weather details of a particular city using an API Key from the website of Open Weather Platform viz.www.openweathermap.org .

C.3.2. mqtt in Node

Connects to a MQTT broker and subscribes to messages from the specified topic. The subscription topic can include MQTT wildcards, + for one level, # for multiple levels. This node requires a connection to a MQTT broker to be configured. This is configured by clicking the pencil icon.
Several MQTT nodes (in or out) can share the same broker connection if required.

mqtt out, websocket in, websocket out, tcp in, tcp request, are a few other examples of the network nodes.

C.4. Dashboard Nodes

These are the nodes that specifically focus on building and development of the user interface (UI) of the desired web application and are used to display the collected and manipulated, data and in the information in the form of text, gauges, charts etc. Also these provide functionality in the form of buttons, switches, sliders etc.

C.4.1. button Node

Adds a button to the user interface. Clicking the button generates a message
with msg.payload set to the Payload field. If no payload is specified, the node id is used.
The Size defaults to 3 by 1. The Icon can be defined, as either a Material Design icon *(e.g. 'check', 'close')* or a Font Awesome icon *(e.g. 'fa-fire'),* or a Weather icon. You can use the full set of google material icons if you add 'mi-' to the icon name. e.g. 'mi-videogame_asset'. The colours of the text and background may be set. They can also be set by a message property by setting the field to the name of the property, for example {{msg.background}}. The label can also be set by a message property by setting the field to the name of the property, for example {{msg.topic}}. If set to pass through mode a message arriving on the input will act like pressing the button. The output payload will be as defined in the node configuration.
The Topic field can be used to set the msg.topic property that is output.
Setting msg.enabled to false will disable the button.

The two buttons used in the Node-Red flow for turning the motor on and off take the user input when the particular button is pushed, turns the motor on or off and prints the action to the Python Shell.

C.4.2. Guage

Adds a gauge type widget to the user interface. The msg.payload is searched for a numeric *value* and is formatted in accordance with the defined Value Format, which can then be formatted using Angular filters. For example : {{value | number:1}}% will round the value to one decimal place and append a % sign. The colours of each of 3 sectors can be specified and the gauge will blend between them. The colours should be specified in hex (#rrggbb) format. If you specify numbers for the sectors then the colours changes per sector. If not specified the colours are blended across the total range. The gauge has several modes. Regular gauge, donut, compass and wave. The label can also be set by a message property by setting the field to the name of the property, for example {{msg.topic}}.

The gauges used in the project display the temperature, humidity and object temperature received from the IBM IoT Sensor, in the form of a guage.

C.4.3. Text

Will display a non-editable text field on the user interface. Each received msg.payload will update the text based on the provided Value Format. The Value Format field can be used to change the displayed format and can contain valid HTML and Angular filters. For example: {{value | uppercase}} ° will uppercase the payload text and add the degree symbol. The label can also be set by a message property by setting the field to the name of the property, for example {{msg.topic}}. The following icon fonts are also available: Material Design icon *(e.g. 'check', 'close')* or a Font Awesome icon *(e.g. 'fa-fire')*, or a Weather icon. You can use the full set of google material icons if you add 'mi-' to the icon name. e.g. 'mi-videogame_asset'. The widget also has a class of nr-dashboard-widget-{the_widget_label_with_underscores} which can be used for additional styling if required. You may need to use the *!important* flag to override the theme.

The text Nodes used in the project display the temperature, humidity and atmospheric pressure received from the Open Weather Platform, in the form of text to the webpage.

Other dashboard nodes that add various functionalities to the UI are dropdown, switch, slider, numeric, chart, notification etc.

C.4. Common Nodes

Common nodes comprise of inject node, debug, status, comment nodes etc.

The debug node out of all the common nodes is employed in the project. It displays selected message properties in the debug sidebar tab and optionally the runtime log. By default it displays msg.payload, but can be configured to display any property, the full message or the result of a JSONata expression. The debug sidebar provides a structured view of the messages it is sent, making it easier to understand their structure.
JavaScript objects and arrays can be collapsed and expanded as required. Buffer objects can be displayed as raw data or as a string if possible.

Alongside each message, the debug sidebar includes information about the time the message was received, the node that sent it and the type of the message. Clicking on the source node id will reveal that node within the workspace.

The button on the node can be used to enable or disable its output. It is recommended to disable or remove any Debug nodes that are not being used.

The node can also be configured to send all messages to the runtime log, or to send short (32 characters) to the status text under the debug node.

D. OPEN WEATHER PLATFORM

Open Weather Platform is like a weather forecast service that enables a user to get the information and weather details of any particular city or location at any part of the world using the name of the place or the latitudes and longitudes of a location. Open Weather Platform also offers a five day weather forecast for a city which can be very useful for the future reference to the user, to schedule the operation of devices on the fields.

