

GREEN HOUSE MONITORING AND CONTROL SYSTEM

SUBMITTED BY:

JEEVAN.A
SACHITH
SHASHANK

1.INTRODUCTION

1.1. OVERVIEW

The greenhouse industry is the fastest growing sector world-wide. The greenhouse separates the crop from the environment, thus providing some way of shelter from the direct influence of the external weather conditions. This enables the production of crops which otherwise could not be produced at that specific location. The greenhouse enclosure enables the manipulation of the crop environment. This asset allows the farmer to improve the cultivation in a way the plants need. It leads to higher crop yield, prolonged production period, better quality, and less use of protective chemicals.

1.2. PURPOSE

A greenhouse is a structure that is built of walls and a transparent roof and is designed to maintain regulated climatic conditions. These structures are used for the cultivation of plants, fruits, and vegetables which require a particular level of sunlight, temperature, humidity and soil moisture. IOT and Arduino based Greenhouse Environment Monitoring and Controlling Project is designed to maintain these conditions in the greenhouse. By using this project one can monitor the temperature, humidity and the soil moisture details in real time. He/she can also switch on or off the motor and decide the crop to yield using a web application or mobile application.

2. LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

Currently, there are numerous researches on greenhouse automation. These researches differ depending on the components that can be categorized as communication and control infrastructure, embedded system used on greenhouse, sensors and convertors gaining skills to system and user-interface. For infrastructure, several communication protocols such as wired data

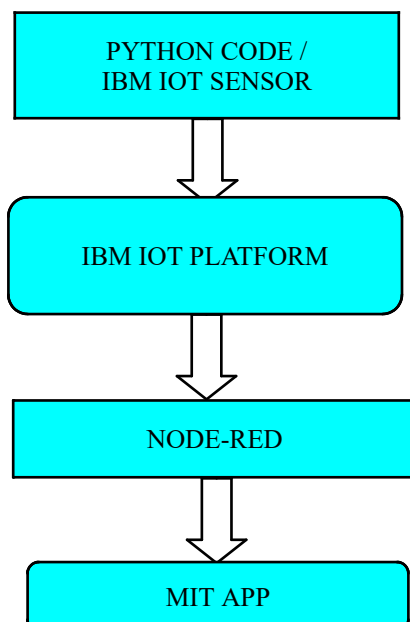
communication-based RS485, Bluetooth , CAN (Controller Area Network) , GPRS (General Packet Radio System), GSM (Global System for Mobile Communications) and Internet which hinge on system installation cost, distance and data transmission rate have been used. Since it communicates to the client with SMS functionality, the data updating will be extreme slow and the user interface are complicated.

2.2 PROPOSED SOLUTION :

The solution proposed to solve the problem is an integrated system using IBM IoT Platform where all the devices are connected through IBM Watson and two interfaces one is website and other is a mobile application is created to monitor the current temperature and humidity readings and with the help of the readings soil moisture can be calculated thus Motor can be turned on and off depending on the crop choosen at any time from any place.

3. THEORITICAL ANALYSIS:

3.1 BLOCK DIAGRAM :



3.2. SOFTWARE DESIGNING :

In the software designing part create a IBM cloud platform. In this design the raspberry pi model is used. the software should be design by taking avalues from the dht11 sensor and then sent to the IBM cloud services and then the data send to the mobile application which was developed using MIT app inventor. Here we use python language for coding, Node-Red ,etc.

