A

PROJECT REPORT

ON

SMART AGRICULTURE SYSTEM

SUMMER INTERNSHIP PROGRAM

AT

SMART BRIDGE

BY

VAMSHI KUNDANAPALLY

SMART BRIDGE

HYDERABAD

Contents

1. Introduction

1.1 Overview

Smart Agriculture System based on IOT can monitor soil moisture and climatic conditions to grow and yield a good crop. The farmer can also get the real-time weather forecasting data by using external platforms like Open Weather API. Farmer will be provided a mobile app using which he can monitor the temperature, humidity and soil moisture parameters along with weather forecasting details. Based on all the parameter, farmer can water his crop by controlling the motor using the mobile application. Thus even if the farmer is not present near his crop he can water his crop by controlling the motors using the application from anywhere.

2. Literature Survey

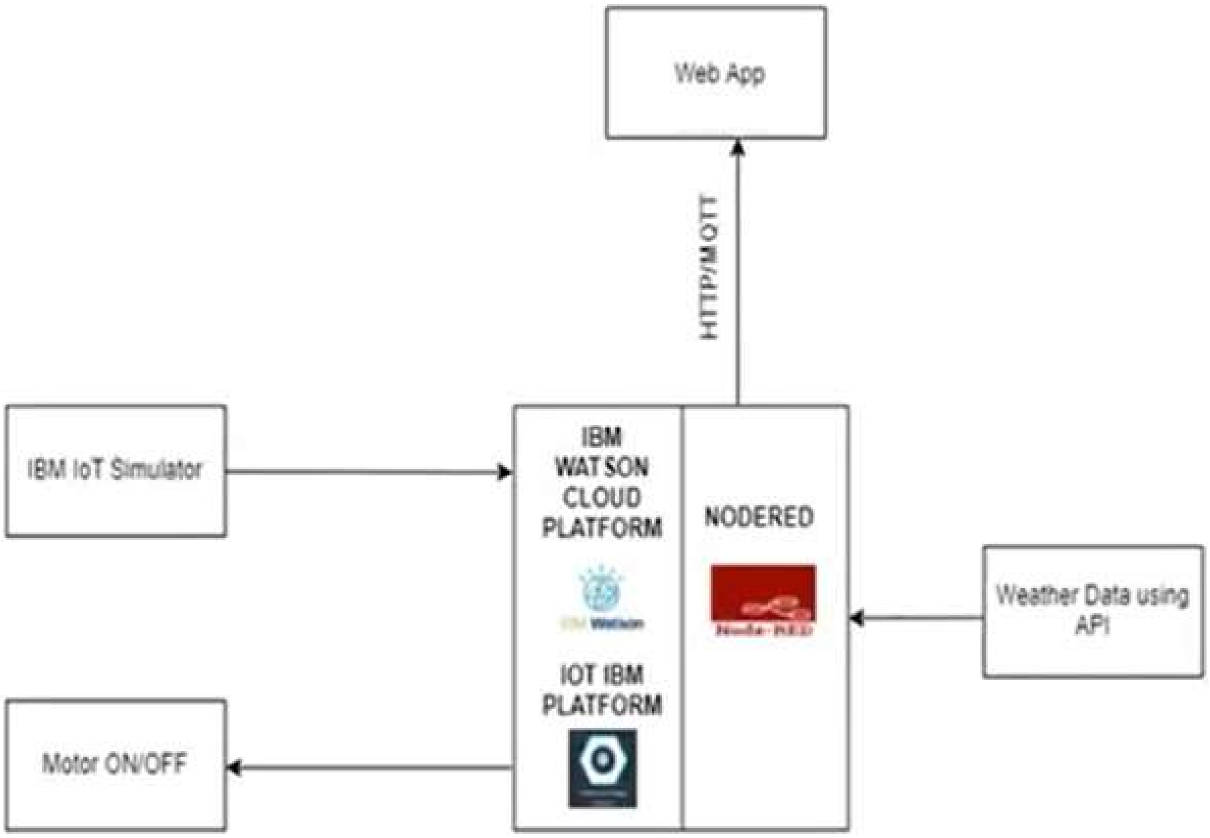
2.1Proposed Solution

IoT is based on SMART AGRICULTURE SYSTEM is regarded as the IoT gadget focusing on Live Monitoring of Environmental data in terms of Temperature, Moisture and Humidity of atmosphere and the plant/crop. The system provides the concept of “Plug and Sense” in which farmers can directly implement smart farming by such as putting the System on the field and getting Live Data feeds on various electronic devices using Web Application. Moreover, the data generated via sensors can be easily shared and viewed by agriculture consultants anywhere remotely via Cloud Computing technology integration. The system allows manually to turn the pumping motor ON and OFF on sensing the moisture content of the soil.

