SUMMER INTERNSHIP REPORT

SMART AGRICULTURE SYSTEM BASED ON IOT

Carried out with

SMARTBRIDGE

Submitted by

SNIGDHA

(41896302817)

Submitted in partial fulfillment of the requirements for the award of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

NEW DELHI

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

An IoT ecosystem consists of web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware, to collect, send and act on data they acquire from their environments.

The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed. IoT can also make use of artificial intelligence (AI) and machine learning to aid in making data collecting processes easier and more dynamic.

Some of the common benefits of IoT enable businesses to:

1. Monitor their overall business processes
2. Improve the customer experience
3. Save time and money
4. Enhance employee productivity
5. Integrate and adapt business models
6. Make better business decisions
7. Generate more revenue

1.2 PURPOSE

With the continuous increase in world's population, demand for food supply is extremely raised. We know that India is an agricultural economy. Its 55% population is employed by agriculture and it contributes around 14% of the Indian GDP. 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.

Governments are helping farmers to use advanced techniques and research to increase food production. Farmers are using meaningful insights from the data to yield better return on investment. Sensing for soil moisture and nutrients, controlling water usage for plant growth and determining custom fertilizer are some simple uses of IoT.

Smart farming is one of the fastest growing field in IoT. 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. IoT-based smart farming is highly efficient when compared with the conventional approach. In terms of environmental issues, IoT-based smart farming can provide great benefits including more efficient water usage, or optimization of inputs and treatments.

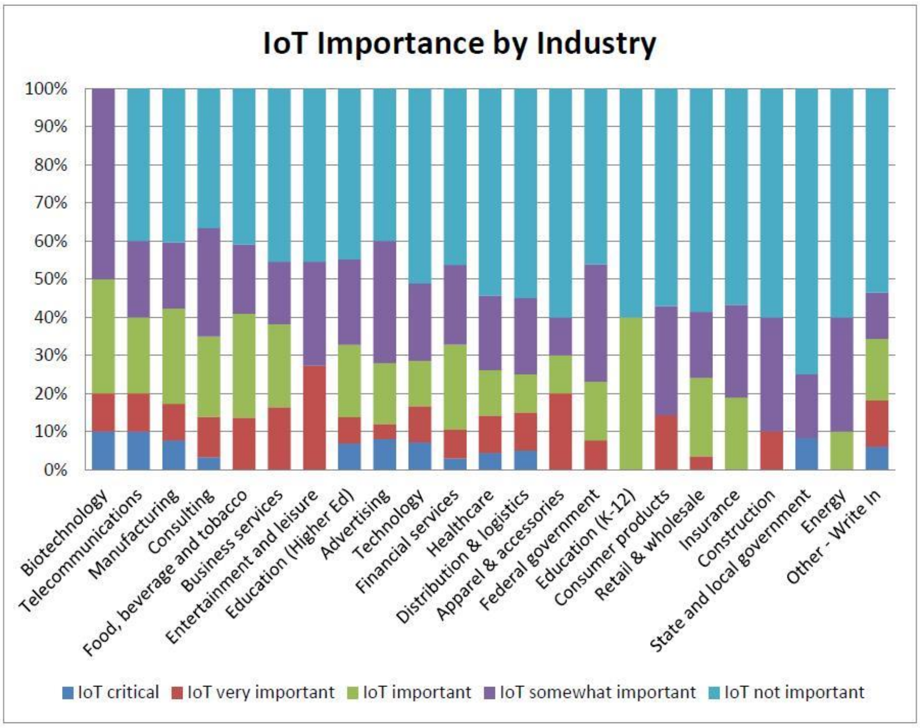


Fig 1.1 IoT importance by industry

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

Agriculture is the oldest industry on earth and since its beginning, it has always been a risky enterprise. It always falls prey to weather, narrow profits and uncertain markets. Other challenges include water shortage, low fertility, pest attack, disease attack, low yield and variety of monetary problems.

In addition to this, farmers are faced with new opportunities every day -- from feeding an expanding global population while meeting strict new emissions requirements, to producing more food on fewer acres while minimizing their environmental footprint.

In fact, agriculture occupies nearly 40% of the earth's surface, far more than any other human activity. In addition, irrigation of agricultural crops comprises 70% of global water use, and agriculture directly contributes to around 11% of global greenhouse gas (GHG) emissions (mostly through cattle). Expanding agricultural land can also lead to deforestation, additional GHG emissions, and a loss of biodiversity.

These three challenges – feeding a growing population, providing a livelihood for farmers, and protecting the environment – must be tackled together if we are to make sustainable progress in any of them. But making progress is difficult, as initiatives to solve one challenge can have an unintended consequences on another.

Sometimes, the consequences are positive. For instance, raising farm productivity can generate income growth in agriculture, make more food available for consumers at lower prices, and – in some cases – reduce pressure on the environment. But sometimes the consequences are negative and require balancing trade-offs. For example, policies to increase the environmental sustainability of agriculture could impose increased costs on farmers and lead to higher prices for consumers.

2.2 PROPOSED SOLUTION

In the past, these challenges were solved by using mechanization as it proved to be the key to farming progress for food producers. Unfortunately, the idea of using mechanical equipment is not sufficient enough, therefore farmers are not only forced to embrace digital, but a whole host of innovative technologies to reduce costs, increase efficiency and yield a better return on their investment.

While there is unlikely to be a "one-size-fits-all" solution for dealing with environmental concerns in agriculture, as agro-ecological conditions and public preferences differ vastly, we must have at their disposal a deep understanding of, and capacity to measure, the linkages between solutions and their outcomes in order to evaluate and achieve better environmental outcomes in a cost-effective manner.

To improve the sustainability of agriculture, this project has used the concept of Internet of Things and the state-of-art sensors to precisely measure soil parameters like temperature, humidity and soil moisture. The farmer can have insights on the current values of these variables using a mobile or web application. To identify other conditions like maximum temperature, minimum temperature and wind speed that might impact his yield, real-time weather forecasting is done using an external platform (Open Weather API). The farmer can also use this application for irrigating his field. The motor can be controlled using the application from anywhere as long as each of these devices remain connected to the network.

CHAPTER 3

THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM

The IBM Watson Cloud platform receives the soil parameter values from the IOT sensor and reports these values to the user through a mobile or web application. It also gets the real-time weather forecasting data from weather service provider and updates the user with the current weather of the user's location. After analysing these values, the farmer can control his motor using application to irrigate his field from anywhere in the world.