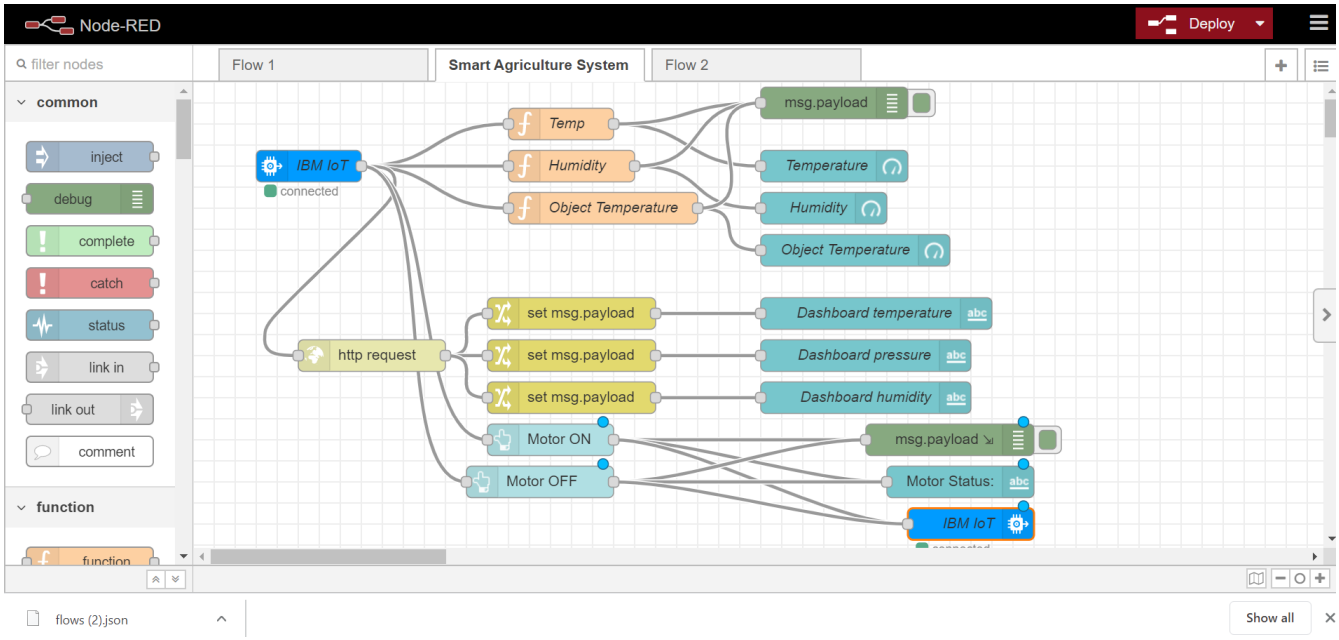
In the project the UI shows the weather details of a location using the Application Programming Interface (API). A network node which in here is a http request node requests data from the official website of the Open Weather Platform which is accessed using an API Key generated beforehand.

SOFTWARE DESIGNING

Step 1: Create IBM Cloud account.
Step 2: Install Node-Red locally.
Step 3: Create IOT Service, Node-Red Service and Continuous Delivery Service on IBM cloud.
Step 4: Create required devices on the IBM Watson IoT Platform and connect it to the IBM IoT Sensor.
Step 5: Installation of Python IDE.

Step 6: Install all the required Node-Red nodes.
Step 7: Connect the node to your IBM IoT device to get the simulator data.
Step 8: Create an account in Open Weather API and configure Open Weather API Platform.
Step 9: Configure the Node-Red to get the weather forecasting data using the http request node.
Step 10: Configure the nodes to display the weather parameters which are obtained from the IoT Simulator and the Open Weather API in the user interface (UI).
Step 11: Configure the nodes for creating buttons and sending commands to IoT Platform.
Step 12: Write a Python code to subscribe to IBM IoT Platform and get the commands.

4. NODE-RED FLOW



All the required nodes and devices are connected using the wired connection on Node-Red which would then form the UI of the web application for the user to access all the information and control the devices.

