DESIGN AND DEVELOPMENT OF IoT BASED AGRICULTURAL POLYHOUSE MONITORING SYSTEM

A project report in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND SYSTEMS ENGINEERING

Submitted by

T. Aishwarya (312106410001) **D.S. Kalyani (312106410011)**

D.V.H. Surya Teja (312106410012) **Md. Karimulla(312106410021)**

Ch. Sai Puneeth (312106410025) **U. Maneesha (312106410037)**

DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS ENGINEERING

ANDHRA UNIVERSITY COLLEGE OF ENGINEERING (A)

ANDHRA UNIVERSITY, VISAKHAPATNAM-530003

APRIL 2017

DEPARTMENT OF COMPUTER SCIENCE AND SYSTMS ENGINEERING

ANDHRA UNIVERSITY COLEGE OF ENGINEERING (AUTONOMOUS)

ANDHRA UNIVERSITY, VISAKHAPATNAM -530003

CERTIFICATE

This is to certify that the project titled "**DESIGN AND DEVELOPMENT OF IoT BASED AGRICULTURAL POLYHOUSE MONITORING SYSTEM**" is a certified record of work done by **Mr.D.V.H.SURYA TEJA (Regd No: 313106410012), Mr.CH.SAI PUNEETH (Regd No: 313106410025), Mr.MD.KARIMULLA (Regd No: 313106410021), Ms.U.MANEESHA (Regd No: 313106410037), Ms.T.AISHWARYA (Regd No: 313106410001), Ms. D.SATYA KALYANI (Regd No: 313106410011),** students of B.Tech in the Department of Computer Science & Systems Engineering, Andhra University College of Engineering (A), Andhra University, Visakhapatnam during the period 2013-2017 in the partial fulfilment of the requirements for the award of B.Tech. This work has not been submitted to any other university for the award of Degree or Diploma.

Prof. M. S. Prasad Badu
Project Guide
Department of CS & SE
Andhra University,

Prof. P. V. G. D. Prasad Reddy
Head of the Department
Department of CS & SE
Andhra University,

DECLARATION

We hereby declare that the project entitled **“DESIGN AND DEVELOPMENT OF IoT BASED AGRICULTURAL POLYHOUSE MONITORING SYSTEM”** is an original and authentic work done in the Department of Computer Science & Systems Engineering, Andhra University College of Engineering (A), Andhra University, Andhra University, Visakhapatnam, submitted in partial fulfilment of the requirements for the degree of Bachelor of Technology.

D. V. H. Surya Teja (313106410012)
Ch. Sai Puneeth (313106410025)
T.Aishwarya (313106410001)
D. S. Kalyani (313106410011)
U. Maneesha (313106410037)
Md. Karimulla (313106410021)

Signature

ACKNOWLEDGEMENT

We have immense pleasure pleasure in expressing our earnest gratitude to my Project Supervisor **Prof. M. S. Prasad Babu**,. Department of CS&SE, AUCE, Andhra University for his inspiring and scholarly guidance. Despite his pre-occupation with several assignments, he has been kind enough to spare his valuable time and gave me the necessary counsel and guidance. We received at every stage of planning and constitution of this work.

I express sincere gratitude to **Prof. P. V. G. D. Prasad Reddy**, Head of the Department, Computer Science and Systems Engineering, for having accorded us permission to take up this Project work and for helping us graciously throughout the execution of this work.

We are grateful to **Prof P. Avdhani, Principal**, College of Engineering, Andhra University, Visakhapatnam for having accorded us permission to visit Dr. Y. S. R. Horticulture University, Tadepalligudem.

We are deeply indebted to **Dr. J.Dilip Babu**, Director of Research **Dr. R.V.S.K.Reddy**, Director of Extension, Dr.Y.S.R Horticultural University, for their support in their campus.

We extend our sincere thanks to our academic teaching shaff and office-staff for their affection and help throughout our study.

CONTENTS

Page

LIST OF FIGURES **Page No**

ABSTRACT

Agricultural revolution is an emerging field focusing on the enhancement of agricultural and rural development through improved information and communication processes. More specifically, Agricultural revolution involves the conceptualization, design, development, evaluation and application of innovative ways to use information and communication technologies (IT) in the rural domain, with a primary focus on agriculture.

Internet and Communication Technologies (ICT) used to address these issues effectively. ICT is used extremely over a wide range of applications. The broadness of ICT covers any product that will store, retrieve, manipulate, transmit and receive information electronically in digital form. Smart farming has been representing the application of modern Information and Communication Technologies (ICT) into agriculture, leading to what can be called the third green revolution. These smart farming techniques have been incorporated in the design of a polyhouse.

Sensors have been used for the recording of data regarding the parameters of air quality, temperature, soil moisture, humidity and light intensity. This data is later stored in the database which is stored and used for analysis. Time to checking and monitoring is done to keep track of changes in the data. If there are any aberrations then measures are taken to bring the parameters back to normal.

The different data obtained from the sensors can be used for prediction and analysis helping the crop to yield better. The changes beyond a specific threshold are obtained and studied upon to understand why the change occurs and what can be done to maintain the regularity of getting stable outputs. In order to make this possible data the is read out of each sensor is wirelessly transmitted with the help of a transmitter and receiver using radio waves to the personal computer and then this data is stored in the database from which the data is retrieved for surveying.

Polyhouse farming is an alternative new technique in agriculture gaining foothold in rural India. It reduces dependency on rainfall and makes the optimum use of land and water resources. Potentially, polyhouse farming can help the farmer generate income around the year growing multiple crops. This also helps them spread their risks.

