# 1.INTRODUCTION

## 1.1 OVERVIEW

Greenhouse Environment, used to grow plants under controlled climatic conditions for efficient production, forms an important part of the agriculture and horticulture sector .To create an optimal environment the main parameters such as temperature, humidity, light intensity ,ground water ,etc. needs to be controlled. The main objective of this project work is to design an automated greenhouse which is purely sensor based system .The system inputs from various sensors and displays output .The developed system is simple, cost efficient and easily installable. The results show that the system could be more efficient in man power saving and raising the economic value of products.

Greenhouses are controlled area environment to grow plants. in order to achieve maximum plant growth, the continuous monitoring and controlling of environmental parameters such as temperature, humidity, soil moisture, light intensity, soil ph etc. are necessary for a greenhouse system.

## 1.2 PURPOSE

In recent scenario of climate change and its effect on the environment has motivated the farmers to install greenhouses in their fields. But maintaining a greenhouse and its plantation is very labour intensive and majority of them perform vital operations intuitively. Also agricultural researchers are facing shortage of good quality of data which is crucial for crop development. Thus we have developed such a cost effective system using Internet of Things (IoT) technology which focuses on solving the particular problems, our system automates the greenhouse maintenance operations and monitor the growth conditions inside the greenhouse closely .

# 2.LITERATURE SURVEY

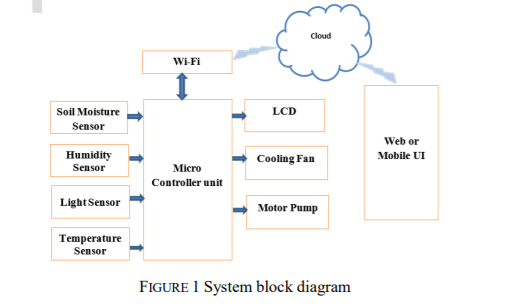
## 2.1 EXISTING PROBLEM

The newer scenario of decreasing water tables, drying up of rivers and tanks, unpredictable environment presents an urgent need for proper utilization of water. To cope up with this use of temperature and moisture sensor at suitable locations for monitoring of crops is implemented in. We also visited few greenhouses and observed and recorded the working methods of the framers, which provided me a very

clear idea how the maintenance and monitoring activities. proposed the techniques for selection hardware, provided basics and reference models on which an IoT system can be based and developed. Comparative study of some existing systems provided insights provides our first node to start, already available systems in IoT-powered gardening and agriculture, like Plantlink, Bitponics, and Harvest Geek are either not available in India or are very costly which add up a considerable production cost overhead on the crops or Agro-based products.

## 2.2 PROPOSED SOLUTION

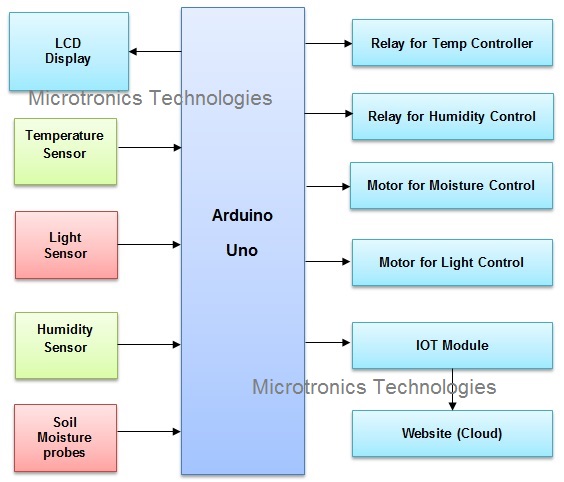
Appropriate environmental conditions are necessary for optimum plant growth, improved crop yields, and efficient use of water and other resources. Automating the data acquisition process of the soil conditions and various climatic parameters that govern plant growth allows information to be collected 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.The proposed design grabs data (such as temperature, humidity and moisture) from environments (such as greenhouse) and send to remote server via modem by using internet connection and also could operate remote commands. Data on the remote server could be viewed via both web and smart phone. Besides, remote server should keep track of records to check values whether they are in limits. When a value overflows outside of the boundaries, like if temperature increases dramatically, remote server should send alert to subscriber’s smart phone and turn on the cooling fan automatically. When user receives alert, he could also step in and prevent emergency situations manually .When the device is switched to automatic mode, the system will work as an intelligent system. That means, if the parameter value is exceeds the limit it will automatically trigger an event. When the manual mode is selected, we need to turn on and off each device individually.