The following block diagram shows the overall working of the project:

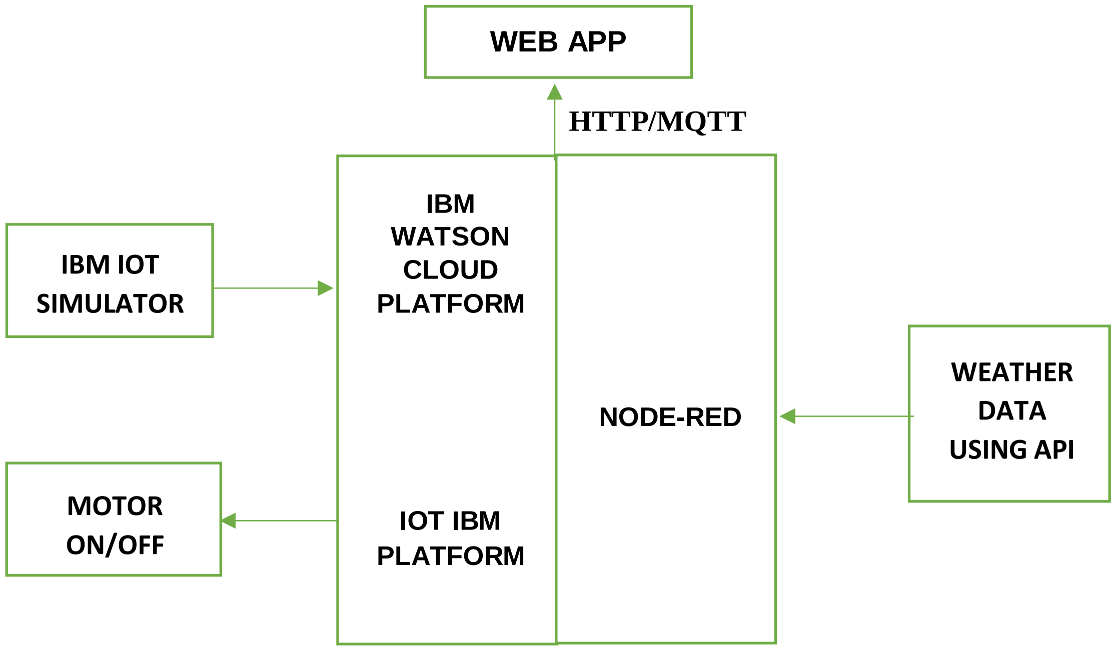


Fig 3.1 Block diagram of the project

3.2 HARDWARE/SOFTWARE DESIGNING

IBM cloud is used to build a web application that receives the data from the sensor. Using the IBM Watson platform, a device is created that connects to an IOT simulator.

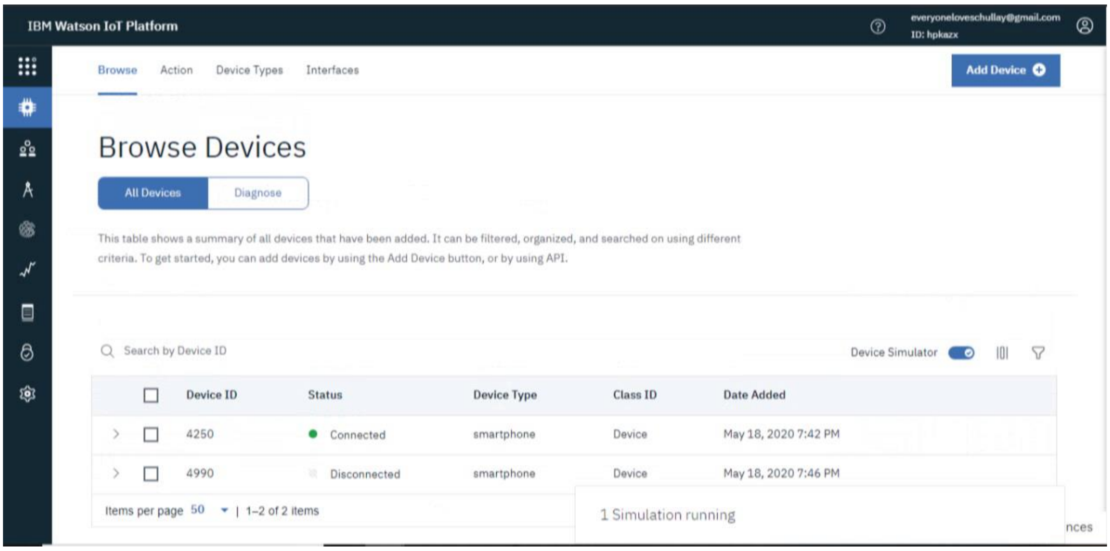


Fig 3.2 IBM Watson platform

IBM Bluemix is used to simulate the sensor for temperature, humidity and soil moisture. This sensor simulator is connected to the device created on the IBM Watson platform using valid credentials.

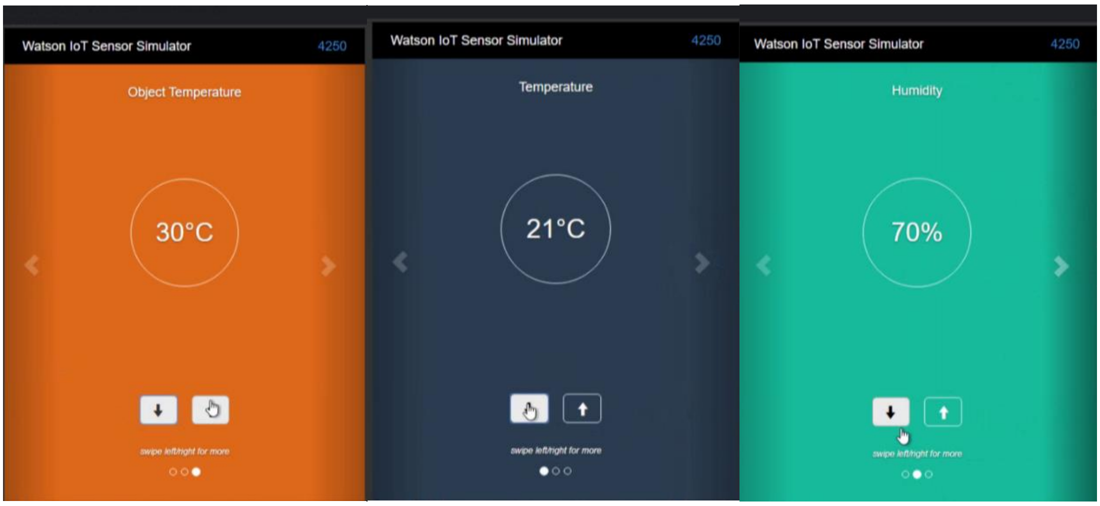


Fig 3.3 IOT sensor simulator

To view the simulator data, cards are created on the Watson platform. The data can be viewed as line chart, bar graph, donut chart, etc. When values are changed in the sensor simulator, they are reflected and analysed on these cards.

