Green House Monitoring & Control System

Surya Prakash S (suryas190299@gmail.com)

Student, Dept. of ECE, MIT Manipal, India

## *Abstract*— conventionally, crops are chosen based on previous history of climatic conditions. This has become an inefficient decision making system given the current issue of climate change. The modern scenario demands accurate values of temperature, humidity, soil moisture and other relevant data in real time to maintain a robust and efficient green house. IoT allows us to implement such systems where the efficiency of the Green House is maximized to such an extent that it can be done with little supervision due to the benefits of automation.

**Index Terms— IBM, Node-Red, Watson IoT Sensor Simulator, IBM cloud platform, Python, MIT app inventor.**

**1. INTRODUCTION**

**Overview:** In recent times, huge scientific advancement has been implemented in   
various agricultural fields for the betterment   
of the future. Despite various drastic steps,  
proper assessment and productivity cannot be   
reached without removal of human error. IoT   
is an appropriate solution for this issue. Automation of motor controls and data collection from sensors would result in a more   
efficient and reliable system.

**Purpose:** The purpose is to setup up a system that allows remote monitoring of the green house. The system is setup to be as self-sufficient as possible with standards and conditions set pre-emptively to help the decision making process.

The purpose is to ensure that crop wastage is reduced and that resource wastage is eliminated. Allowing the data computation to  
be done on real time data allows this objective to become reality. The direct benefiters to this   
will be the farmers and the global impact will   
also be of significance as wastage of   
resources and man power is reduced.

**2. LITERATURE SURVEY**

## Existing Problems:

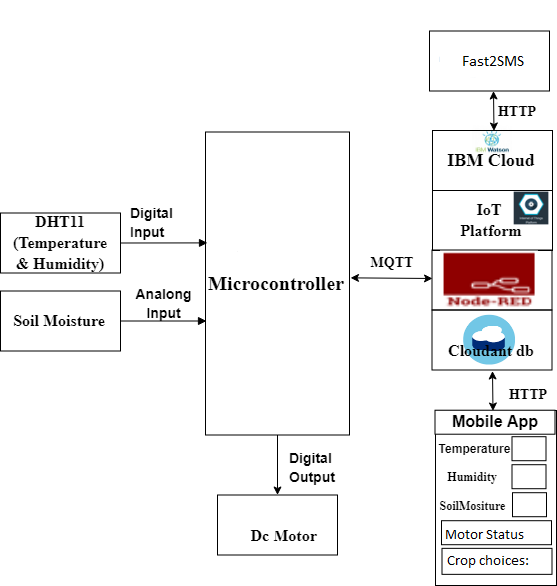
The current roadblock faced by farmers ishe lack of immediate and accurate data of the climatic conditions so that they can take the required action based on that. Such data is of significance even while choosing the crop to be planted. Another issue is the lack of man power in huge production cases where a single person cannot control all devices like the water sprinklers. Third issue is that remote monitoring of the crops and the conditions is not possible thru the conventional way of Greenhouse establishment.

## Proposed Solution:

The Internet of things (IOT) is revamping the  
situation by developing feasible and impactful resources to go up against challenges in the field. In this project we implement a system where one can monitor different environmental parameters effectively using sensor devices such as temperature sensor, relative humidity sensor and soil moisture sensor. These data are then stored online using cloud computing and the data is processed and certain actions are enforced based on the values read. The mobile app lets the user to monitor the sensor data. Also, the motor (sprinkler) is automated. The user chooses the crop thru the mobile app. The crop chosen and the current soil moisture determine whether the motor is ON/OFF for the said crop. The status of the motor is also shown on the app. Along with this, the threshold levels exist for temperature and humidity. Once the threshold is exceeded, a text message/alert is sent to the pre-determined mobile number(s). This ensures that necessary actions are taken immediately.