4. EXPERIMENTAL INVESTIGATIONS:

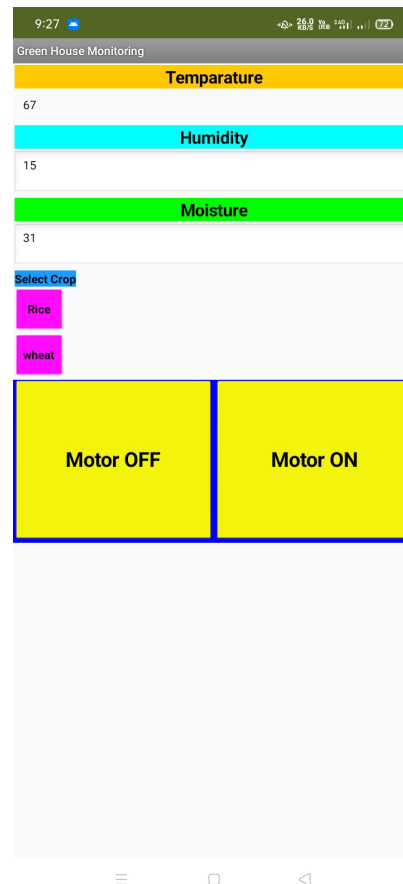
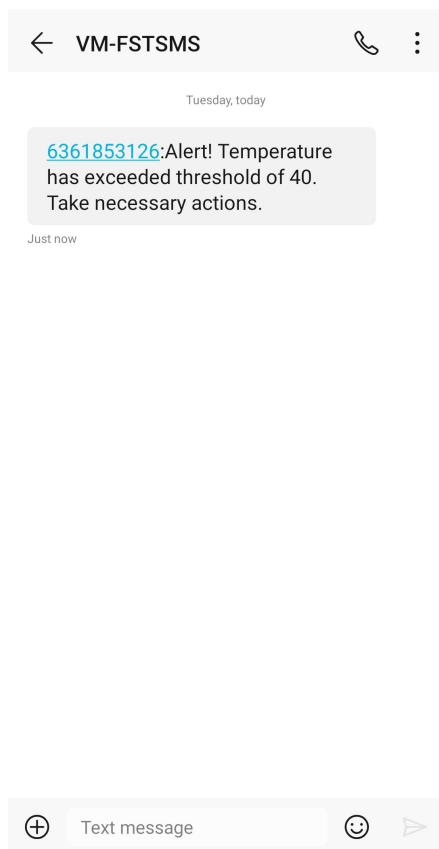
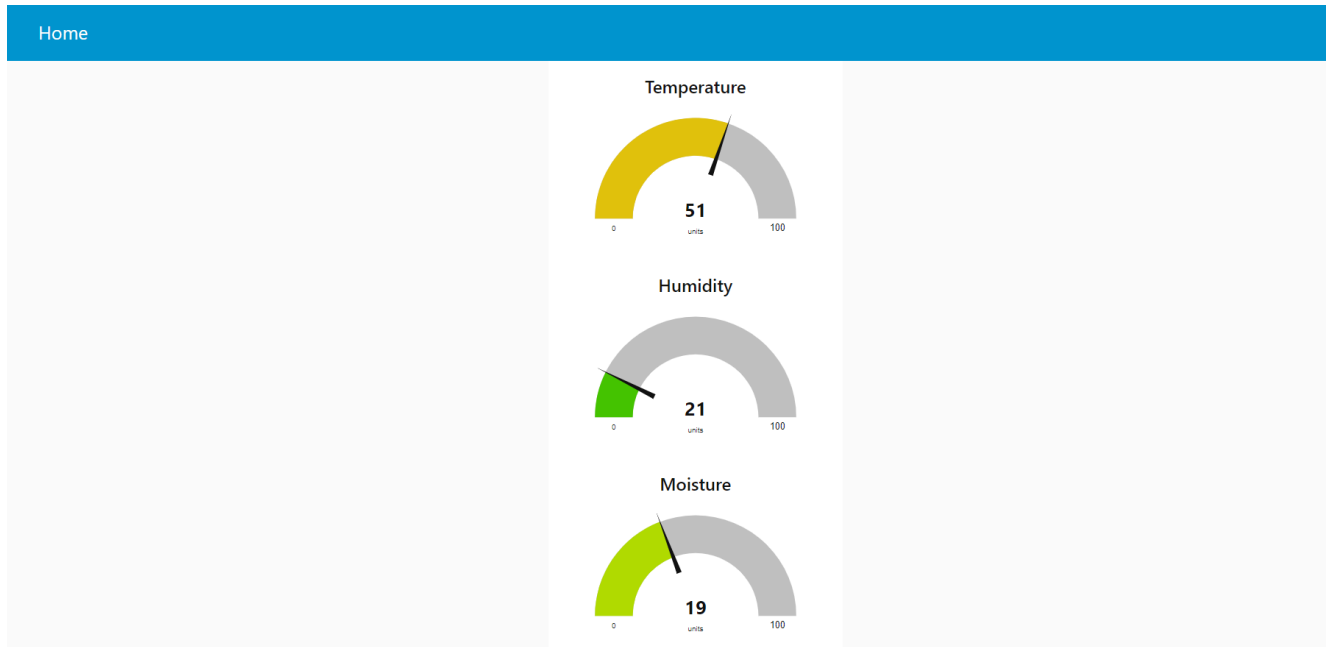
By integrating the applications using Watson IoT all the devices can be connected and thus they can be controlled from any place and at any time. Real time data can be seen from the mobile application and the online web application. Appropriate environmental conditions are necessary for optimum plant growth, improved crop yields, and efficient use of water and other resources. Automating the data acquisition process of the soil conditions and various climatic parameters that govern plant growth allows information to be collected with this system with less labor requirements. This IOT Greenhouse monitoring systems employs PC or phone-based systems for keeping the owner continuously informed of the conditions inside the greenhouse.