3. Project Description

- Smart Agriculture System based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop.
- The farmer can also get the realtime weather forecasting data by using external platforms like Open Weather API.
- Farmer is provided a mobile app using which he can monitor the temperature,humidity and soil moisture parameters along with weather forecasting details.
- Based on all the parameters he can water his crop by controlling the motors using the mobile application.
- Even if the farmer is not present near his crop he can water his crop by controlling the motors using the mobile application from anywhere.
- Here we are using the Online IoT simulator for getting the Temperature,Humidity and Soil Moisture values.

3.1 **Block Diagram :**



4. Project Scope

4.1 Project Summary

- Smart Agriculture System based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop.
- A Farmer is provided a mobile app using which he can monitor the temperatur e,humidity and soil moisture parameters along with weather forecasting detail s.
- Based on all the parameters he can water his crop by controlling the motors using the mobile application

4.2Project Requirements

- IOT Application Development
- IOT Cloud Platform

4.3 Software Requirements

- GIT tool and Nodered
- Python IDE

4.4 Project Deliverables

- Explore IBM Cloud Platform
- Connect The IOT Simulator To Watson IOT Platform
- Configure The Nodered To Get The Data From IBM IOT Platform And Open Weather API
- Building A Web App
- Configure Device To Receive The Data From The Web Application And Control Motors

5.Node-Red Flow

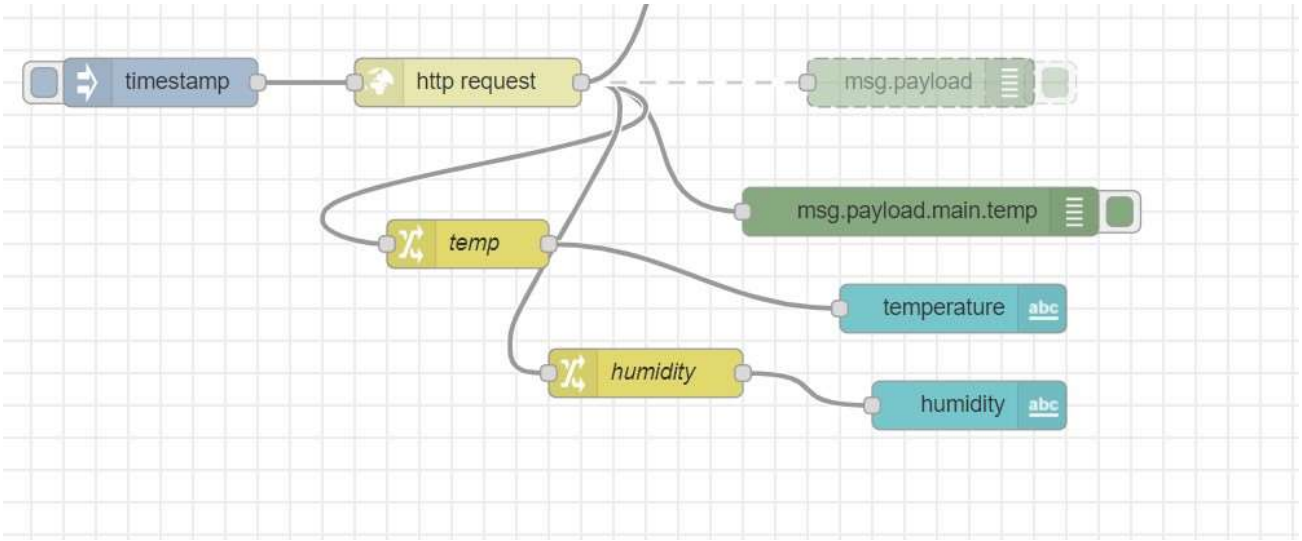
Now once we got the data in the cloud, we will use node red to get the data in a web app. To install node-red in windows follow this link. After this we would need to externally install IBM iot node in node red using the below code.