**3. THEORETICAL ANALYSIS**

**Block Diagram:**

 Figure 1. System Block Diagram

**Software Design:**

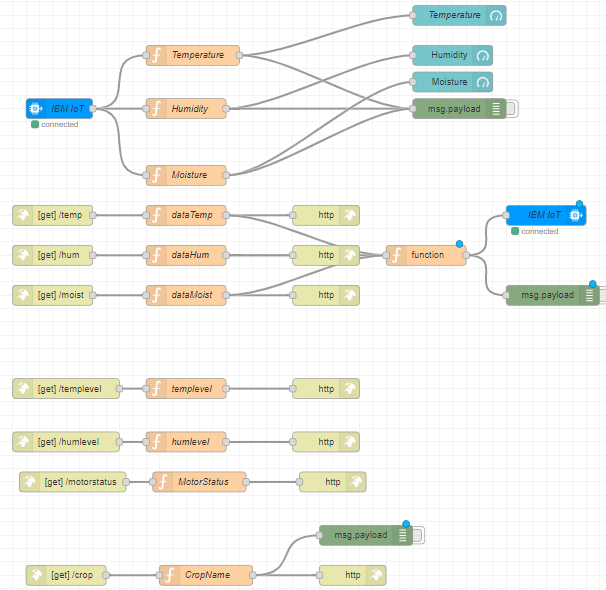
**Watson IoT Platform**

Device has been created on Watson IoT platform so that the sensor data measured (thru the Watson IoT sensor simulator) is sent to this device. The data measured are Temperature, Humidity and Soil moisture.

## NodeRed

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. 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. Device is connected to the nodered flow where the decisions regarding the motor control are made. The crop chosen is sent to the NodeRed flow as a HTTP input. The NodeRed UI is also utilised to display the real time sensor readings.

*NodeRed Flow:*

Figure 2. NodeRed flow

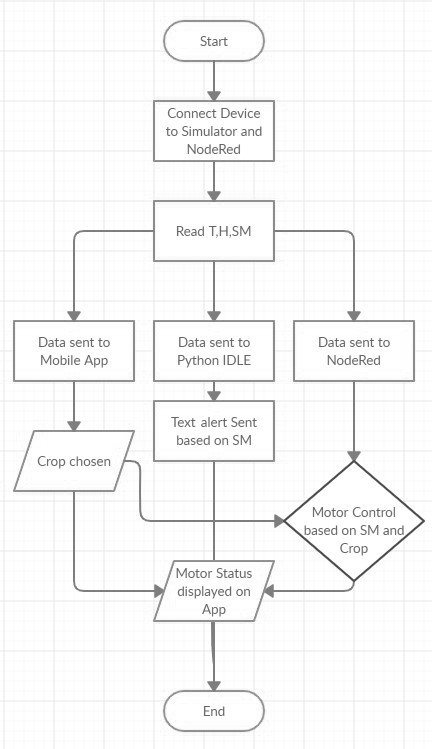
## Mobile App

## The mobile app is designed using MIT app inventor. The crop is chosen thru the Mobile app and is sent to the NodeRed flow as a HTTP in so that the motor control can be carried out accordingly. The app displays sensor readings and the current status of the motor. It also displays significant info as to whether the temperature and humidity is within the threshold values.

**Python**

The python code is to display the sensor data to the user in real time. It also monitors the temperature and the humidity so that the threshold value is not exceeded. In case either the temperature or humidity reaches above the threshold, a text message alert is sent utilizing the Fast2SMS API to the mobile number edited into the python code. This ensures quick notification and ensures immediate actions.

**4. FLOWCHART**

Figure 3. Flowchart

**5. RESULTS**

The objective has been met, and the green house monitoring and control system has been designed and implemented. The following contain the results of the implemented modules. The sensor readings are diplayed on the NodeRed UI, mobile app and as python Code output. Motor status is displayed on the mobile app. Text alerts are sent to the designated mobile phone.

