

# **Smart Agriculture System Based on IoT REPORT**

**Submitted by**  
Archana Suresh

# **CONTENT**

## **1 INTRODUCTION**

1.1 Overview

1.2 Purpose

## **2 LITERATURE SURVEY**

2.1 Existing problem

2.2 Proposed solution

## **3 THEORITICAL ANALYSIS**

3.1 Block diagram

3.2 Hardware / Software designing

## **4 FLOWCHART**

## **5 RESULT**

## **6 ADVANTAGES & DISADVANTAGES**

## **7 APPLICATIONS**

## **8 CONCLUSION**

## **9 FUTURE SCOPE**

## **10 BIBILOGRAPHY**

### **APPENDIX**

A. Source code

# **1. INTRODUCTION**

## **1.1 Overview**

In India, agriculture plays an important role for development in food production. In our country, agriculture depends on the monsoons which are not a sufficient source of water. So irrigation is used in agriculture field. But this has so many challenges. Consumption of water resources leads to depletion of water level. This rises the need for an efficient irrigation system as well as a monitoring system. Internet of Things (IoT) is a milestone in the evolution of technology. IOT plays an important role in many fields, one of that is Agriculture by which it can feed billions of people on Earth in future.

## **1.2 Purpose**

The objective of this project is to overcome the challenges in the irrigation system of agricultural activities. The aim is to develop a system. The system can be operated from remote location through wireless transmission so there is no need to concern about irrigation timing as per crop or soil condition. The farmer is notified with the information regarding field condition through mobile periodically. This system will be more useful in areas where there is scarcity of water and will be worth efficient with satisfying its requirements. The objective of the system is to a) conserve energy & water resources b) handles the system manually and automatically.

## **2. LITERATURE SURVEY**

### **2.1 Existing problem**

Due to unplanned use of water the ground water level is decreasing day by day, lack of rains and scarcity of land water also results in decrement in volume of water on earth. Nowadays, water shortage is becoming one of the biggest problems in the world. We need water in each and every field. In our day to day life also water is essential. Agriculture is one of fields where water is required in tremendous quantity. Wastage of water is the major problem in agriculture. Every time excess of water is give to the fields. There are many techniques to save or to control wastage of water in agriculture. This brings in the dire need of a system that proposes effective usage of water, without wasting even a drop and also a system that helps a farmer in controlling his crops even if he is away.