Conclusion:

This system collects and automatically controls the condition of greenhouse environment by using different sensors. The existing control system monitors the temperature, humidity, light intensity, water level and air quality in the greenhouse to solve the problem of plant disease. Regular monitoring of air quality give an indication of the detection of harmful gases that can cause problems that they affect to plant growth and quality. The system gives efficient information regarding the soil moisture levels to improve production and profit. During this project we realize that environment control processes are very precise and well controlled. We use the knowledge we obtain from the sensors and help the crop to use resources conservatively.

CHAPTER - 1

INTRODUCTION

Agriculture: Agriculture being one of the most important occupation for most Indian families is considered as a backbone of the country's economy. The technological advancements in this field are of at most importance. Collaborating the application of modern Information and Communication Technologies with agriculture results in smart farming which implements simpler and resourceful methods of farming.

1.1. Information and Communication Technologies (ICT)

Internet and Communication Technologies (ICT) are used to address these issues effectively for a wide range of applications in a very efficient and effective manner. The broadness of ICT actually involves the processes such as storing, retrieving, manipulating, transmitting and receiving information electronically in digital form.

ICT is used in various fields like Bio-Diversity, Agriculture, Landscape Ecology, Eco-Industrial applications, Health care, Environmental risk Management, Sustainable urban Development etc.

ICT in agriculture, also known as E-agriculture or Digital Agriculture, is the development and application of innovative concepts to use ICTs in the rural domain, with a primary focus on agriculture. ICT is used as a umbrella term encompassing all the information and communication technologies including devices, networks, mobiles, services, and applications.

1.2. Polyhouse

A polyhouse is a tunnel made of polyethylene, usually semi-circular, square or elongated in shape. In Polyhouse the crops can be protected from adverse environment such as high humidity or high temperature.

1.3. Poly-house Automation

PolyhouseAutomation involves the automatic monitoring of climatic parameters directly/indirectly govern the plant growth and their production.

1.3.1. Need of Polyhouse Automation

It is almost impossible for human being to understand and manipulate system with more than two dependent processes without additional aid. Hence the automatic controllers and computer-controlled polyhouses were introduced.

Automation leads to the increase in crop productivity as it provides automatic control, reduces human error and manages the unwanted environmental conditionals. Automation also minimizes the labour cost.

1.4. Scope of the Present Study

Designed to enable farming in controlled conditions which makes use of an automated mechanism for keeping track of the changes in various parameters of the cropsPolyhouse enables to grow crops throughout the year irrespective of the season. The quality of produce is better than open field cultivation.Farmers get to use resources more conservatively and it is more cost effective.Minimal human effort is required in this design of cultivation due to automation. It requires minimum labour, fertilizer and water to produce high quality crops.The storage of the data helps to keep track of trends of the data and perform analysis to get a better understanding of the type of the crops.

1.5. Organization of the Project Report

The project report is organized into seven chapters.

Chapter 1: INTRODUCTION

This chapter discusses brief description of Information and Communication Technologies. Polyhouse farming and Polyhouse automation is examined and the scope of the project is defined.

Chapter 2: PROBLEM ANALYSIS

This chapter deals with the motivation behind the problem and the problem statement. The current scenario is described in the existing system. A proposed system is discussed which is the main idea of the project. Objectives of the project are thoroughly analysed.

Chapter 3: REQUIREMENT ANALYSIS

This chapter gives a detail description of all the requirements either software or hardware. It presents an evaluation of the input and output requirements, computer and storage requirements, automation requirements and communication requirements.

Chapter 4: METHODOLOGY

In this chapter the proposed system has been discussed. The system architecture is described in detail where the system architecture contains 4 units namely data acquisition unit, data relay unit, data storage unit and data display unit.

Chapter 5: CODING

This chapter contains the code of various modules from retrieving sensor data and transmitting it wirelessly to storing the data and displaying the output.

Chapter 6: RESULTS AND DISCUSSIONS

This chapter evaluates the results obtained from the sensors and data storage and the outputs are discussed at each stage.

Chapter 7: CONCLUSIONS AND FUTURE EHANCEMENTS

This chapter describes the final conclusions and evaluates if the project meets the desired objectives successfully. The chapter also explores the future enhancements that can be accomplished in order to modify the project.

CHAPTER - 2

PROBLEM ANALYSIS

In this chapter, the problem statement is discussed. The motivation of the project in section 2.1, the objectives of the project in section 2.2, the existing system in section 2.3, the proposed system with system specifications, software and hardware components in the section 2.4 are discussed in this chapter.

Due to the unpredictable weather conditions that have been brought due to world's climate change, the global food shortage is being experienced. In conventional Agronomical practices, the crops are being cultivated in the open field under natural conditions where the crops are susceptible to sudden changes in climate like temperature, humidity, light intensity, soil moisture and other conditions that effects the crop production. In order to face these problems, greenhouse practices have been in existence for a very long time, which are now modernized and deployed in many parts of the world.

2.1. Motivation

In emerging greenhouse areas all over the world, there is a need to find balance between the cost of investment that goes into process of cultivation and the returns obtained, at the same time reducing the complexity and making the system easily accessible to farmers. One such approach is the practice of polyhouse farming.