Node-red-contrib-scx-

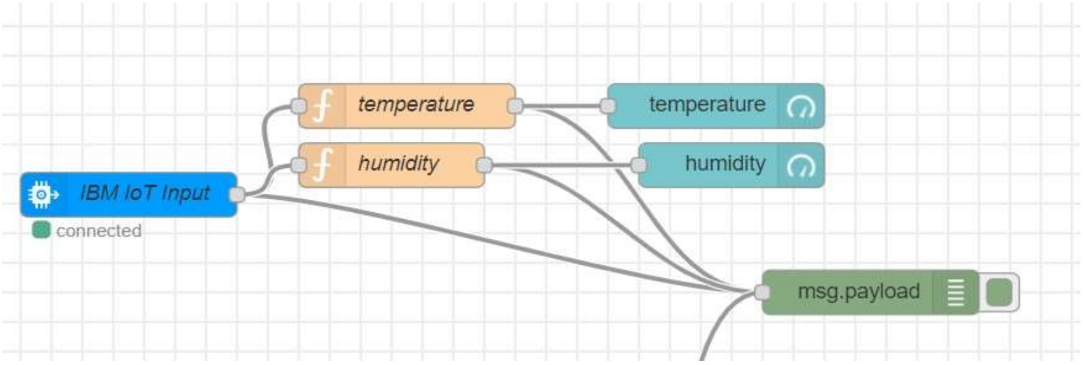
ibmiotapp We would need 3 flows:

1. To take the weather data from OpenWeather API.
2. To take sensor data from the IBM cloud.
3. Finally, to transfer the motor control data to the cloud.

Flow1

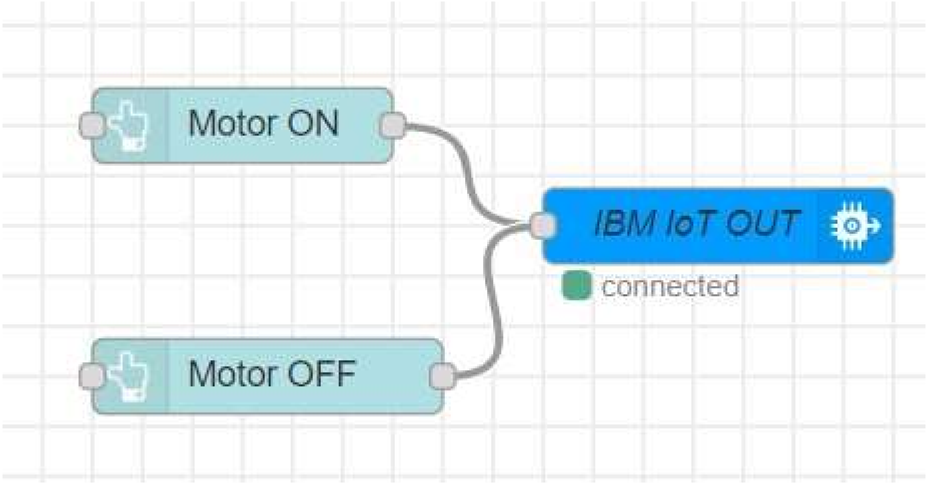


Flow-2

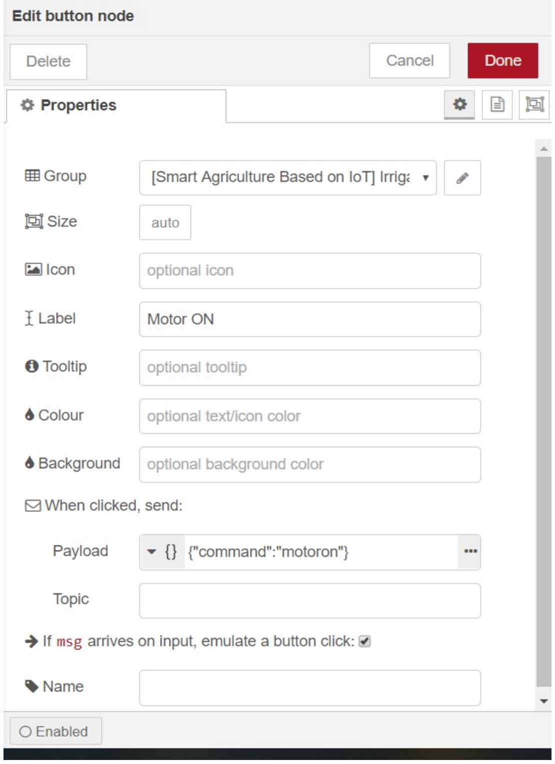


Now we need to put API in IBM IOT node which we can get form IBM cloud from Apps tab.