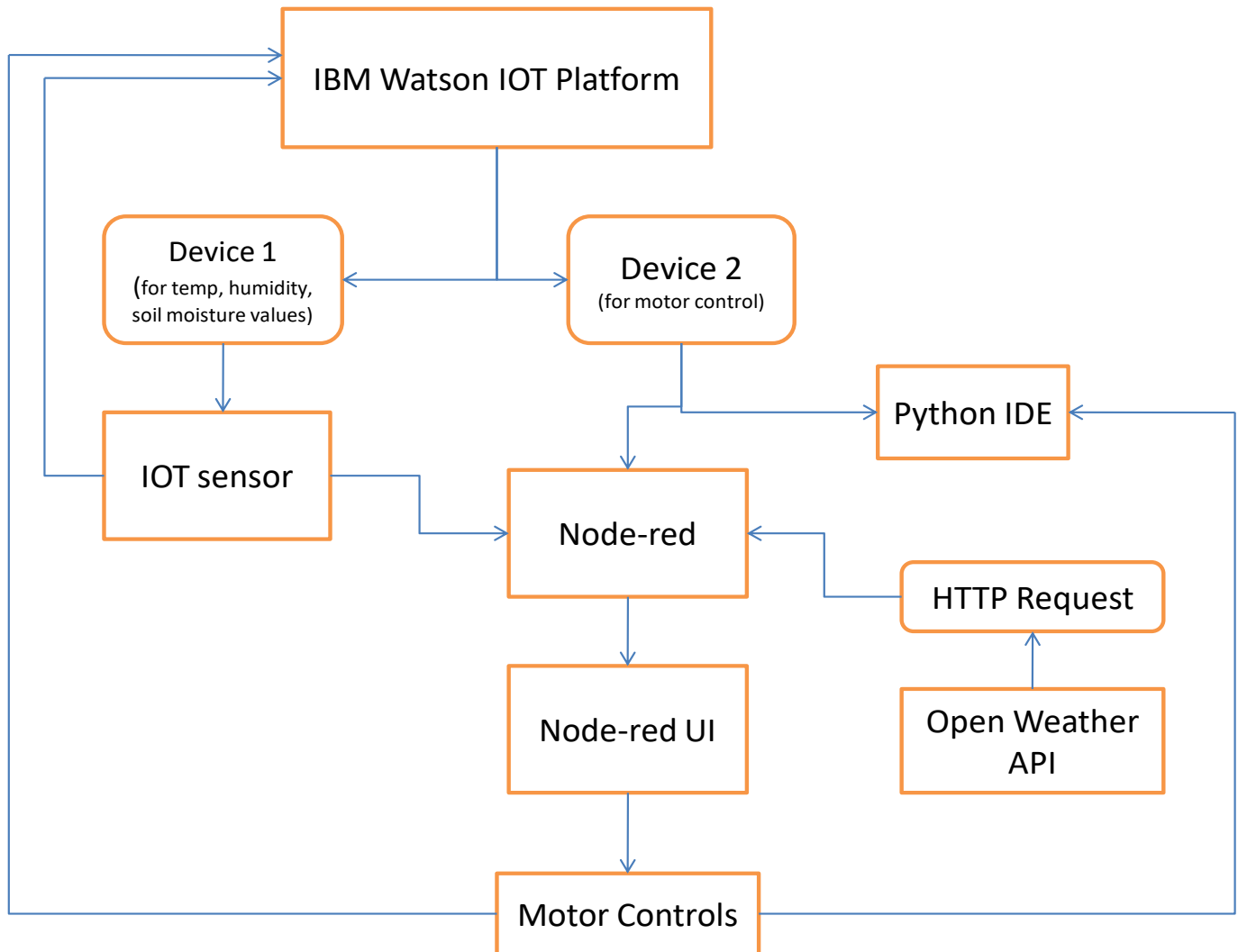
### **2.2 Proposed solution**

Smart Agriculture System based on IoT helps a farmer yeild a good crop. The objective of the system is to a) conserve energy & water resources b) handles the system manually and automatically.

The farmer can watch his crops from anywhere using the mobile application. The app provides him details about soil mosture, climatic conditions, temperature, humidity and weather forecasting and other parameters with which the farmer can decide when and how much amount of water to be irrigated to his crops. He can irrigate the crops by controlling the motor using the application