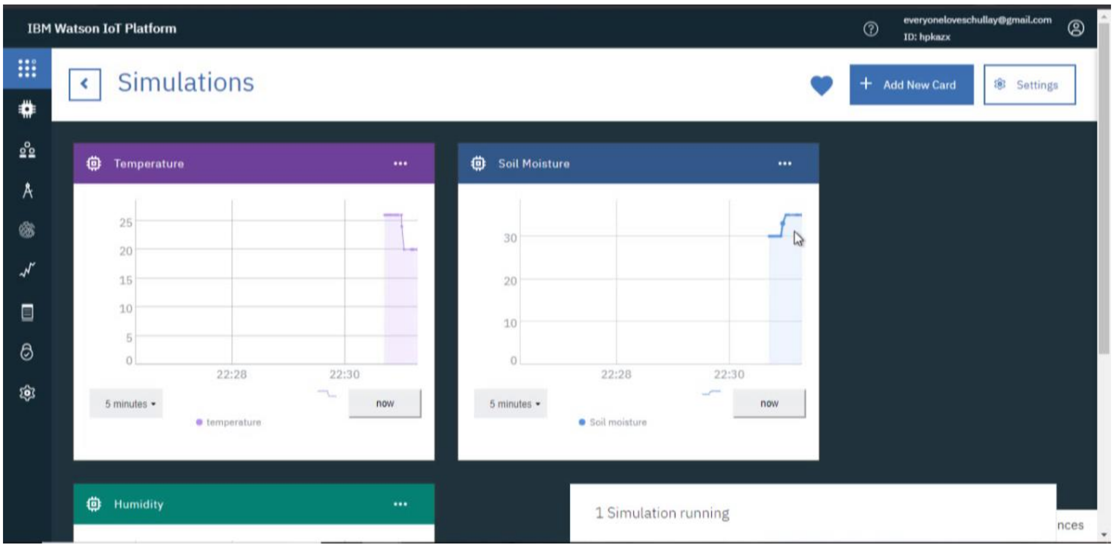


Fig 3.4 Cards to visualize simulator data

Node-RED is a programming tool for wiring together hardware devices, APIs and online services. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click. To create a web application, node-RED is used.