2.2. Objectives

- Designed to enable farming in controlled conditions which makes use of an automated mechanism for keeping track of the changes in various parameters of the crops
- Polyhouse enables to grow crops throughout the year irrespective of the season. The quality of produce is better than open field cultivation.
- Farmers get to use resources more conservatively and it is more cost effective.
- Minimal human effort is required in this design of cultivation due to automation. It reuires minimum labour, fertilizer and water to produce high quality crops.

- The storage of the data helps to keep track of trends of the data and perform analysis to get a better understanding of the type of the crops.
- The analysis of data helps to recapitulate and evaluate the condensed specifics and makes it easy for prediction for the measures to be taken.

2.3. Existing System

The existing system of cultivation is mostly monitored manually. The available management is not sufficient to that of requirement. Today the farmer themselves manages all the things that crop requires, but every person do not have the exact knowledge of the quantity and the time required by the crop to produce more yield. The following are the drawbacks of existing system:

- Difficult to monitor manually.
- Requires more agricultural land.
- Crops are usually seasonal which curtails their availability during off- season.
- Difficult to handle adverse conditions like high temperature or high humidity

2.4. Proposed System

The proposed system is designed using Wireless Sensor Network(WSN) implemented using Arduino Programming, Python programming andStructured Query Language (SQL). The system features optimizing five parameters inside the polyhouse – temperature, light intensity, humidityand soil moisture and air quality. In order to measure these parameters, suitable sensors are deployed in the field which continuously monitors the environmental conditions. Microcontroller accesses the data from the sensors and transmits the data using transmitter. At the receiver's end, the transmitted data is received by the receiver which is read into the serial port of the system using a microcontroller. This data is stored in an SQL database which is further used for predictive analytics to predict yield of a crop, climate monitoring and forecasting, equipment monitoring, livestock tracking and geofencing and Sensor-based field and resource mapping etc.

2.4.1. System Specifications

The specifications of the system, used in the process of data communication, are as follows:

- Operating System: Windows 10
- RAM : 4GB
- Hard Disk : 1TB
- Server : Apache

The software and hardware components that are used in the design of the project are as follows:

2.4.2. Software Components

- Arduino IDE1.8.2
- Pycharm
- MySQL
- Android Studio 2.1

2.4.3 . Hardware Components

- Micro-controller :Arduino Uno(ATmega328P)
- Sensors : The following sensors are used
    - Temperature Sensors (LM35)
    - Light Intensity Sensor (LM 393)
    - Humidity Sensor (DHT11)
    - Air Quality Sensor (MQ135)
    - Soil Moisture Sensor (YL69 connected to YL38)
- RF module consisting of transmitter and receiver for wireless transmission
- Breadboards
- Male and Female connecting Wires
- 9V Battery
- USB cables
- Copper Wires

CHAPTER - 3

REQUIREMENT ANALYSIS

In this chapter, the input requirements in section 3.1 , output requirements in section 3.2, computer requirements in section 3.3, storage requirements in section 3.4, automation requirements in section 3.5 and communication requirements in section 3.6 are discussed.

3.1. Input Requirements

Plant growth in polyhouse is effected by various parameters like temperature, soil moisture, air quality, humidity, light intensity etc. Various sensors are used to collect data about these parameters. Sensors used in this study are:

- Temperature Sensors (LM35)
- Light Intensity Sensor (LM 393)
- Humidity Sensor (DHT11)
- Air Quality Sensor (MQ135)
- Soil Moisture Sensor (YL69 connected to YL38)

3.2. Output Requirements

System must be designed to enable farming in controlled conditions which makes use of an automation for keeping track of the changes in various parameters of the crops. Minimal human effort must be required in this design of cultivation due to automation.

3.3. Computer Requirements

The specifications of the computer system used in this study are:

Operating System : Windows 10

RAM : 4GB

Hard Disk : 1TB

Server : Apache

3.4. Storage Requirements

The data received by the system is stored in excel files. This data is then stored into database using Structured Query Language (SQL) along with the time stamp. The data is also sent to the web server using PHP. The saved data can be periodically analyzed to predict situations that are most suitable for the plant growth. These conditions are then reciprocated to increase the crop yield. The software necessary for storage is:

- Apache server
- MySQL Database

3.5. Automation Requirements

Based on the changes in parameters discussed, Automation mechanisms are developed. Water sprinklers are deployed for maintaining moisture content in the soil.

Heat generators are deployed for providing heat during adverse cold conditions. Fans are used to maintain the temperature levels. Bulbs are used to provide light for plants when needed.

3.6. Communication Requirements

The data from the sensors is collected from the microcontroller using RF transmitter and receiver. The microcontroller connected to the computer is used to read the data received by the RF receiver. This data is then transmitted serially to the serial port system. Python code is used to extract data from the serial port. The data thus collected is stored in the form of excel file in the system.

CHAPTER – 4

METHODOLOGY

In this chapter, the design of the proposed system has been discussed. The system architecture has been dicussed in the section 4.1 by explaining the components involved in the polyhouse automation. The working of the design is explained in 4ins 4 sections namely data acquisition unit in section 4.2.1, data relay unit in section 4.2.2, data storage unit in section 4.2.3 and data display unit in section 4.2.4.

The proposed system makes use of five parameters namely-temperature, air quality, soil moisture, light intensity and humidity. These five parameters are extracted from the environment using respective sensors. This data is then sent remotely to the receiver which in turn sends the data to the system through the serial port. The data thus collected is stored in a database on which analysis is performed at the later stages. The process goes as follows:

- The sensors get the data which is read by the micro-controller at the transmitter sideusing the data pin of the respective sensor.
- This data is then transmitted remotely to a receiver.
- The micro-controller at the receiver end receives the data and puts it in the serial port.
- The system reads the data from the serial port.
- The values are then placed in a database.