# 

# 3.THEORETICAL ANALYSIS

## 3.1 Block Diagram



## 

## 

## 3.2 Hardware Description

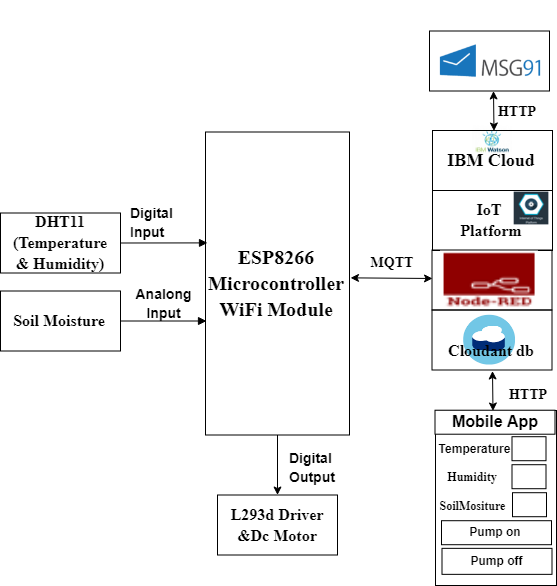
Design of hardware for green house monitoring and controlling are used to control the environment condition of green house to get a good condition. The parameters are humidity and temperature in the greenhouse. The monitoring and controlling of

greenhouse component consists of sensor for the humidity, Arduino UNO microcontroller, serial communication, wireless connection, LED module change for water sprayer, stepper motor, model of greenhouse, personal computer as server, and power supply unit. The output for the sensor become an input to microcontroller and sent to computer through serial communication. The task of the computer is to transfer the data through wireless communication to application software at Android Smartphone.The microcontroller will read the sensor periodically and updates the value of sensor to android.

## 3.3 Software Description

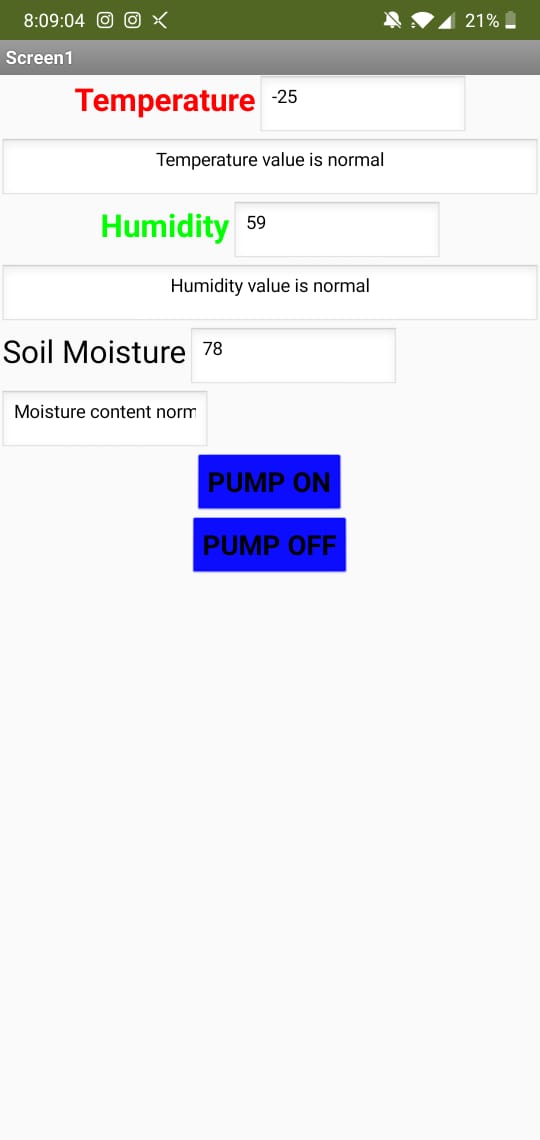
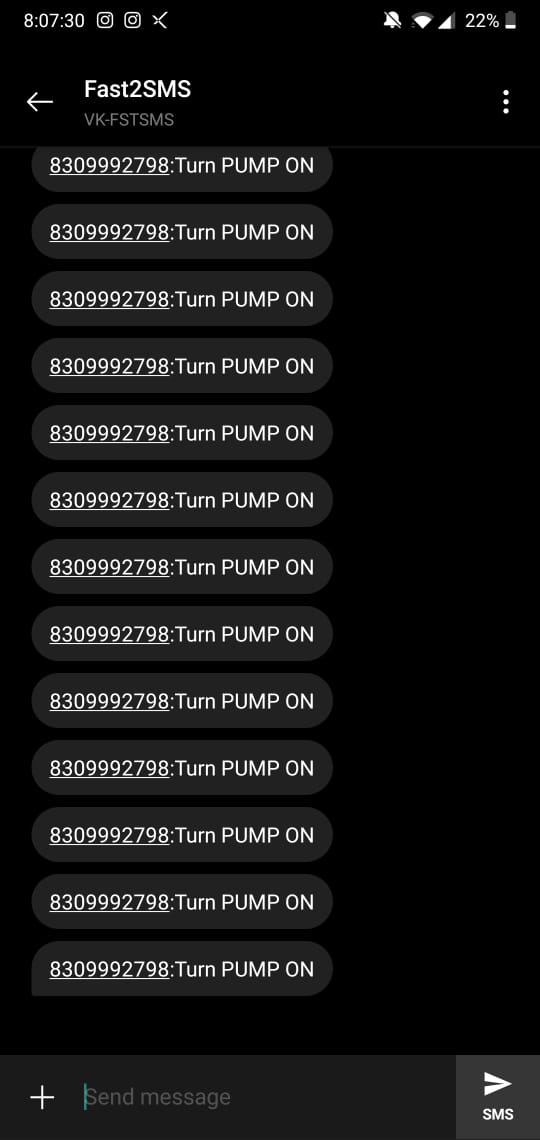
The software is designed to process the humidity value,Temperature and soil moisture monitoring and controlling the green house. The software includes the various measurements of the sensor, analog to digital converters, send humidity value from sensor to microcontroller. Then continue to display the value in application at Android, control the microcontroller from the application in Android and update to user by sending the value of sensor for monitoring the green house. The program is written in Python IDLE and the program is integrated with IBM IOT device. Node-RED is a programming tool for wiring together hardware devices, the values of IBM IOT device are shown in node red debug section after the connection of all the nodes. Mit App Inventor is used to create the app to show the values of Temperature, Humidity and Soil moisture values. If the sensor values exceed the threshold value then a notification is sent to the user to turn the pump on. This is done by using fast2sms.