### 3. THEORETICAL ANALYSIS

#### 3.1 Block diagram

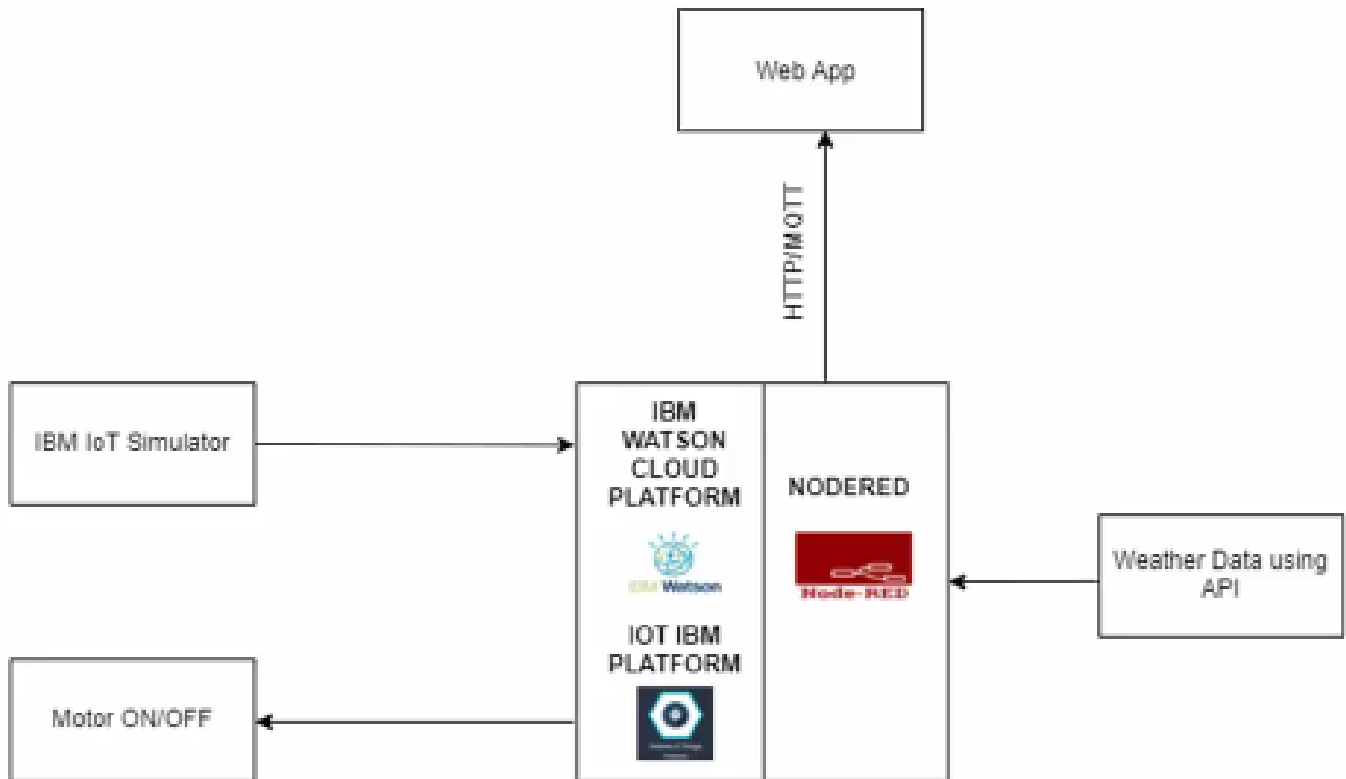


#### 3.2 Hardware/Software analysis

- IBM Watson IOT Platform

- a managed, cloud-hosted service designed to make it simple to derive value from your **IoT** devices.
- Node-red
  - a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.
- Node-red UI
- Open Weather API
  - helps you find useful information, links, documents to start using our **weather API** services smoothly
- Python IDE

## 4. FLOWCHART



## 5. RESULT

Developed a web application that displays weather conditions of a particular area (the area selected by the user) and also temperature, humidity and soil moisture of the crops. User can control the motor operations after analyzing the environmental conditions of his crops

Smart Agriculture

Smart home

MOTOR ON

MOTOR OFF

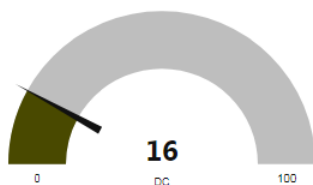
Temperature: **297.15**

Humidity: **88**

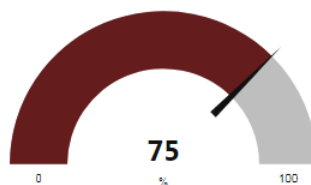
Atmospheric Pressure: **1006**

Motor Status: **Motor ON**

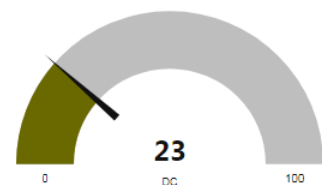
Temp Gauge



Humidity



Soil moisture





## **6. ADVANTAGES AND DISADVANTAGES**

### **Advantages:**

- conserve energy & water resources
- system can be handled manually and automatically
- system can be operated from a distance

### **Disadvantages:**

- does not give idea about the amount of water resource being used
- requires continuous internet connection to obtain weather conditions

## **7. APPLICATIONS**

The application has found its base on agricultural field but can be extended to other areas too like constructional activities. It, currently, helps in

- Displaying environmental conditions of the crops
- Controlling the motor operation
- Monitoring the crops even far away from the place

The application can be modified to meet new requirements and expectations

## **8. CONCLUSION**

The automated irrigation system implemented was found to be feasible and cost effective for optimizing water resources for agriculture production. This irrigation system allows cultivation in places with water scarcity thereby improving sustainability. The irrigation system helps the farmer by making his work smarter. As the demand for water increases, along with the need to protect aquatic habitats, water conservation practices for irrigation need to be effective and affordable. As multiple sensors are used water can be provided only to the required area of land. This system reduces the water consumption to greater extent. It needs minimal maintenance. The power consumption has been reduced very much. The crop productivity increases and the wastage of crops are very much reduced.

## **9. FUTURE WORK**

The extension work is to make user interface much simpler by just using SMS messages for notifications and to operate the switches.

Since different regions have different weather conditions, another feature that can be added to the system is displaying or providing information on which crop is suitable for which weather conditions.

## 10. BIBILOGRAPHY

1. Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra NietoGaribay, and Miguel Ángel Porta- Gándara” Automated Irrigation System Using a Wireless Sensor Network and GPRS module” , Ieee Transactions OnInstrumentation And Measurement, Vol. 63, No. 1, January 2014.

## 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 = "cv03ko" #replace the ORG ID
deviceType = "simple_device"#replace the Device type wi
deviceId = "Motor_Control"#replace Device ID
authMethod = "token"
authToken = "123456789" #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()
```