Flow 3



Settings for “MOTOR” and Buttons



Group: [Smart Agriculture Based on IoT] Irriga

Size: auto

Icon: optional icon

Label: Motor ON

Tooltip: optional tooltip

Colour: optional text/icon color

Background: optional background color

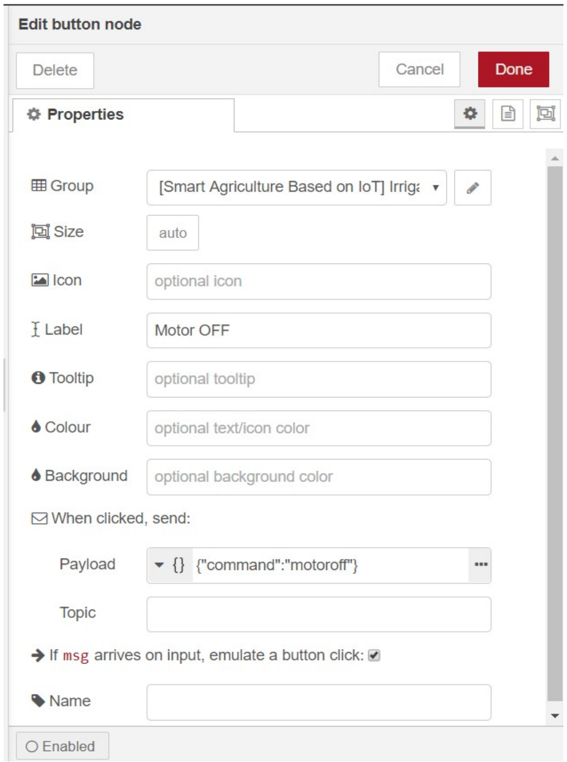
When clicked, send:

Payload: {} {"command":"motoron"}

Topic:

If msg arrives on input, emulate a button click: ☑

Name:



Group: [Smart Agriculture Based on IoT] Irriga

Size: auto

Icon: optional icon

Label: Motor OFF

Tooltip: optional tooltip

Colour: optional text/icon color

Background: optional background color

When clicked, send:

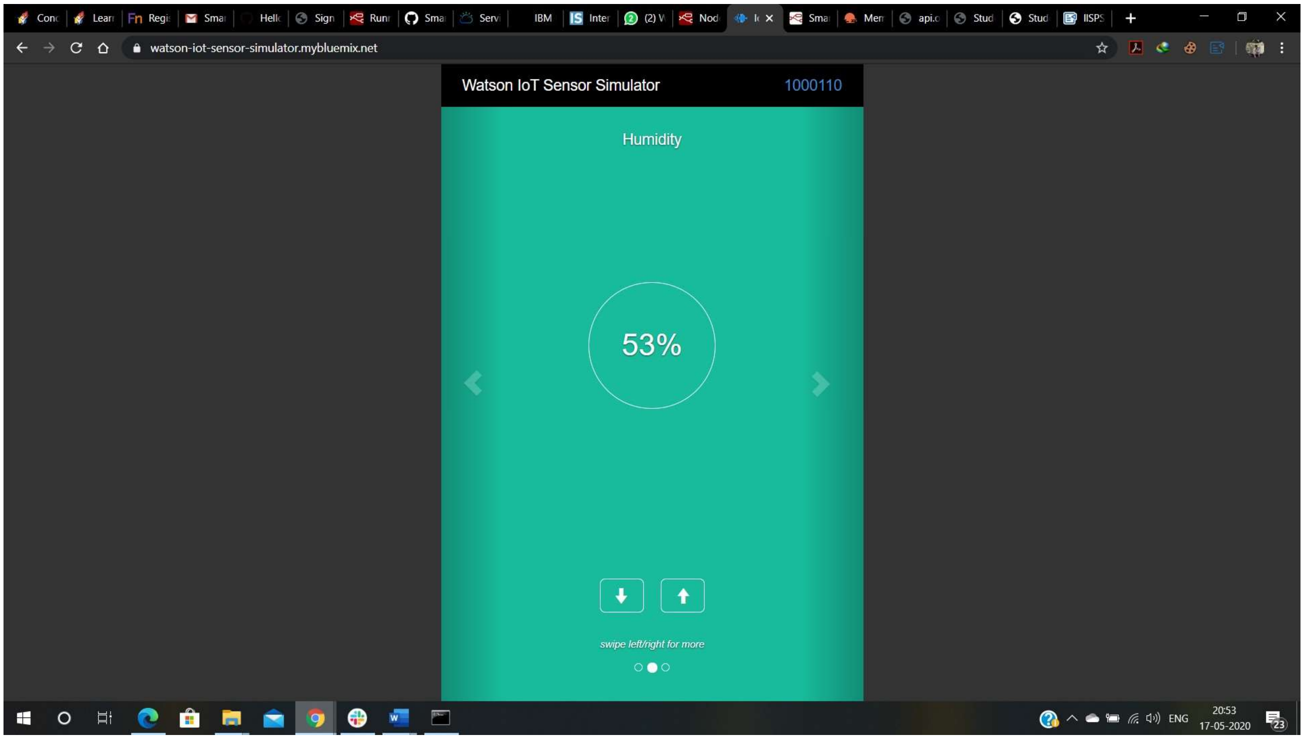
Payload: {} {"command":"motoroff"}

Topic:

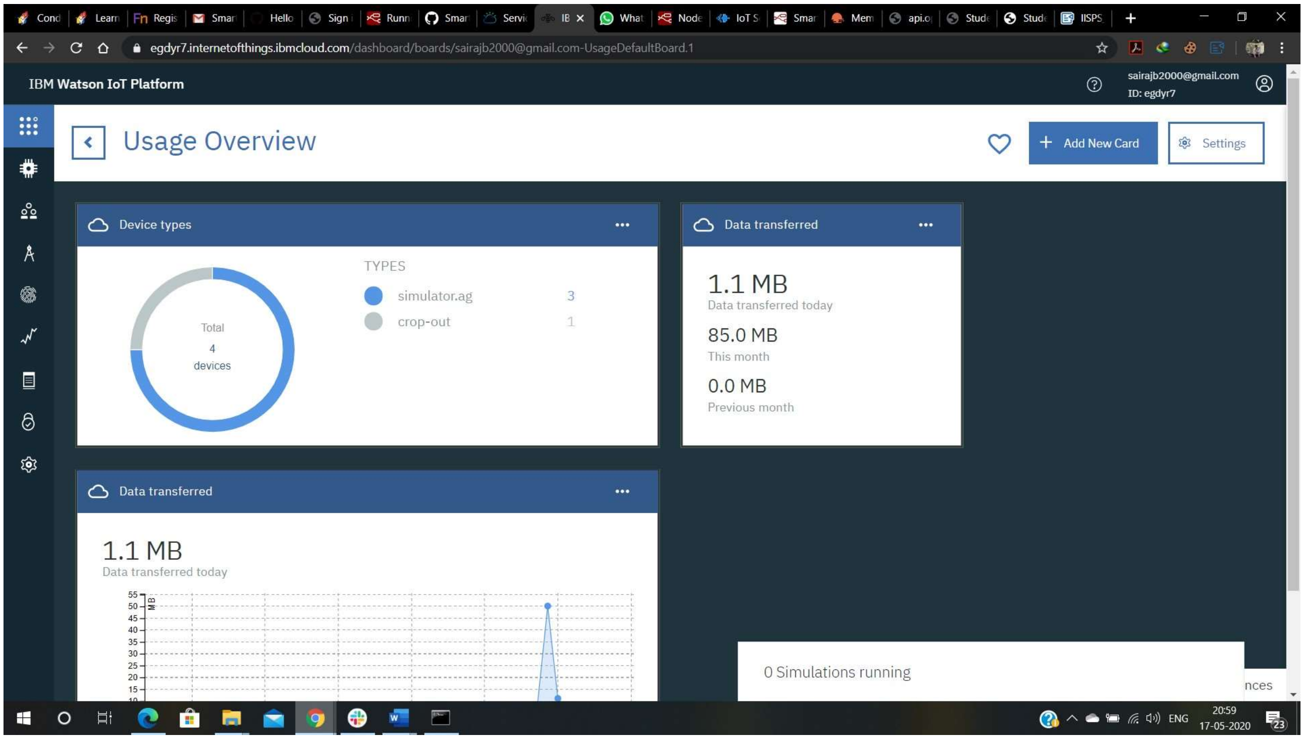
If msg arrives on input, emulate a button click: ☑

Name:

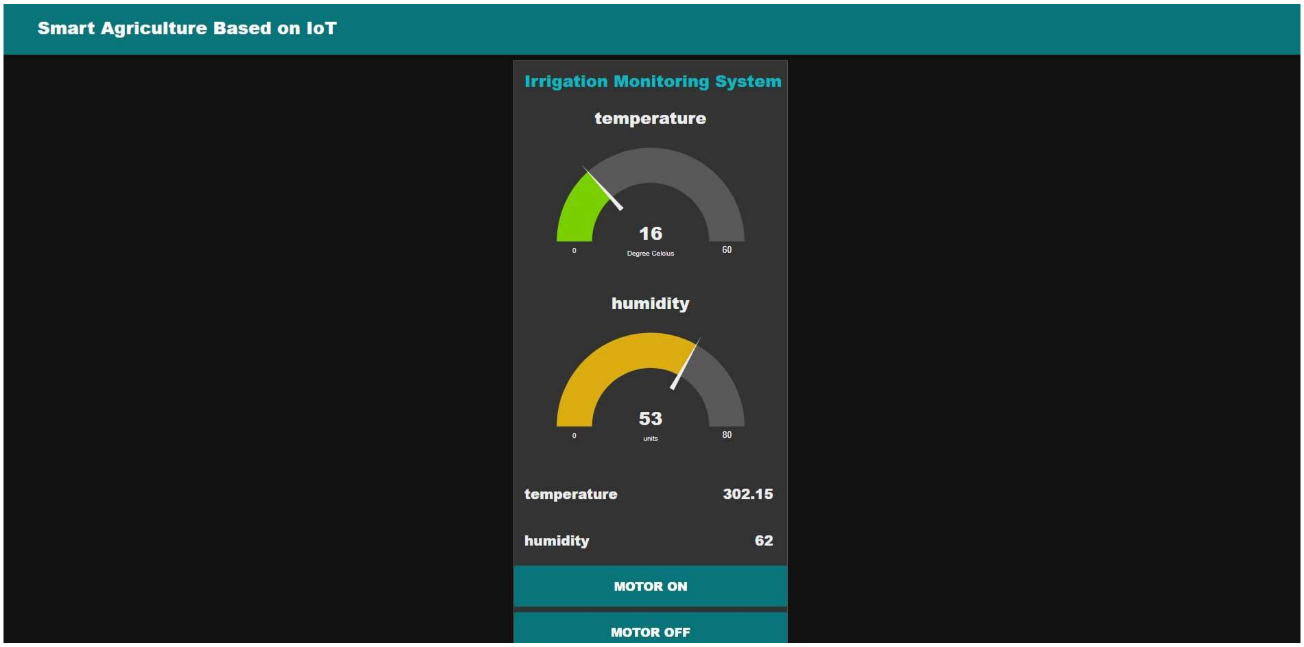
6. Watson IOT Sensor


watson-iot-sensor-simulator.mybluemix.net
Watson IoT Sensor Simulator
1000110
Humidity
53%
swipe left/right for more
20:53
17-05-2020

1. Now in the cloud we can create cards to view the simulator data.



7. Node-Red UI Dashboard



Smart Agriculture Based on IoT

Irrigation Monitoring System

temperature 302.15

humidity 62

8. Advantages and Disadvantages

8.1 Advantages

- Small sized customer support team is enough as only a low volume of queries are redirected.
- Cost efficient.
- Only a small amount of queries need representatives, hence the customer is al so satisfied with quick and easy solution.

8.2 Disadvantages

- The IBM Watson Discovery service returns wrong results if not properly configured.
- Same answers might be given for different queries.

9. Applications

- Using the IoT concept in the agriculture field will help farmers not only reduce waste but also increase in yield production varying from the quantity of fertilizer utilized to the quality of the production achieved.
- These days IoT has also been implemented in these following practices.
    1. Crop Monitoring: Using IoT technique we can monitor the quality of crop.
    2. Precision Farming: Precision farming is a farming practice that is more accurate and controlled. It deals with production of crop along with raising livestock.
    3. Livestock Monitoring: With the help of sensor, health of the livestock can be monitored which will directly help in the yield production of good produced from them.
    4. Agricultural drones: It is a good example for farming and in order to improve the various agricultural practices.

10.Conclusion

By following the above-mentioned steps, we create a basic chatbot which can help us to answer the basic questions of the customer or user related to location of the office, working hours and the information about the product. We have successfully created the intelligent help desk smart chatbot using IBM Watson Assistant, IBM Cloud Function, IBM Watson Discovery and Node-Red.

11. Future Scope

In the future, the bot can be improvised with live UI. Also, features like similar information, helpline contacts, Watson text to audio, extended user requests, a database to store and display recent chats, etc.

12.Source Code

Python Code

```
import time
import sys
import ibmiotf.application
import ibmiotf.device

organization = "egdyr7" #replace the ORG ID deviceType = "crop-out"#replace the Device
type wi deviceId = "1000111"#replace Device ID authMethod = "token"
authToken = "vamshi123" #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()
```