4.1. System Architecture

The proposed system is designed using Wireless Sensor Network(WSN) implemented using Arduino Programming, Python programming andStructured Query Language (SQL). The system features optimizing five parameters inside the polyhouse – temperature, light intensity, humidityand soil moisture and air quality. In order to measure these parameters, suitable sensors are deployed in the field which continuously monitors the environmental conditions. Microcontroller accesses the data from the sensors and transmits the data using transmitter. At the receiver's end, the transmitted data is received by the receiver which is read into the serial port of the system using a microcontroller. This data is stored in an SQL database which is further used for analysis.

The system architecture for monitoring the polyhouse is shown below:

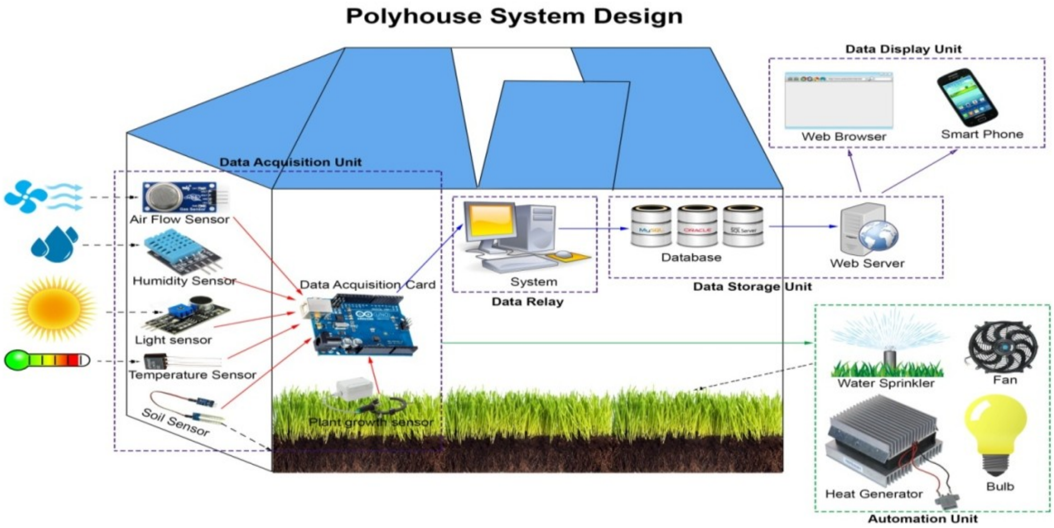


Figure 4.1 System Architecture

For the present proposed system mainly five climatic parameters are monitored using their respective sensors. The sensors used in the design are as following:

- <u>Temperature Sensor (LM35)</u>: This sensor is used to measure the temperature with an electrical output proportional to the temperature (in °C).

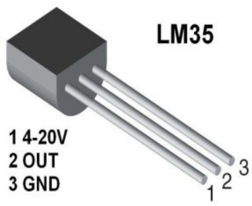


Figure 4.2 LM35 Temperature Sensor

- <u>Soil Moisture Sensor (YL69 connected to YL38):</u> This sensor is used to measure the water content in the soil.

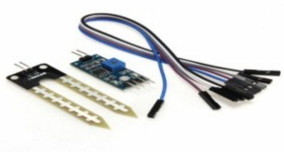


Figure 4.3 YL69 connected to YL38 Soil Sensor

- Air Quality Sensor (MQ135): This sensor is used to detect smoke and harmful gases especially ammonia (NH_3).

Figure 4.4 MQ135 Air Quality Sensor

- Humidity Sensor (DHT11): The humidity sensor senses, measures, and reports both moisture and air temperature.

Figure 4.5 DHT11 Humidity Sensor

- Light Intensity Sensor (LM 393): The light senor is used to detect the current ambient light level. The brightness of the surrounding environment and the light intensity can be detected.

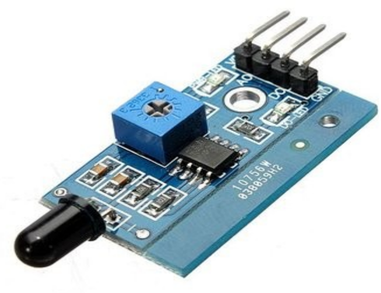


Figure 4.6 LM393 Light Sensor

* **Microcontroller**

Arduino UNO is based on the ATmega328p micro-controller. The ATMEGA328p on the Arduino UNO comes pre-programmed with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol.

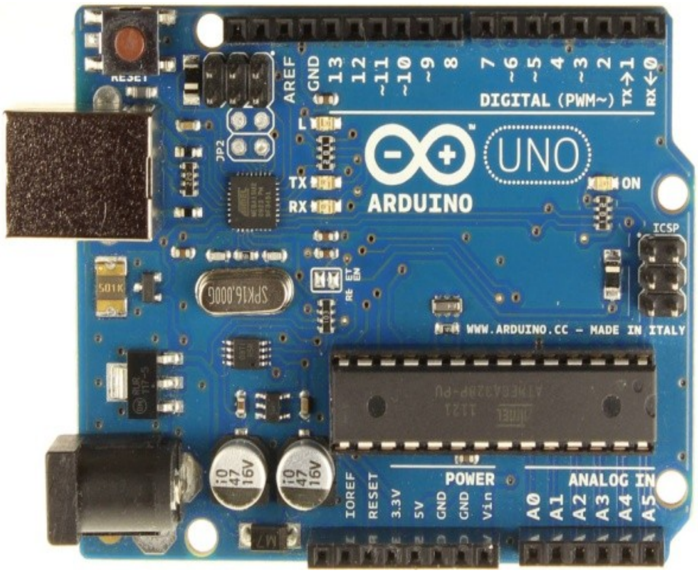


Figure 4.7 Arduino UNO Board

* **RF Module**:

The RF module consists of a transmitter and a receiver. The RF module is a small electronic device used to transmit and/or receive radio signals between two devices. In an embedded system it is often desirable to communicate with another device wirelessly. This wireless communication can be may be accomplished through radio frequency communication.