# 5.FLOWCHART



# 6.RESULT

The approximate values of temperature, humidity and soil moisture are taken from python IDLE . These result can be seen on a internet website. A smart green house monitoring system is implemented successfully using the concept of Internet of Things which would be a boon for agriculture sector.

# 

# 

# 7.ADVANTAGES

The proposed system is going to play an important role in future of agriculture system and hopefully it would going to help in boosting the efficiency of growth and production of agriculture industry. Apart from that some of the important advantages of the proposed system are listed below.

 Easy to use

 Easy to implement

 More accurate results

 Increase Fertility

 Better Productivity

 Focus is on important parameters

# 8.APPLICATIONS

The project has a great application in agriculture sector and can be used in greenhouses, botanical gardens and agriculture farms. Temperature monitoring and controlling action can be used in home or various halls like conference room, seminar hall to control the temperature of room. With little modification, this project can be used in Mechanical companies to measure various parameters of operating machines like temperature and light.

# 8. CONCLUSION AND FUTURE SCOPE

This project describes about the various aspects of green house monitoring using IOT. Farming activities, even in urban zones are on an ascent as of late, in remarkable structures. Innovative advance makes the agrarian area develop high, which here is made by the IoT. The IoT will be playing a great role in changing the day to day life experience. The proposed project could be beneficial as it will help in advancing the assets in the nursery. And right now we are using limited number of parameters in our project but with the further advancement more number of parameters could be added for boosting the production. In future by building up a versatile application for IoT framework makes more adaptable to the people groups.

# BIBLIOGRAPHY:

[1] Yuquan, M., Shufen, H., Qingzhu, W., “New Environment Parameters Monitoring and Control System for Greenhouse Based on Master-Slave Distributed”, IEEE Conference Publications: 978-1- 4244-6947-5/10, 2010, Page(s): 31-35

2] Eldhose.K.A, Rosily Antony, Mini.P.K, Krishnapriya.M.N, Neenu.M.S, “Automated Greenhouse Monitoring System”, International Journal of Engineering and Innovative Technology (IJEIT) Volume 3, Issue 10, April 2014.

[3] Alausa Dele W.S, KeshinroKazeemKolawole, “Microcontroller Based Green House Control Device”, The International Journal Of Engineering And Science (IJES), Volume 2, Issue 11, Pages129-135, 2013.

# 

# APPENDIX

import time

import sys

import ibmiotf.application

import ibmiotf. device

import random as r

#Provide your IBM Watson Device Credentials

organization = "u55e4o"

deviceType = "raspberrypi"

deviceId = "1103"

authMethod = "token"

authToken = "8309992798"

# Initialize GPIO

def myCommandCallback(cmd):

print("Command received: %s" % 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= r.randint(0,100)

#print(hum)

temp = r.randint(-40,125)

moist = r.randint(0,100)

#Send Temperature & Humidity to IBM Watso

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 IoTF")

time.sleep(2)

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()

# 