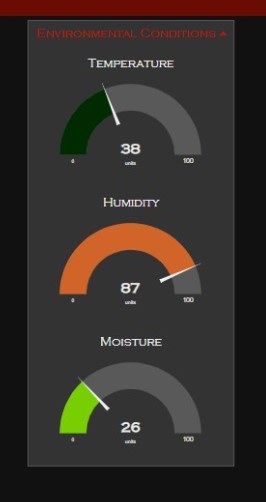


Figure 4. NodeRed UI

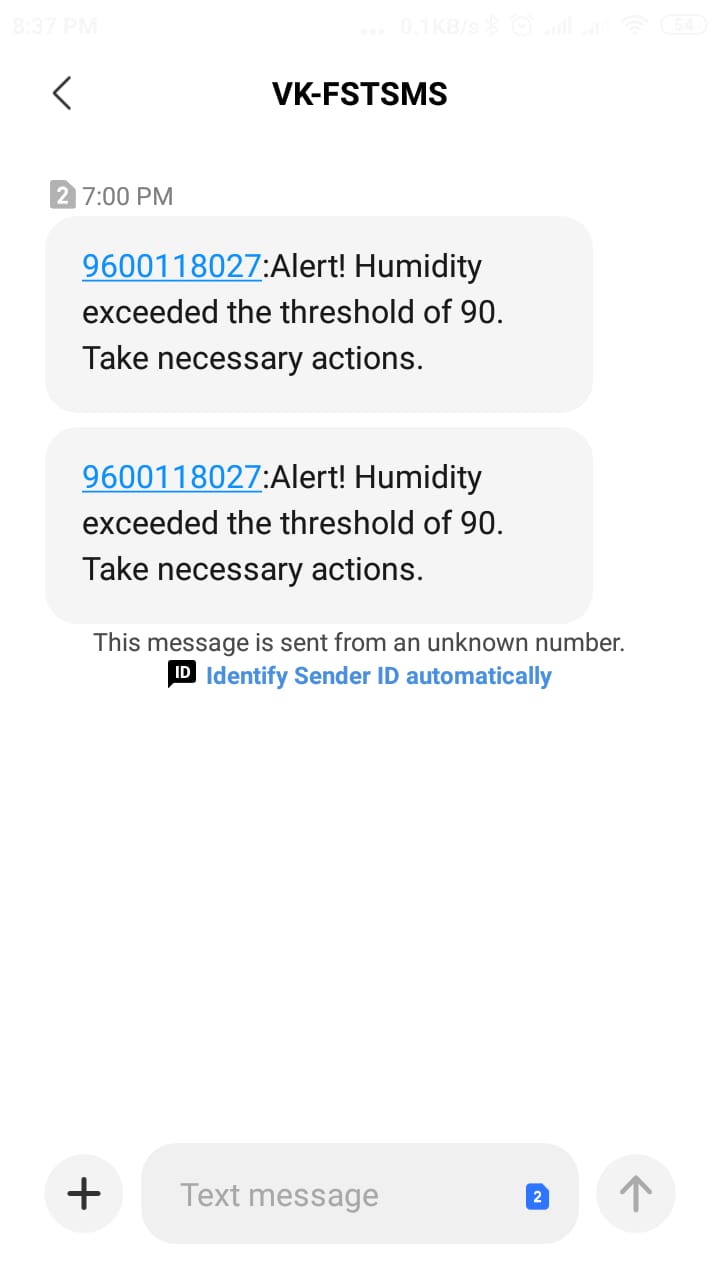


Figure 5. Text alert

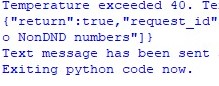
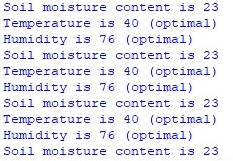