Figure 4.8 RF Module

4.2. Design Implementation

The system architecture consists of four units namely data acquisition unit, data relay unit, data storage unit and data display unit.

4.2.1 Data Acquisition Unit

The core of this unit is the data acquisition card i.e. the microcontroller. Arduino UNO is the micro controller used in the system to which the sensors are interfaced to it using breadboard and connecting wires.

The sensors used to monitor polyhouse are as follows:

- Temperature sensor (LM35)
- Humidity sensor (DHT11)
- Soil Moisture Sensor (YL69 connected to YL38)
- Air Quality Sensor (MQ135)
- Light Intensity Sensor(LM393)

Initially the micro-controller is loaded with a program that enables it to read the data from the data pins of the sensors. This data is sent through the RF transmitter to the RF receiver in the form of radio waves. On the receiver side, the RF receiver that is connected to the microcontroller senses the radio waves. The micro-controller which is loaded with program that enables the data is received via antenna of the RF receiver and read into the microcontroller.

The interfacing of sensors with the microcontroller using bread board and connencting wires is shown as below.

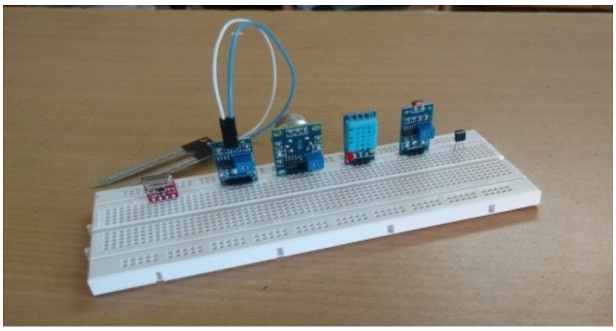


Figure 4. 9 Sensors connected to the breadboard before interfacing

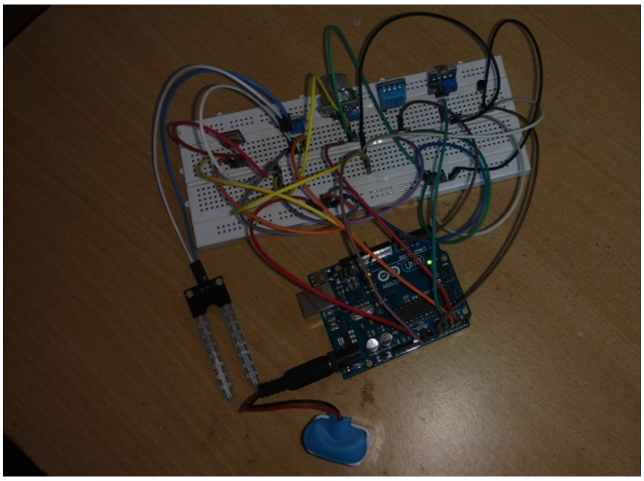


Figure 4. 10 Sensors interfaced to the microcontroller

4.2.2 Data Relay Unit

The data relay unit works as intermediate node between the data acquisition unit and the data storage unit. This unit provides reliable communication between these units.

The microcontroller connected to the computer is used to read the data received by the RF receiver. This data is then transmitted serially to the serial port system. Python code is used to extract data from the serial port. The data thus collected is stored in the form of excel file in the system.

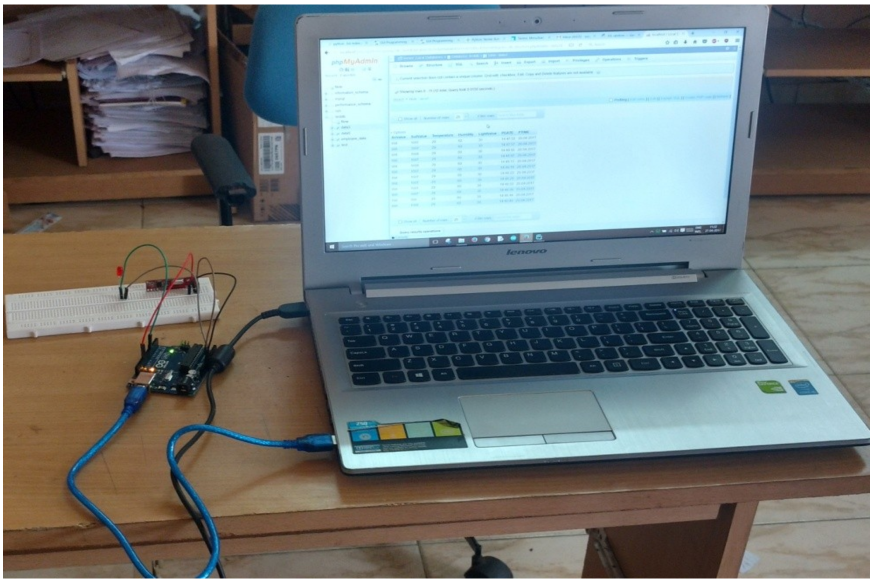


Figure 4. 11 Computer Receiving the data from the sensors using RF receiver

4.2.3 Data Storage Unit

The data storage unit involves storing the data in the database and web server from which the data can be extracted for future use.