5. PYTHON CODE

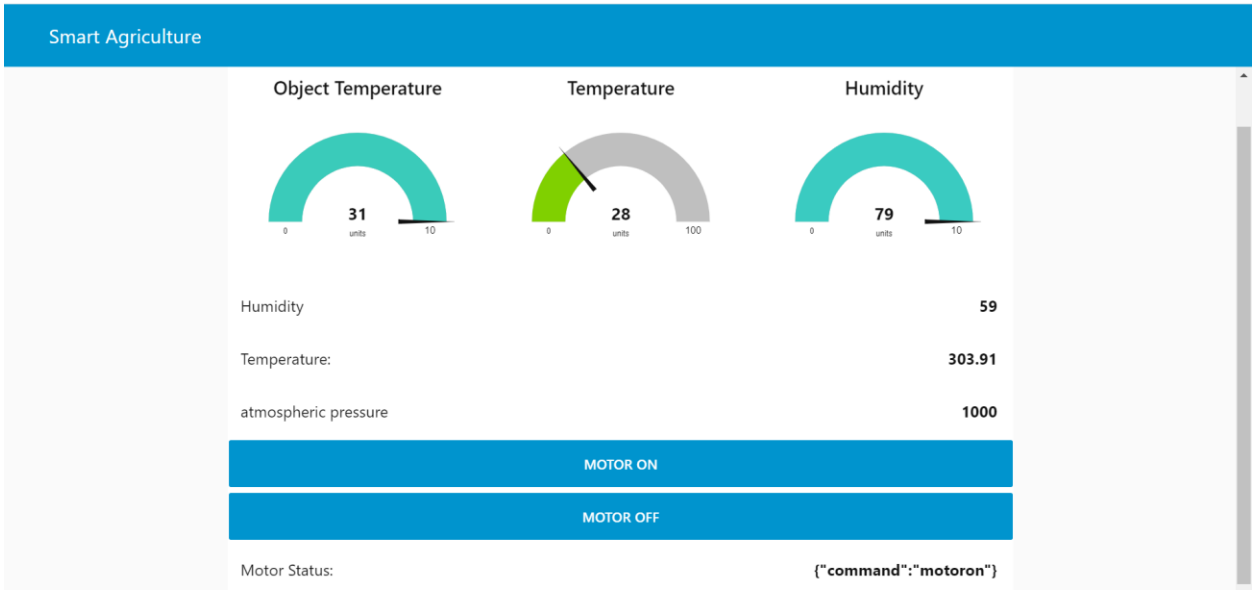
```
import time

import sys

import ibmiotf.application # to install pip install ibmiotf

import ibmiotf.device


#Provide your IBM Watson Device Credentials

organization = "7rw5ks" #replace the ORG ID

deviceType = "IOTDevice"#replace the Device type wi

deviceId = "Arduino1"#replace Device ID

authMethod = "token"

authToken = "9584745679" #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()
```

6. RESULT

User is served with the Web Application with a user interface to interact with the cloud, sensors and the collected data and to control the working of the devices employed on the field. The farmers can then use the web application to monitor the relevant weather conditions and control the environment. A picture of the UI is displayed as below:



7. ADVANTAGES

A. It allows farmers to maximize yields using minimum resources such as water, fertilizers, seeds etc.

B. Solar powered and mobile operated pumps save cost of electricity.

C. Smart agriculture use drones and robots which helps in many ways. These improves data collection process and helps in wireless monitoring and control.

D. It is cost effective method.

E. It delivers high quality crop production.

DISADVANTAGES

A. 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.

B. 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.

8. APPLICATIONS

With the adoption of IoT in various areas like Industry, Homes and even Cities, huge potential is seen to make everything Intelligent and Smart. Even the Agricultural sector is also adopting IoT technology these days and this in turn has led to the development of "AGRICULTURE Internet of Things (IoT)".

Application Name	Description
Crop Water Management	In order to perform agriculture activities in inefficient manner, adequate water is essential. Agriculture IoT is integrated with Web Map Service (WMS) and Sensor Observation Service (SOS) to ensure proper water management for irrigation and in turn reduces water wastage.
Precision Agriculture	High accuracy is required is required in terms of weather information which reduces the chances of crop damage. Agriculture IoT ensures timely delivery of real time data in terms of weather forecasting, quality of soil, cost of labour and much more to farmer.
Integrated Pest Management or Control (IPM/C)	Agriculture IoT systems assures farmers with accurate environmental data via proper live data monitoring of temperature, moisture, plant growth and level of pests so that proper care can be taken during production
Food Production & Safety	Agriculture IoT system accurately monitors various parameters like warehouse temperature, shipping transportation management system and also integrates cloud based recording systems.

9. CONCLUSION

IoT based SMART FARMING SYSTEM for Live Monitoring of Temperature and Soil Moisture has been proposed using Arduino and Cloud Computing . The System has high efficiency and accuracy in fetching the live data of temperature and soil moisture. The IoT based smart farming System being proposed via this report will assist farmers in increasing the agriculture yield and take efficient care of food production as the System will always provide helping hand to farmers for getting accurate live feed of environmental temperature and soil moisture with more than 99% accurate results.

10. FUTURE SCOPE

Future work would be focused more on increasing sensors on this system to fetch more data especially with regard to Pest Control and by also integrating GPS module in this system to enhance this Agriculture IoT Technology to full-fledged Agriculture Precision ready product.

11. BIBILOGRAGHY

11.1 https//:www.wikipedia.org

11.2 https//:www.youtube.com

11.3 https://www.youtube.com/watch?v=w3jLJU7DT5E&feature=youtu.be

11.4 https://nodered.org/

11.5 https//:www.researchgate.net

11.6 https://www.iotforall.com/