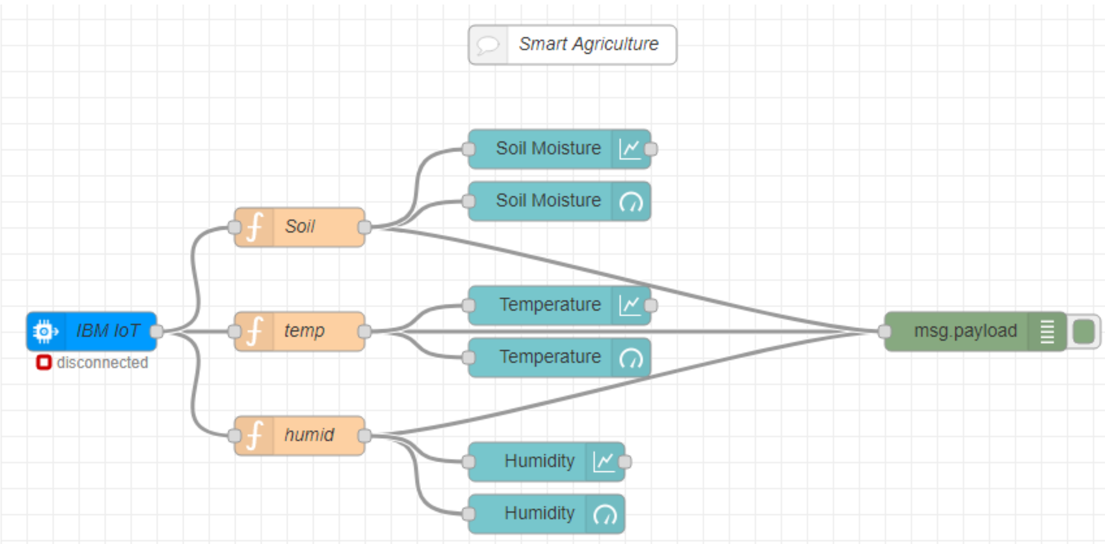


Fig 3.5 Flow to take sensor data from IBM cloud

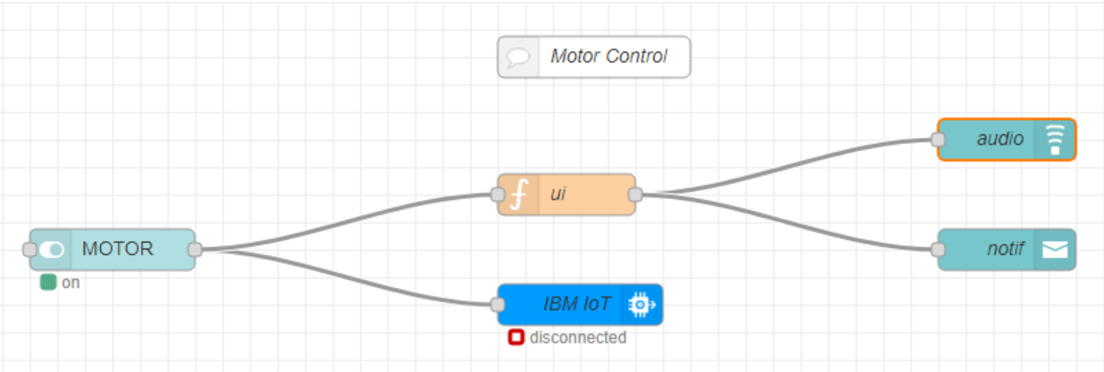


Fig 3.6 Flow to transfer the motor control data to the IBM cloud

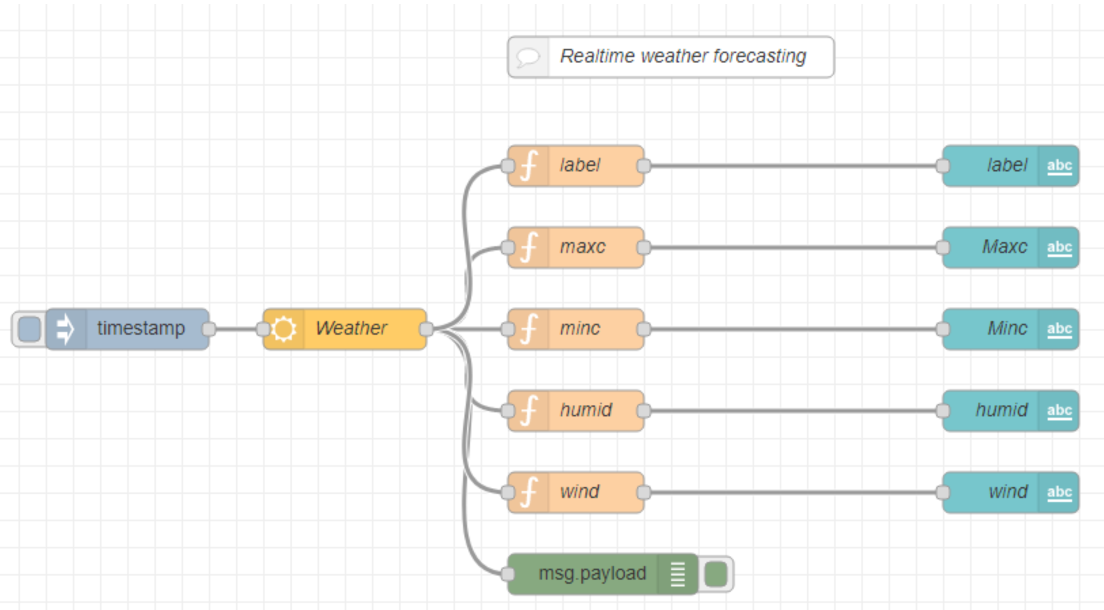


Fig 3.7 Flow to obtain real-time weather data from OpenWeather API

A user friendly UI is designed using the node-RED dashboard. The mentioned flows are displayed in three different tabs. The "Smart Agriculture" tab shows the soil parameter values that are obtained from the IOT sensor. The line graphs provide the user a way to analyse the data variations over a period of time. The "Realtime weather" tab uses the OpenWeather API to show the real-time values of weather, maximum temperature, minimum temperature, humidity and wind speed. Finally, the "Motor control" tab is used to control the motor. When the motor is switched on/off, a notification pops on top of the screen and a voice is outputted that informs the user about the command.