The data received by the system is stored in excel files. This data is then stored into database using Structured Query Language (SQL) along with the time stamp. The data is also sent to the web server using PHP. The saved data can be periodically analyzed to predict situations that are most suitable for the plant growth. These conditions are then reciprocated to increase the crop yield.

4.2.4 Data Display Unit

The data display unit involves the displaying of the data obtained from the sensors to the users.

The data stored in the database or the web server is displayed on the web browser or inside an application in the smart phones. The user can see the data values that are obtained from the sensors and the user can take necessary actions based on these values. When the data from a particular sensor overcomes the threshold value specified, a notification can be generated to the user on the application in the phone.

CHAPTER – 5

CODING

5.1 Arduino Code for RF transmitter

5.1.1 Algorithm for RF transmitter

Step 1: start

Step 2: Include the libraries dht.h and VirtualWire.h

Step3:Define the transmit pin, receive pin, data pin and sensor pins

Step4: Define the setup()

Step4.1: Configure the transmit pin.

Step4.2: Configure the receive pin

Step4.3: Configure the transmit enable pin, or "push to talk".

Step4.4: Configure the "push to talk" polarity.

Step4.5: Begin using all settings and initialize the library.

Step5:Define the loop()

Step5.1: write the led pin to high

Step5.2: read the data from the temperature sensor

Step5.3: read the data from the light sensor

Step5.4: read the humidity sensor pin

Step5.5: Store the humidity in an integer variable

Step5.6: Store the temperature in an integer variable

Step5.7: read the data from the soil sensor

Step5.8: read the data from the air sensor

Step5.9: array of characters to store the message

Step5.10: store the character in the character array

Step5.11: append a break character

Step5.12: store the character in the character array

Step5.13: append a break character

Step5.14: store the character in the character array

Step5.15: store the character in the character array

Step5.16: append a break character

Step5.17: store the character in the character array

Step5.18: wait till whole message is sent

Step5.19: write the digital pin to low

Step5.20: delay for 5 seconds before the next value is taken

Step6:Stop

5.1.2. Code for RF Transmitter

```
#include<dht.h>   // include the library required for accessing DHT 11 sensor

#include <VirtualWire.h>

const int ledPin = 13;  //defining the Led Pin

const int transmit_pin = 12;  //defining the transmitpin

const int receive_pin = 2;    //defining the receive pin

const int transmit_en_pin = 3;            //defining the transmit pin

const int tempPin = A0;       //Pin at which temperature sensor is connected

const int lightPin = A1; //Pin at which light sensor is connected

const int soilPin = A2; //Pin at which soil senor is connected

const int airPin = A3;  //Pin at which air Flow sensor is connected

#define DHT11_PIN 7           // defining the data pin

dht DHT;    //defining a variable of the type DHT

void setup() {

// put your setup code here, to run once:

  vw_set_tx_pin(transmit_pin);            //defining the transmit pin

  vw_set_rx_pin(receive_pin);       //defining the receive pin

  vw_set_ptt_pin(transmit_en_pin);  //defining the transmit enable pin

  vw_set_ptt_inverted(true);        // Requiredfor DR3100

  vw_setup(2000);

}

void loop() {

  // put your main code here, to run repeatedly:

 digitalWrite(ledPin,HIGH);         //write the led pin to high

int val = analogRead(tempPin);            //read the data from the temperature sensor

int mv = (val/1024)*5000;

int cel = mv/10;
```

```
int lightValue = analogRead(lightPin);    //read the data from the light sensor

int densityValue = DHT.read11(DHT11_PIN); //read the humidity sensor pin

int l = DHT.humidity;          // Store the humidity in an integer variable

int t = DHT.temperature;            // Store the temperature in an integer variable

int soilValue = analogRead(soilPin);      //read the data from the soil sensor

int airValue = analogRead(airPin);  //read the data from the air sensor

int m;

int j=0;

char sensorCharMsg[50]; //array of characters to store the message

while(cel>0){

m = lightValue%10;

sensorCharMsg[j] = m+'0';     //store the character in the character array

j=j+1;

cel =cel/10;

}

sensorCharMsg[j] = 'b';       //append a break character

j=j+1;

while(lightValue>0){

  m = lightValue % 10;

  sensorCharMsg[j] = m+'0';   //store the character in the character array

  j=j+1;

  lightValue = lightValue/10;

}

sensorCharMsg[j] = 'b'; //append a break character

j=j+1;

while(l>0){

  m = l % 10;

  sensorCharMsg[j] = m+'0';    //store the character in the character array

  j=j+1;

  l=l/10;
```

```
}
sensorCharMsg[j] = 'b';       //append a break character
j=j+1;
while(t>0){
  m = t % 10;
  sensorCharMsg[j] = m+'0';   //store the character in the
character array
  j=j+1;
  t=t/10;
}
sensorCharMsg[j] = 'b';       //append a break character
j=j+1;
while(soilValue > 0)
{
  m = soilValue%10;
  sensorCharMsg[j]=m+'0';     //store the character in the
character array
  soilValue = soilValue/10;
  j=j+1;
}
sensorCharMsg[j]='b';   //append a break character
j=j+1;
while(airValue>0)
{
m=airValue%10;
sensorCharMsg[j]=m+'0'; //store the character in the character
array
airValue = airValue/10;
j=j+1;
}
sensorCharMsg[j]='b';   //append a break character
j=j+1;
sensorCharMsg[j]='a'; //store the character in the character
array
j=j+1;
vw_send((uint8_t *)sensorCharMsg, j);//send the character array
over the transmitter
vw_wait_tx(); //wait till whole message is sent
```

```
digitalWrite(ledPin,LOW);//write the digital pin to low

delay(5000);//delay for 5 seconds before the next value is
taken

}
```