Figure 6. Python Code output

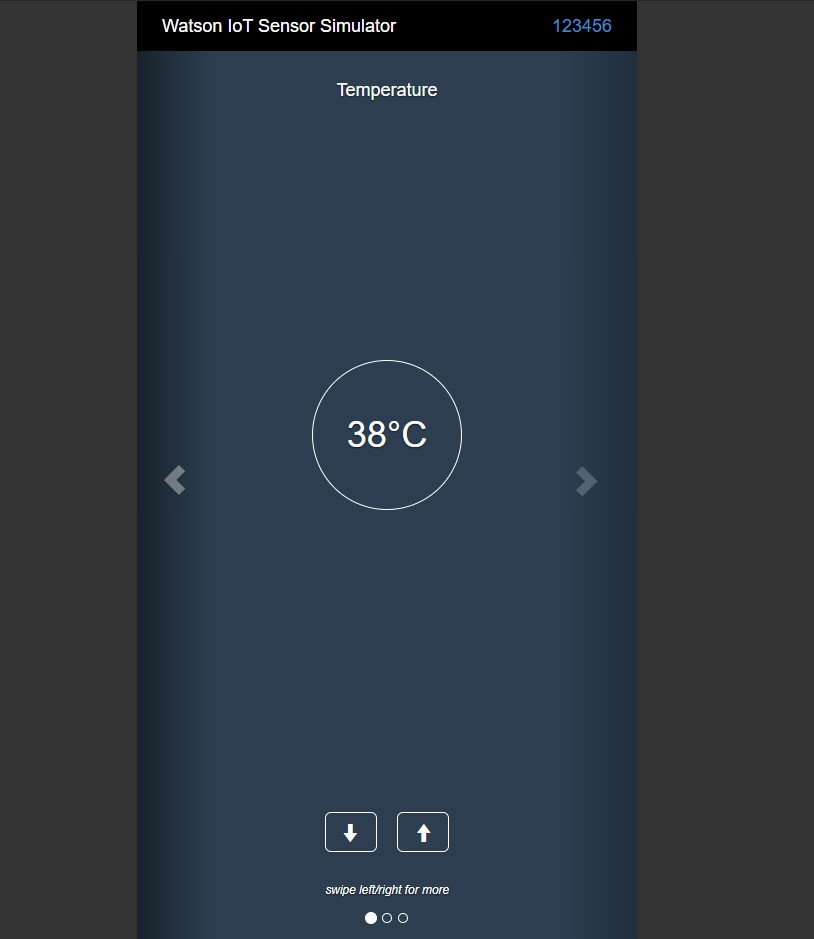


Figure 7. Watson IoT sensor simulator

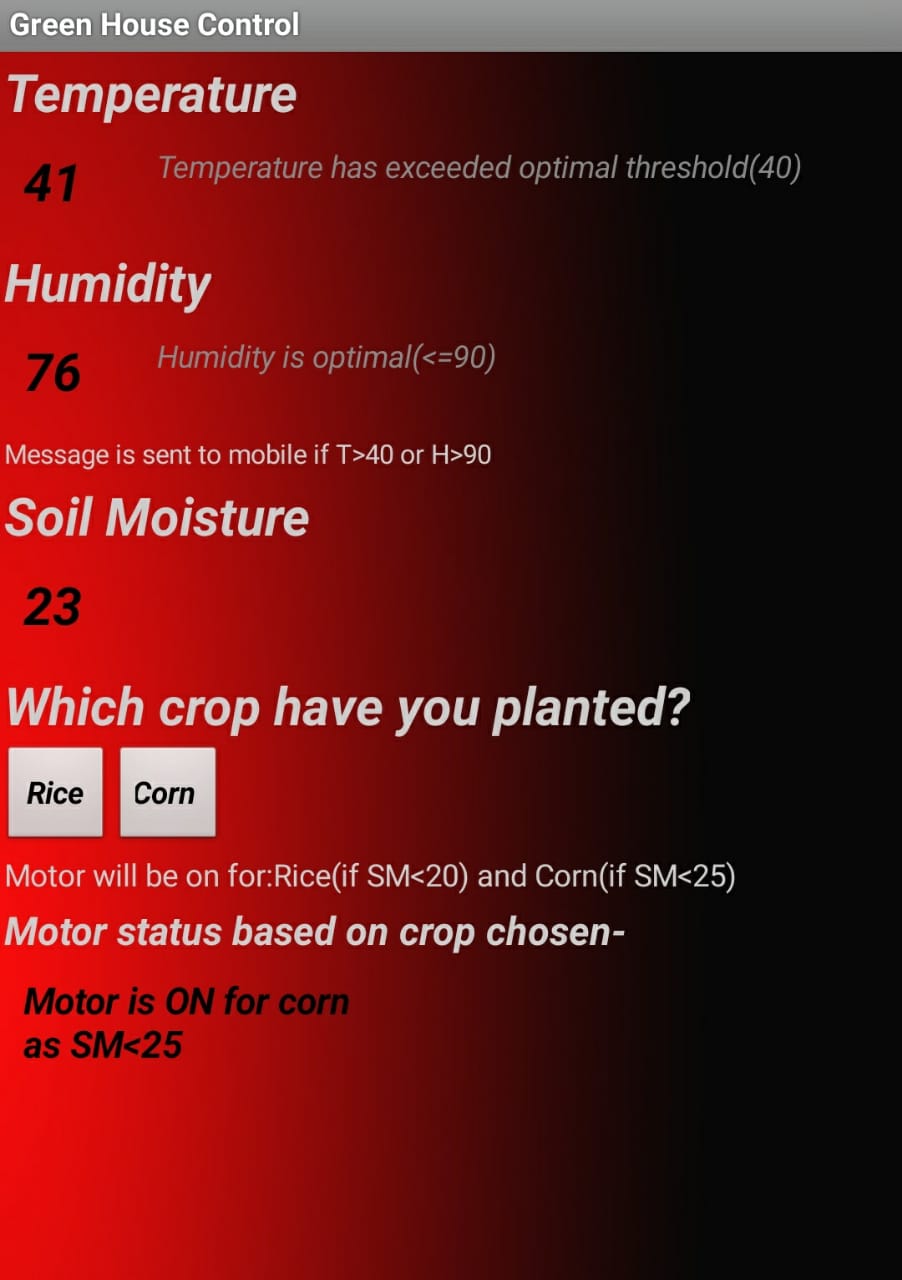


    Figure 8. Mobile app

## 6. ADVANTAGES AND DISADVANTAGES

The following are the benefits of adopting new technology - Internet of Things:

Climate Condition:

IoT solution enables us to know the real-time weather conditions. Sensors are placed inside and outside of the agriculture fields. They collect data from the environment which is used to choose the right crops which can grow and sustain in the particular climatic conditions.

Precision Farming:

The goal of precision farming is to analyse the data, generated via sensors, to react accordingly. Precision Farming helps farmers to generate data with the help of sensors and analyse that information to take intelligent and quick decisions. With the help of Precision farming, you can analyse soil conditions and other related parameters to increase the operational efficiency.

Smart Greenhouse:

To make our greenhouses smart, IoT has enabled to automatically adjust the climate conditions according to a particular set of instructions. Adoption of IoT in Greenhouses has eliminated the human intervention, thus making entire process cost- effective and increasing accuracy at the same time.

Data Analytics:

Cloud based data storage and an end-to-end IoT Platform plays an important role in the smart agriculture system. These systems are estimated to play an important role such that better activities can be performed. In the IoT world, sensors are the primary source of collecting data on a large scale. The data is analysed and transformed to meaningful information using analytics tools. The data analytics helps in the analysis of weather conditions, livestock conditions, and crop conditions and for predicting the course for the same.  
  
One major disadvantage with this is the delay in technological advancement and the initial costs to setup. Another disadvantage is the dependency of the robustness of the sensors used. Flawed data readings could cause massive financial loss. Automation might also cause unemployment.