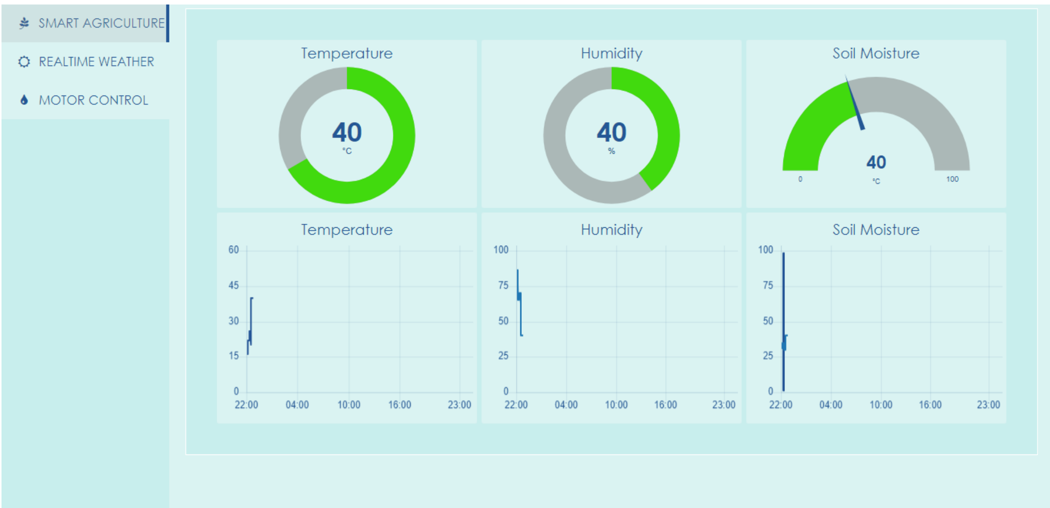


Fig 3.8 Smart Agriculture tab

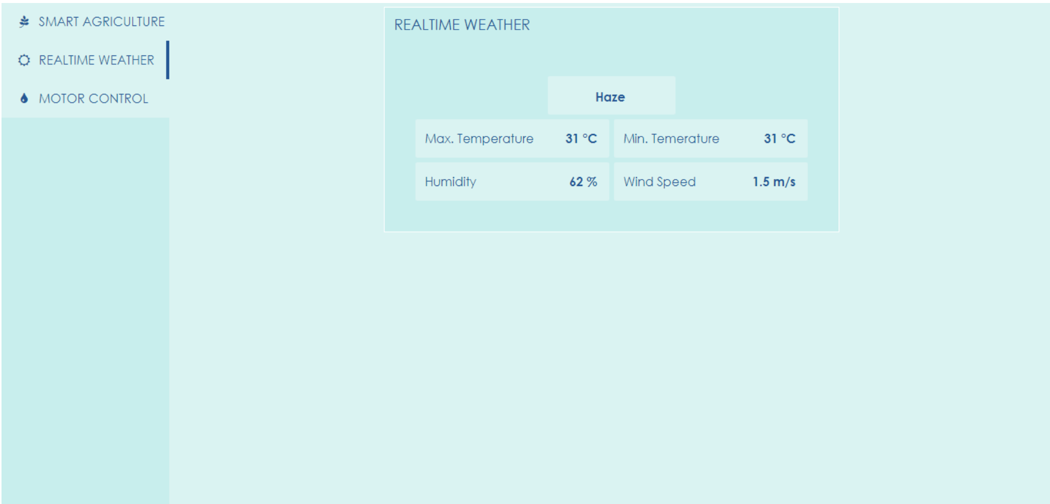


Fig 3.9 Realtime weather tab

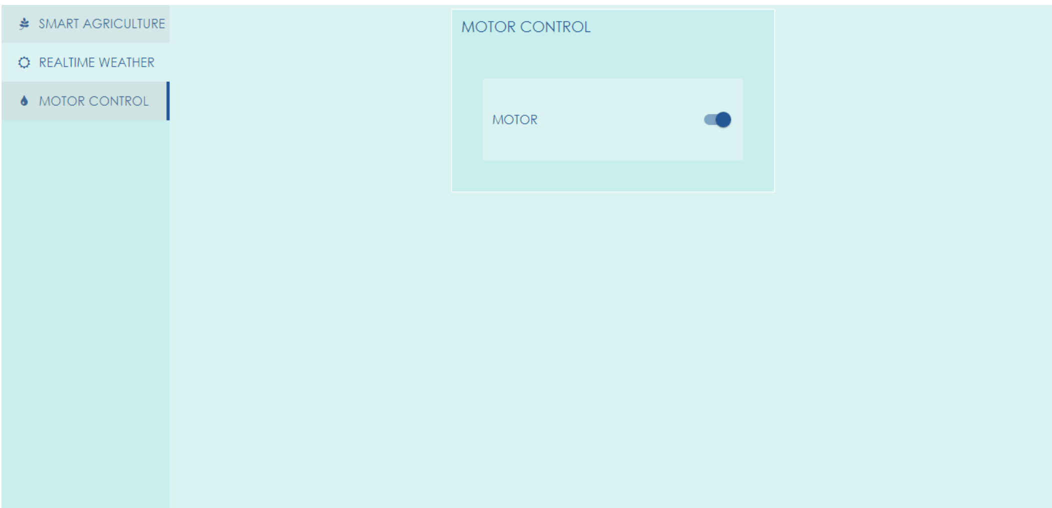


Fig 3.10 Motor control tab

CHAPTER 4

EXPERIMENTAL INVESTIGATION

After creating an IBM cloud account, we use the IBM Watson IoT platform to create a device. An API key is also generated on the Watson platform which will later be used on Node-RED.

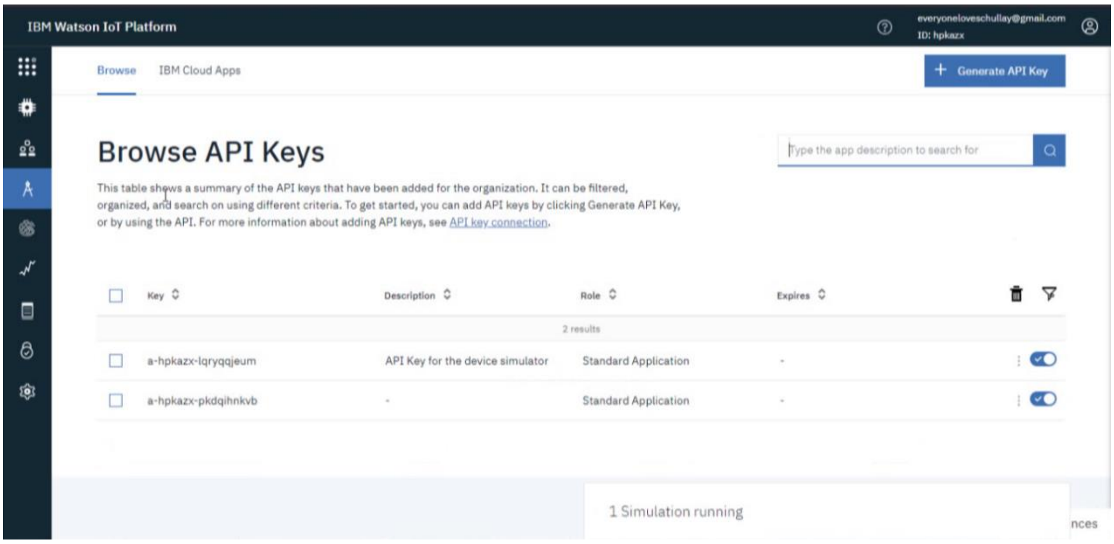
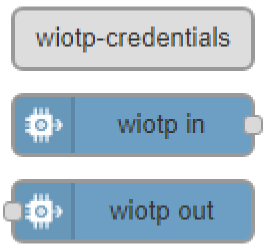


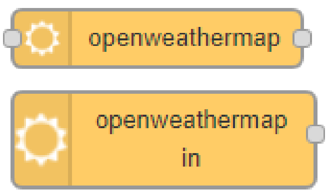
Fig 4.1 API keys on IBM Watson IoT Platform

NODE-RED is a programming tool that is installed for connecting API and the online simulator. Following are the additional required nodes that are installed on the Node-RED flow editor to create the flows:

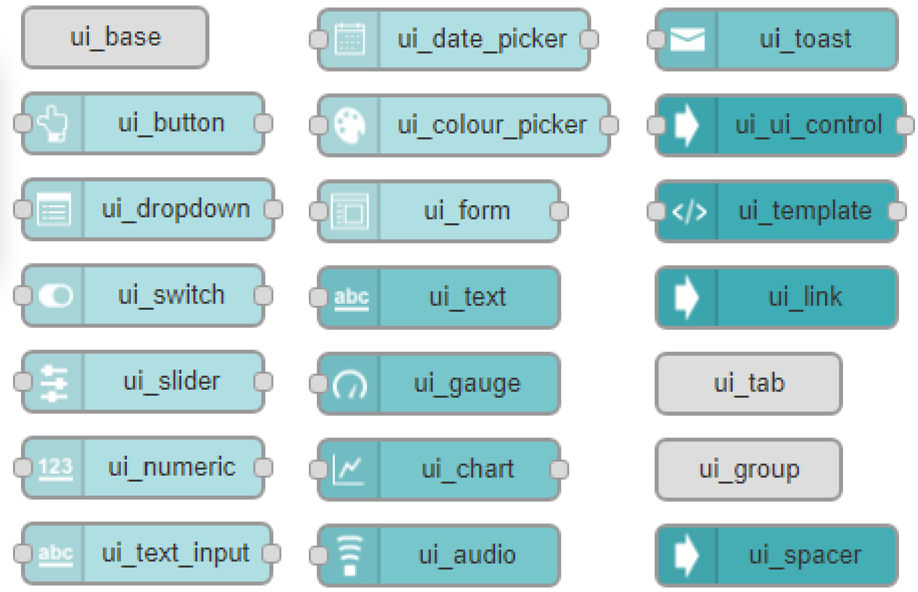
1. node-red-contrib-scx-ibmiotapp



2. node-red-node-openweathermap



3. node-red-dashboard



4.1 SMART AGRICULTURE FLOW

Figure 3.5 shows the smart agriculture flow. The configuration of the respective nodes are described below.