5.2 Arduino Code for RF Receiver

5.2.1 Algorithm for RF Receiver

Step 1: start

Step 2: Include the libraries dht.h and VirtualWire.h

Step3:Define the transmit pin, receive pin, data pin and sensor pins

Step4: Define the setup()

Step4.1: receive the data every second

Step4.2: Baud rate or Data rate

Step4.3: enable the transmit pin

Step4.4: enable the receive pin

Step4.5: enable the receive pin

Step4.6: required for DR3100

Step4.7: Bits per sec

Step4.8: Start the receiver PLL running

Step5: Define the loop()

Step5.1: buffer to store the data sent

Step5.2: send the length of messages

Step5.3:Check for non-blocking

Step5.4: Flash a light to show received good message

Step5.5: Initialise for loop till the end of the buffer

Step5.6: Send the data to the port

Step5.7: print new line on the serial port

Step6:Stop

5.2.2 Code for RF Receiver

```
#include <VirtualWire.h>//include the virtual wire library for
accessing the Rf module

const int led_pin = 6; //defining the Led Pin
```

```
const int transmit_pin = 12;//defining the transmit pin

const int receive_pin = 11;//defining the receiver pin

const int transmit_en_pin = 3;//defining transmit enable pin

void setup()

{

    delay(1000);  //receive the data every second

    Serial.begin(9600);  // Baud rate or Data rate


// Initialise the IO and ISR

    vw_set_tx_pin (transmit_pin);//enable the transmit pin

    vw_set_rx_pin (receive_pin);//enable the receive pin

    vw_set_ptt_pin (transmit_en_pin); //enable the transmit pin

    vw_set_ptt_inverted (true); // required for DR3100

    vw_setup (2000);  // Bits per sec

    vw_rx_start ();       // Start the receiver PLL running

}

void loop ()

{

    uint8_t buf [VW_MAX_MESSAGE_LEN]; //buffer to store the
data

    uint8_t buflen = VW_MAX_MESSAGE_LEN; //length of  messages
sent

    if (vw_get_message (buf, &buflen)) // Non-blocking

    {

  int i;

   digitalWrite (led_pin, HIGH); // Flash a light to show
received good message

  for (i = 0; i < buflen; i++) //Initialise for loop till the
end of the buffer
```

```
{
    Serial.print (char (buf[i])); //Send the data to the port
}
Serial.println();     //print new line on the serial port
      digitalWrite (led_pin, LOW); //Write the LED pin to low
  }
}
```

5.3. Storage of Data in Database

5.3.1 Algorithm for Storage of Data in Database

Step 1: start

Step 2: import the serial module, pymysql module, arrow module
Step 3: connect to the database
Step 4:create an object for the cursor
Step 5: write an sql query for the creation of the table

Step 6: execute the sql query

Step 7: connect to the serial port

Step 8: open a workbook and check for the active worksheets

Step 9: loop as long as the serial port is open

Step 10: read the data from the serial port and store in database

Step 11: reverse the data

Step 12: split the data using 'b' as a delimiter and store in the list named templist

