Project name: **Smart Agriculture system based on IoT - SB13171.**

Introduction

1.1 Overview:

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 by controlling the motors over the internet.

Smart Agriculture System based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop.

In this system, the farmer will get the real-time weather forecasting data by using external platforms like Open Weather API.

Farmer is provided a web app using which he can monitor the temperature, humidity and soil moisture parameters along with weather forecasting details. Considering all the parameters he/she can water the crop by controlling the motors using the mobile application.

1.2 Purpose:

The purpose of the system is providing essential information like soil humidity, ambient temperature, ambient humidity, soil temperature and weather to farmers and allow them to control the motors installed for irrigation so as to meet the crop water demand. This farm monitoring can also be done via web application on smart phones for convenience use and time saving.

The system continuously measures the soil conditions and updates the farmer. This system also provides weather forecast of the location so that the farmer can make an informed decision.

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.

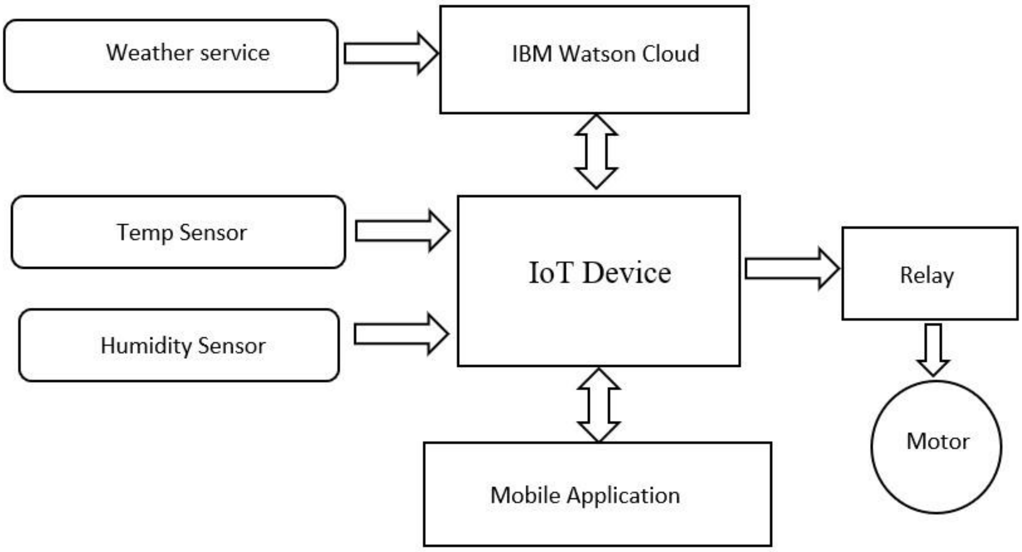
It has also been proposed that by constantly measuring soil parameters like soil type, nutrient presence, flow of irrigation, pest resistance, etc. we can prepare a report, based on that report a farmer may crop seeds which are suitable to that land. Furthermore, the use of unmanned aerial vehicles for 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.