**7. APPLICATION****S**

By implementing the latest sensing and IoT technologies in agriculture practices, every aspect of traditional farming methods can be fundamentally changed. Currently, seamless integration of wireless sensors and the IoT in smart agriculture can raise agriculture to levels which were previously unimaginable. By following the practices of smart agriculture, IoT can help to improve the solutions of many traditional farming issues, like drought response, yield optimization, land suitability, irrigation, and pest control.

**8. CONCLUSION**

IoT based Green House Monitoring & Control System for Live Monitoring of Temperature, humidity and Soil Moisture and to   
self-control motor and light remotely has   
been proposed and realised using   
NodeRed and IBM Cloud Platform. The System has high efficiency and accuracy in fetching the live sensor readings. The IoT based Greenhouse monitoring and control system being proposed via this project will assist farmers in increasing the agriculture yield and take efficient care of foodproduction as the System will always provide helping hand to farmers for getting accurate live feed of environmental temperature and soil moisture. The mobild app provides remote monitoring of the greenhouse and the text message based alert system for critical scenarios has also been realised and implemented.

**9. FUTURE SCOPE**

Apart from greenhouses, this project could be easily implemented for all agricultural activities by simply increasing the number of sensors. The application could be improved to add soil fertility content detections so that the right crop is planted for the right soil. Object detection could be added to detect the presence of weeds and other unwanted plants to ensure prompt removal of the same.

**9. BIBLIOGRAPHY**

1. https://docs.fast2sms.com/?python#authorizat in-key

2.https://cloud.ibm.com/docs/Cloudant?topic=Cloudant-getting-started-with-cloudant  
3.https://cloud.ibm.com/docs/overview?topic=overview-whatis-platform

**10. APPENDIX**

**Source Code:**

import time

import sys

import ibmiotf.application

import ibmiotf.device

import requests

#Edit contact number to the number you want the message alert sent to.

contactnumber="9841124033"

#Provide your IBM Watson Device Credentials

organization = "zsilpn"

deviceType = "raspberrypi"

deviceId = "123456"

authMethod = "token"

authToken = "12345678"

url = "https://www.fast2sms.com/dev/bulk"

# Initialize GPIO

def myCommandCallback(cmd):

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

#print(type(cmd.data))

i=(cmd.data["T"])

j=(cmd.data["H"])

k=(cmd.data["M"])

if(i<=40):

print("Temperature is",i,"(optimal)")

else:

print("Temperature exceeded 40. Text alert is being sent to mobile.")

querystring = {"authorization":"d6tkSx1N5vyGmYjz7uIALZ90MRprehPfJsboTCqlFXQcBEWHg89KBj7LJrmGZRdHYxSavbA164zPcniq","sender\_id":"FSTSMS","message":"Alert! Temperature exceeded the threshold of 40. Take necessary actions.","language":"english","route":"p","numbers":contactnumber}

headers = {

'cache-control': "no-cache"

}

response = requests.request("GET", url, headers=headers, params=querystring)

print(response.text)

print("Text message has been sent about Temperature reaching above threshold value.\nExiting python code now.")

exit(0)

if (j<=90):

print("Humidity is",j,"(optimal)")

else:

print("Humidity exceeded 90. Text alert is being sent to mobile.")

querystring = {"authorization":"d6tkSx1N5vyGmYjz7uIALZ90MRprehPfJsboTCqlFXQcBEWHg89KBj7LJrmGZRdHYxSavbA164zPcniq","sender\_id":"FSTSMS","message":"Alert! Humidity exceeded the threshold of 90. Take necessary actions.","language":"english","route":"p","numbers":contactnumber}

headers = {

'cache-control': "no-cache"

}

response = requests.request("GET", url, headers=headers, params=querystring)

print(response.text)

print("Text message has been sent about Humidity reaching above threshold value.\nExiting python code now.")

exit(0)

print("Soil moisture content is",k)

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 = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()

**Screenshots for each task** :