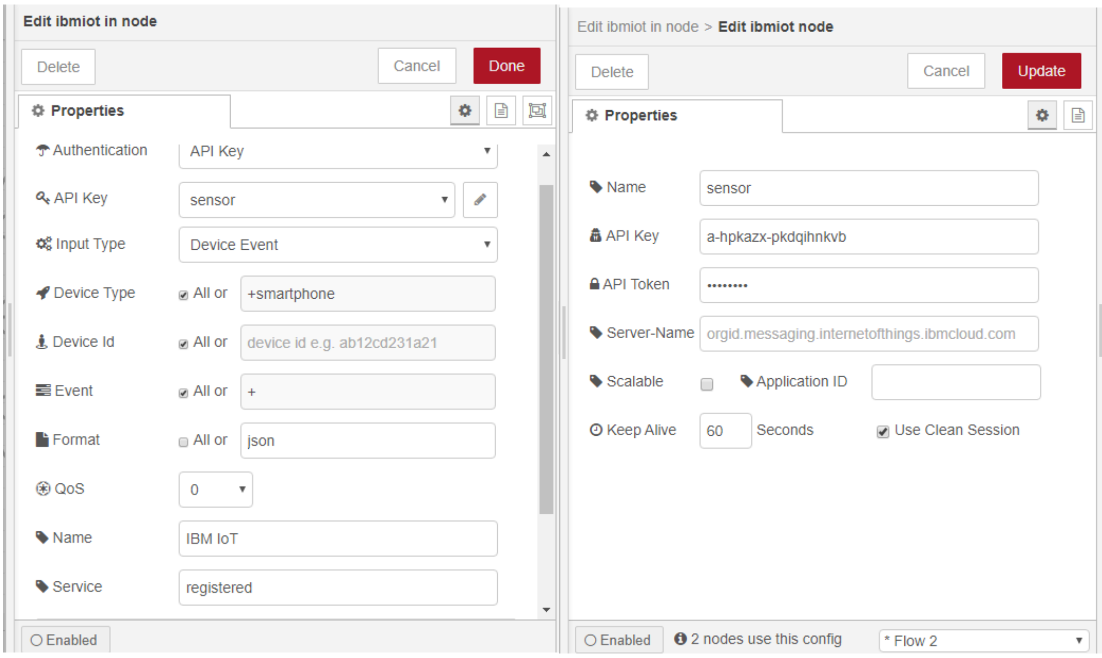


Fig 4.2 ibmiot in node

Code for the function node "Soil":

```
msg.payload=msg.payload.d.objectTemp;
return msg;
```

Code for the function node "temp":

```
msg.payload=msg.payload.d.temperature;
return msg;
```

Code for the function node "humid":

```
msg.payload=msg.payload.d.humidity;
return msg;
```

The data is visualised on the Node-RED UI using the "gauge" and "chart" nodes from the dashboard palette.

4.2 MOTOR CONTROL FLOW

Figure 3.6 shows the motor control flow. The configuration of the respective nodes are described below.

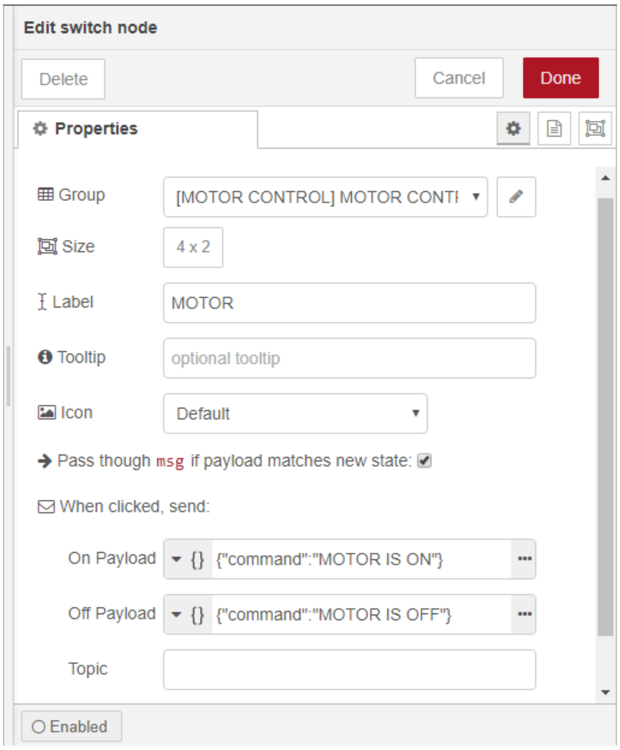


Fig 4.3 Motor switch node

Code for the function node "ui":

```
msg.payload=msg.payload.command;
return msg;
```

The payload from “ui” node is used to give data to the “audio” and “notif” nodes that gives an audio output and a notification every time a motor control command is given.

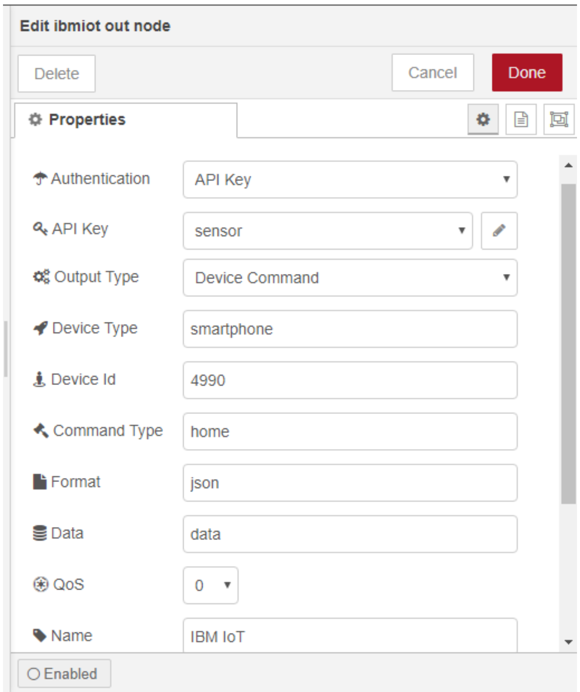


Fig 4.4 ibmiot out node

4.3 REALTIME WEATHER FLOW

Figure 3.7 shows the realtime weather flow. The configuration of the respective nodes are described below.