5. FLOWCHART



6. RESULT :

One can view the temperature, humidity and soil moisture anywhere and anytime and can control the green house.



7.ADVANTAGES :

- User friendly
- Easily implementable
- Focuses on main parameters
- Uses GSM because of their availability
- Easy network coverage
- Cloud computing provides Increased storage
- Easier group collaboration
- Resource continuity

DISADVANTAGES :

Requires higher initial capital investment for the distribution system and controls.

8.APPLICATION :

- Can be used in green houses to control the temperature, soil moisture, humidity and light for the proper growth of plants
- With little modification, this project can be used in Mechanical companies to measure various parameters of operating machines like temperature and light.
- 3. Temperature monitoring and controlling action can be used in home or various halls like conference room, seminar hall to control the temperature of room

9.CONCLUSION :

This project offers a design of fully automated greenhouse management system. From the experiment it could be seen that it is fulfilling all re-

quirements related greenhouse monitoring. The automatic greenhouse sensor design could help in increasing the productivity of plants. As it has been mentioned earlier, we are not only providing automatic control over the devices like light, motor pump but also we have a mechanism to alert farmers regarding the parameter changes in the greenhouse so that early precaution steps can be taken. Thus this construction, productivity of cropping can be continuously increased so it can handle famine problem around the world.

10. FUTURE SCOPE :

- We can monitor more parameters like PH of soil, pressure, water level and at the same time control them
- We can send this data to a remote location using mobile or internet
- We can draw graphs of variations in these parameters using computer
- This project can be further enhanced to monitor and control the pesticide level

11. BIBLIOGRAPHY:

- SmartBridge Videos
- www.slideshare.com
- www.google.com
- www.wikipedia.com

12. APPENDIX :

main.py

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

#Provide your IBM Watson Device Credentials
organization = "yh8ahx"
deviceType = "Rasperypi"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

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

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:

    hum=random.randint(10,40)
    #print(hum)
    temp =random.randint(30,80)
    moist=random.randint(15,60)
```

```

#Send Temperature & Humidity to IBM Watson
data = { 'temperature' : temp, 'humidity': hum , 'moisture': moist}
#print (data)
def myOnPublishCallback():
    print ("Published Temperature = %s C" % temp, "Humidity = %s %" %
    % hum, "Moisture = %s %" % moist, "to IBM Watson")

    success = deviceCli.publishEvent("DHT11", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
        time.sleep(2)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```

Fast2sms.py :

```

import time
import sys
import ibmiotf.application
import ibmiotf.device
import requests

contactnumber="6361853126"

organization ="yh8ahx"
deviceType ="Rasperypi"
deviceId ="1234"
authMethod ="use-token-auth"

```

```
authToken ="12345678"
```

```
url="www.fast2sms.com/dev/bulk"
```

```
def myCommandCallback(cmd):
```

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

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

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

```
    if(i<=40):
```

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

```
    else:
```

```
        print("Temperature exeeded 40. Text alert message sent to mobile")
```

```
    querystring = {"authorization":"vRciF0IXYqEGQM4oCKA-  
dazZS1VLNgrs9lfOB8D65xUwnPm2t3HlcLsQ8PAyST47Xj6fDwKnrqC9FEi-  
da"}
```

```
    headers = {
```

```
        'cache control':"no-cache"
```

```
    }
```

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

```
    print(response.text)
```

```
    print("Text message has been about Temperature reaching above threshold  
value. \nexiting python code")
```

```
    exit(0)
```

```
    if(j<=90):
```

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

```
    else:
```

```
        print("Humidity exeeded 90. Text alert message sent to mobile")
```

```
    querystring = {"authorization":"vRciF0IXYqEGQM4oCKA-  
dazZS1VLNgrs9lfOB8D65xUwnPm2t3HlcLsQ8PAyST47Xj6fDwKnrqC9FEi-  
da"}
```

```

headers = {
    'cache control': "no-cache"
}
response = requests.request("GET",url,headers=headers,params=querystring)

print(response.text)
print("Text message has been about Humidity reaching above threshold va-
lue. \nexiting python code")
exit(0)

if(k<=25):
    print("moisture",k,"(optimal)")
else:
    print("moisture exeeded 25. Text alert message sent to mobile")

querystring = {"authorization": "vRciF0IXYqEGQM4oCKA-
dazZS1VLNgrs9lfOB8D65xUwnPm2t3HlcLsQ8PAyST47Xj6fDwKnrqC9FEi-
da"}

headers = {
    'cache control': "no-cache"
}
response = requests.request("GET",url,headers=headers,params=querystring)

print(response.text)
print("Text message has been about Humidity reaching above threshold va-
lue. \nexiting python code")
exit(0)

```