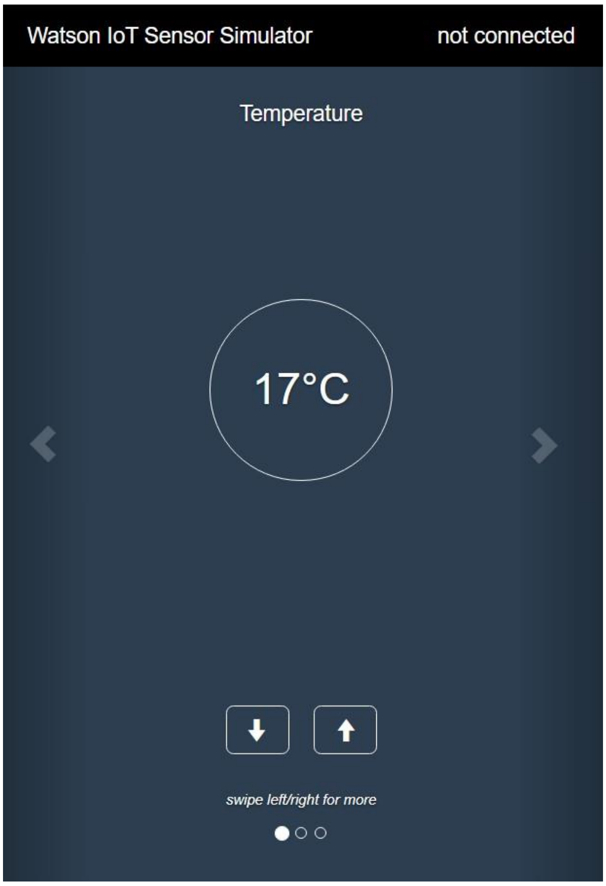


3.2 Hardware/Software Designing:

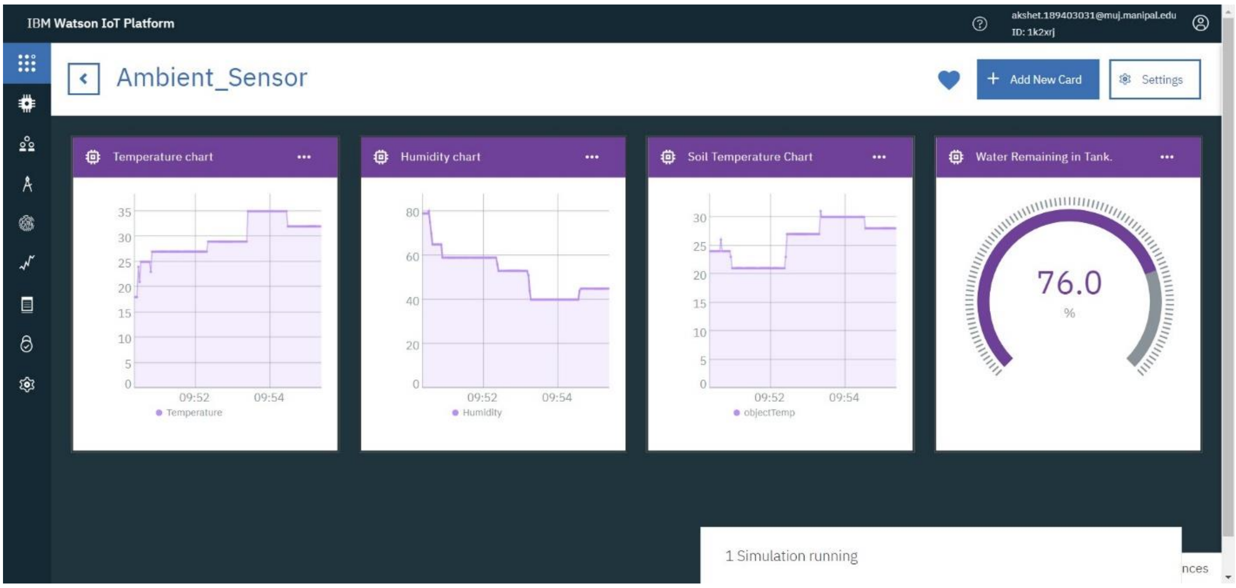
The hardware part consists of very basic components. The temperature and humidity sensors viz. a DHT11, are connected to the analog input pins of IoT device like NodeMcu. In case of RaspberryPi we need to use a Digital to Analog converter to get values from analog sensor. The digital out pin of IoT device 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:



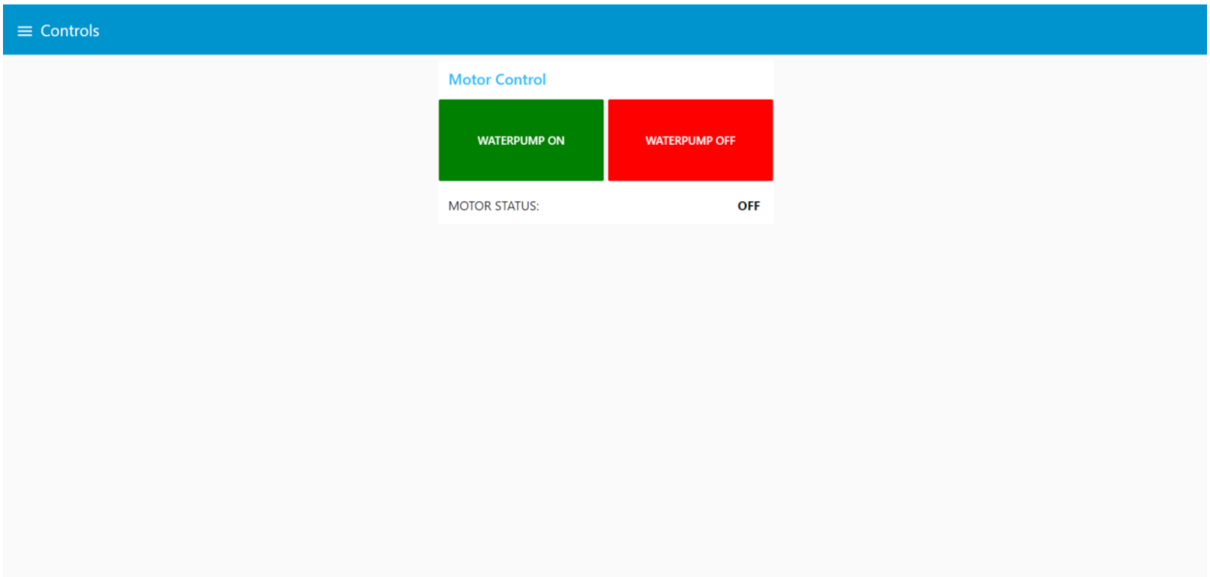
IBM Sensor.



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 IBMout and button nodes in node-red once all these flows are deployed the system starts showing weather forecast and sensor data.

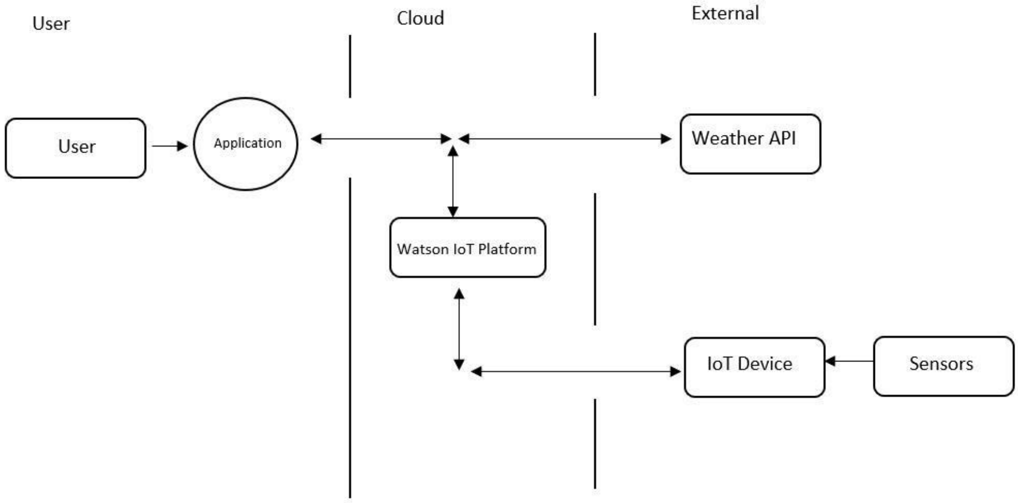
Dashboard.


Controls
Motor Control
WATERPUMP ON
WATERPUMP OFF
MOTOR STATUS:
OFF

Controls.

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.

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.

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.

9. Conclusion:

By using this system farmers can effectively produce more yield and can save water from wastage. With help of weather forecast service farmer can water their land as per weather. He can also turn off motor when water content in soil is sufficient.

10. Future Scope:

With help of artificial intelligence and Machine Learning algorithms, we can suggest farmers to grow a particular crop based on soil data from the sensors. We can also control the water supply to crops with help of artificial learning based on soil moisture.

11. Bibliography:

A. Appendix:

Important websites:

1. IBM Watson IoT Platform.
2. IoT Sensor.
3. https://openweathermap.org/api.

Source Code:

```
import sys
import ibmiotf.device

organization = " "
deviceType = " "
deviceId = " "
authMethod = "token"
authToken = " "


def commandHandler(cmd):
    print("Command received: %s" % cmd.data)

    for key in cmd.data.keys():
        if key == 'motor':
            if cmd.data['motor'] == 'ON':
                print("MOTOR is turned ON")

            elif cmd.data['motor'] == 'OFF':
                print("MOTOR is turned OFF")


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 = commandHandler
```