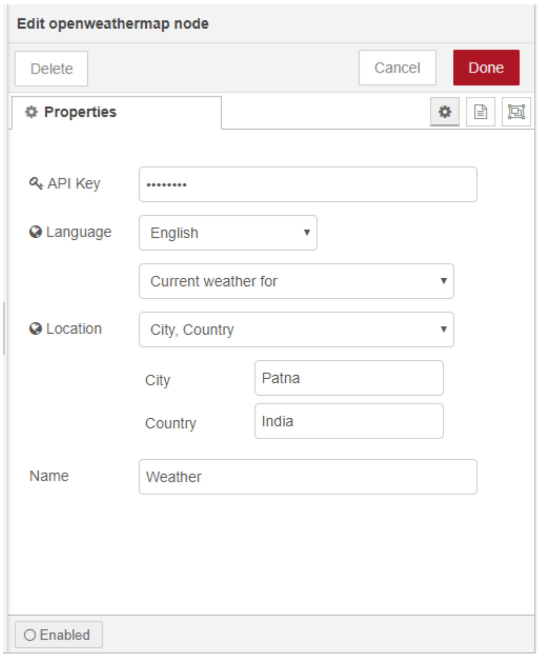


Fig 4.5 openweathermap node

An inject node is used to inject a message into the flow either manually or at regular intervals.

Code for function node “label”:

```
msg.payload=msg.payload.weather;
return msg;
```

Code for function node “maxc”:

```
msg.payload=msg.payload. temp_maxc;
return msg;
```

Code for function node “minc”:

```
msg.payload=msg.payload. temp_minc;
return msg;
```

Code for function node “humid”:

```
msg.payload=msg.payload.humidity;
return msg;
```

Code for function node “wind”:

```
msg.payload=msg.payload.windspeed;
return msg;
```

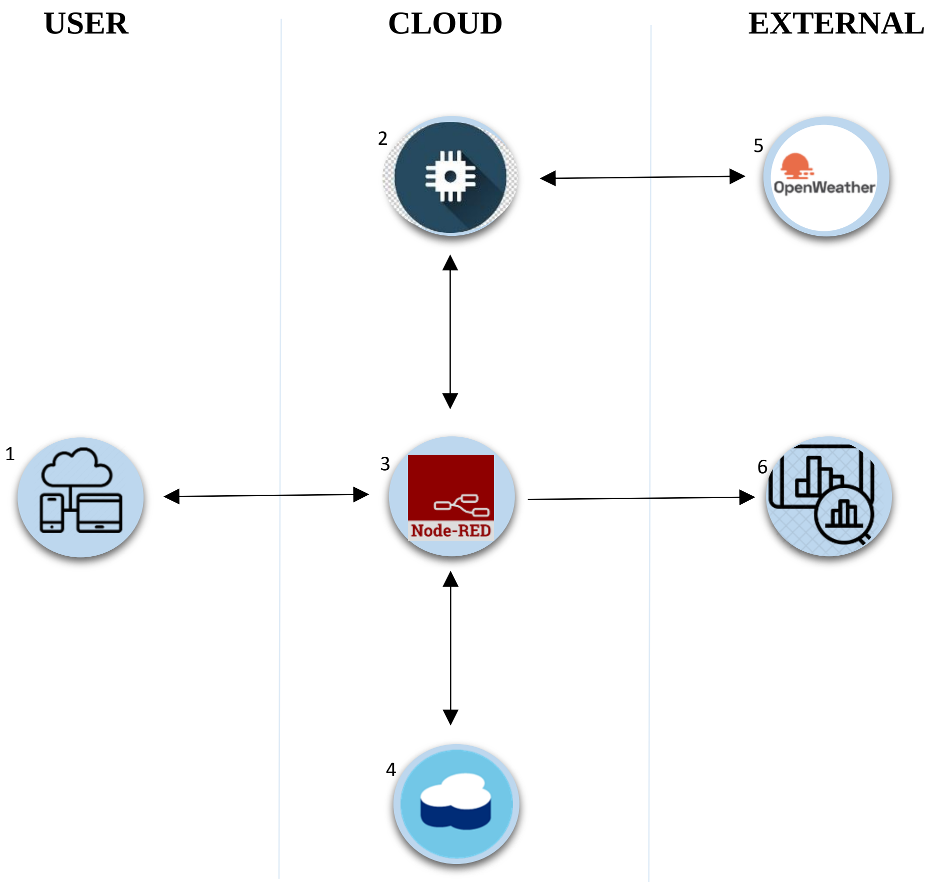
To display these real-time weather values on the user interface, text nodes are used from the dashboard palette.

4.4 PYTHON PROGRAM

A python code is written to receive commands from the IBM IoT Watson platform. The command given to control the motor can be visualised by this program. The source code is mentioned in Appendix (A).

CHAPTER 5

FLOW



1. DEVICE: Create and deploy a Node-RED instance on IBM Cloud and connect it to the Watson IoT Platform
2. WATSON IOT PLATFORM: Save device data by using the Watson IoT Platform dashboards and Cloudant.
3. NODE-RED: Build and deploy the application
4. CLOUDANT: Inject IoT data into the IoT Platform
5. OPENWEATHER API: Get real-time weather forecast
6. CHARTS&ANALYTICS: Monitor and visualize device data by using the Watson IoT Platform dashboards

CHAPTER 6

RESULT

The project has used the concept of Internet of Things and the state-of-art sensors to precisely measure soil parameters (Fig. 3.8) like temperature, humidity and soil moisture. The farmer can have insights on the current values of these variables using a mobile or web application. To identify other conditions like maximum temperature, minimum temperature and wind speed that might impact his yield, real-time weather forecasting (Fig. 3.9) is done using an external platform (Open Weather API). The farmer can also use this application for irrigating his field. The motor can be controlled using the application from anywhere (Fig. 3.10) as long as each of these devices remain connected to the network. The command received from the motor controller can be viewed in Fig. 6.1.

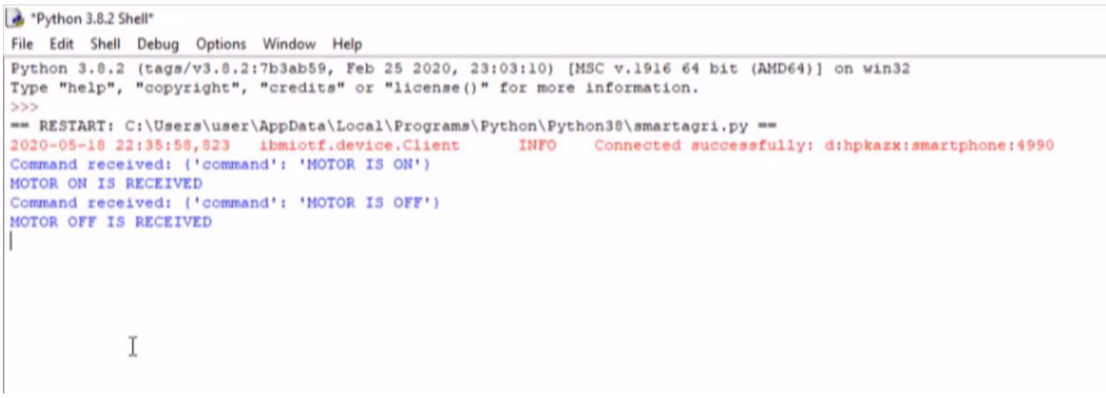


Fig 6.1 Output of python code for receiving motor control commands

CHAPTER 7
ADVANTAGES & DISADVANTAGES

7.1 ADVANTAGES

- Measure soil parameters like temperature, humidity and soil moisture
- Have statistical insights on the current values of these variables using a mobile or web application
- Identify real-time weather conditions like maximum temperature, minimum temperature and wind speed that might impact yield
- Control the motor using the application from anywhere
- User-friendly interface makes it easy for the user to understand and navigate through the application

7.2 DISADVANTAGES

- Weather forecast might be difficult to interpret by a farmer
- Application failure when the network disconnects
- Prior knowledge of the effect of soil parameters on agriculture required
- Poor farmers might not afford the cost of installing sensors

CHAPTER 8
APPLICATION

Smart farming is one of the fastest growing field in IoT. 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. IoT-based smart farming is highly efficient when compared with the conventional approach. In terms of environmental issues, IoT-based smart farming can provide great benefits including more efficient water usage, or optimization of inputs and treatments.

The application can be used to analyse meaningful insights from the data to yield better return on investment. Sensing for soil moisture and nutrients, controlling water usage for plant growth and determining custom fertilizer are some simple uses of IoT.

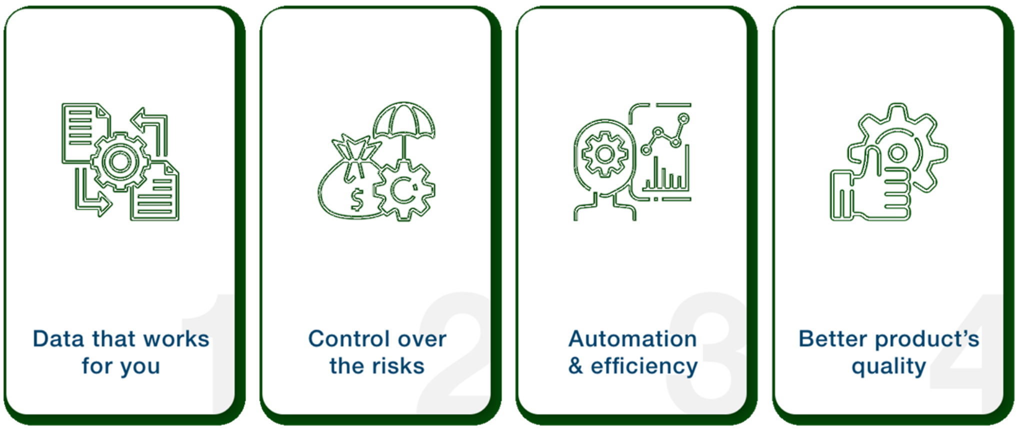


Fig 8.1 Advantages of smart agriculture

CHAPTER 9

CONCLUSION

The research in this work has examined the underlining technologies of cloud computing and Internet of Things and the possibilities of implementing a smart agriculture system using the technology. A design was built and deployed to demonstrate that a user can analyse and control the process of irrigation from a remote location with the help of a mobile or web application. While this work had demonstrated the feasibility of an IoT network, further research is recommended to test security and network performance as well as reduce the costs associated when the system is fully implemented.

CHAPTER 10
FUTURE SCOPE

It is believed that a smart network of sensors, actuators, cameras, robots, drones, and other connected devices will bring an unprecedented level of control and automated decision-making to agriculture, making possible an enduring ecosystem of innovation in this eldest of industries.

Smart Farming and IoT-driven agriculture are paving the way for what can be called a Third Green Revolution. This revolution draws upon the combined application of data-driven analytics technologies, such as precision farming equipment, IoT, “big data” analytics, Unmanned Aerial Vehicles (UAVs or drones), robotics, etc.

In the future this smart farming revolution depicts, pesticide and fertilizer use will drop while overall efficiency will rise. IoT technologies will enable better food traceability, which in turn will lead to increased food safety. It will also be beneficial for the environment, for example, more efficient use of water, or optimization of treatments and inputs.

Therefore, smart farming has a real potential to deliver a more productive and sustainable form of agricultural production, based on a more precise and resource-efficient approach.

REFERENCES

1. https://flows.nodered.org/
2. https://internetofthings.ibmcloud.com/
3. https://cloud.ibm.com/
4. https://www.iotforall.com
5. https://internetofthingsagenda.techtarget.com

APPENDIX (A): SOURCE CODE

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

#Provide your IBM Watson Device Credentials
organization = "hpkazx" #replace the ORG ID
deviceType = "smartphone"#replace the Device type wi
deviceId = "4990"#replace Device ID
authMethod = "token"
authToken = "smartagriculture4990" #Replace the authtoken

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

    elif cmd.data['command']=='MOTOR IS OFF':
        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()
```