Step 13: store the date in t6

Step 14: store the time in t7

Step 15: open a text document

Step 16: checking for the active worksheets in the workbook

Step 17: loop to iterate through all the elements in the templist

Step 18: write the values in to the cells

Step 19: sql query to insert values into the database

Step 20: save the values

Step 21:close the database

Step 22:Stop

5.3.2. Code for Storage in Database

```
import serial    //import  the serial module

import pymysql   //import the pymysql module

import arrow     // import arrow module

from openpyxl import Workbook      //import Workbook from
openpyxl module

db = pymysql.connect("localhost", "root", "", "TestDb")
     //connect to the database

cursor = db.cursor()//create an object for the cursor

rows = 1

sql = """CREATE TABLE DATA3(AirValue VARCHAR(20), SoilValue
VARCHAR (20),Temperature VARCHAR(20),Humidity VARCHAR
(20),LightValue VARCHAR (20), PDATE VARCHAR (20), PTIME
VARCHAR(20))"""         //write an sql query for the creation
of the table

cursor.execute(sql)    //execute the sql query

ser = serial.Serial(port="COM3", baudrate=9600,
bytesize=serial.EIGHTBITS, parity=serial.PARITY_NONE,
timeout=6, stopbits=1)       //connect to the serial port

wb = Workbook()  // open a workbook

worksheet = wb.active       //check for the active worksheets

worksheet.cell(row=rows, column=2).value = "Air Value"

worksheet.cell(row=rows, column=3).value = "Soil Value"

worksheet.cell(row=rows, column=4).value = "Humidity"

worksheet.cell(row=rows, column=5).value = "Temperature"

worksheet.cell(row=rows, column=6).value = "Celsius"

worksheet.cell(row=rows, column=7).value = "Light Value"

worksheet.cell(row=rows, column=8).value = "Time"

worksheet.cell(row=rows, column=9).value = "Date"

rows = 2
```

```
while ser.is_open:     //loop as long as the serial port is
open

data = ser.readline().decode('ascii')    //read the data from
the  serial port and store in data

data = data[::-1]//reverse the data

templist = data.split('b') //split the data using 'b' as a
delimiter and store in the list named templist

t1 = templist[1]

t2 = templist[2]

t3 = templist[3]

t4 = templist[4]

t5 = templist[5]

t6 = templist[6]

t6 = arrow.now().format('DD-MM-YYYY')    // store the date in
t6

t7 = arrow.now().format('HH:mm:ss')            // store the
time in t7

date = arrow.now().format('DD-MM-YYYY')

time = arrow.now().format('HH:mm:ss')

fo = open("temp.txt", "a")   //open a text document

fo.write(data)   //write data to the file

worksheet = wb.active   //checking for the active worksheets
in the workbook

i = 0

for i in range(0, len(templist)): //loop to iterate through
all the elements in the templist

worksheet.cell(row=rows, column=i + 1).value = templist[i]
//write the values in to the cells

rows = rows + 1   //increment the row count

cursor.execute("INSERT INTO
DATA3(AirValue,SoilValue,Temperature,Humidity,LightValue,PTIME,
```

```
PDATE) VALUES (%s,%s,%s,%s,%s,%s,%s)", (t1, t2, t3, t4, t5, t6,
t7))        //sql query to insert values into the database

db.commit()             //save the values

wb.save("data.xlsx")          //save the excel sheet

db.close()        //close  the database
```

5.4.1 Python code for Admin Panel

```
from tkinter import * // import tkinter module

import serial //  import serial module

ser = serial.Serial(port="COM3",  baudrate=9600,
bytesize=serial.EIGHTBITS,        parity=serial.PARITY_NONE,
timeout=6, stopbits=1) //access serial port

root = Tk()

def NewFile():

    print("new file")

def OpenFile():

    print("open file")

def About():

    print("This is a simple example of a menu")

menu = Menu(root)

root.config(menu=menu)

filemenu = Menu(menu)

menu.add_cascade(label="Home", menu=filemenu)

# filemenu.add_command(label="New", command=NewFile)

# filemenu.add_command(label="Open...", command=OpenFile)

# filemenu.add_separator()

# filemenu.add_command(label="Exit", command=root.quit)

helpmenu = Menu(menu)

menu.add_cascade(label="Help", menu=helpmenu)
```

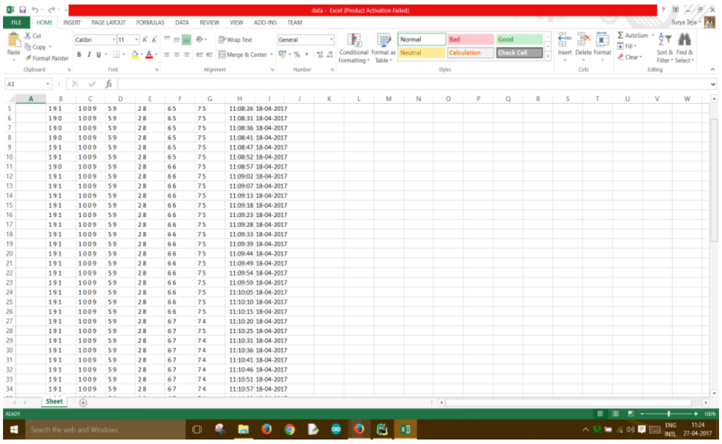
```
 helpmenu.add_command(label="About...", command=About)

 visualizationmenu = Menu(menu)

 menu.add_cascade(label="Visualization",
menu=visualizationmenu)

 def update():

     if ser.is_open:

         data = ser.readline().decode('ascii')

         data = data[::-1]

         templist = data.split('b')

         w.config(text=templist[1])

         t.config(text=templist[2])

         p.config(text=templist[3])

         q.config(text=templist[4])

         r.config(text=templist[5])

         w.place(x=100, y=90)

         t.place(x=250, y=90)

         p.place(x=400, y=90)

         q.place(x=550, y=90)

         r.place(x=700, y=90)

  root.after(5000, update)

 a = Label(root, text="Air Value", bg="blue", fg="white")

 a.place(x=100, y=30)

 b = Label(root, text="Soil Value", bg="blue", fg="white")

 b.place(x=250, y=30)

 c = Label(root, text="Humidity", bg="blue", fg="white")

 c.place(x=400, y=30)

 d = Label(root, text="Temperature", bg="blue", fg="white")

 d.place(x=550, y=30)
```

```
e = Label(root, text="Light Intensity", bg="blue", fg="white")

e.place(x=700, y=30)

w = Label(root, text=0, bg="red", fg="white")

w.place(x=100, y=90)

t = Label(root, text=0, bg="red", fg="white")

t.place(x=250, y=90)

p = Label(root, text=0, bg="red", fg="white")

p.place(x=400, y=90)

q = Label(root, text=0, bg="red", fg="white")

q.place(x=550, y=90)

r = Label(root, text=0, bg="red", fg="white")

r.place(x=770, y=90)

root.after(5000, update) //refresh at every 5s interval

root.mainloop() //loop indefinetly
```

CHAPTER – 6

RESULTS AND DISCUSSIONS

6.1 Sensors data



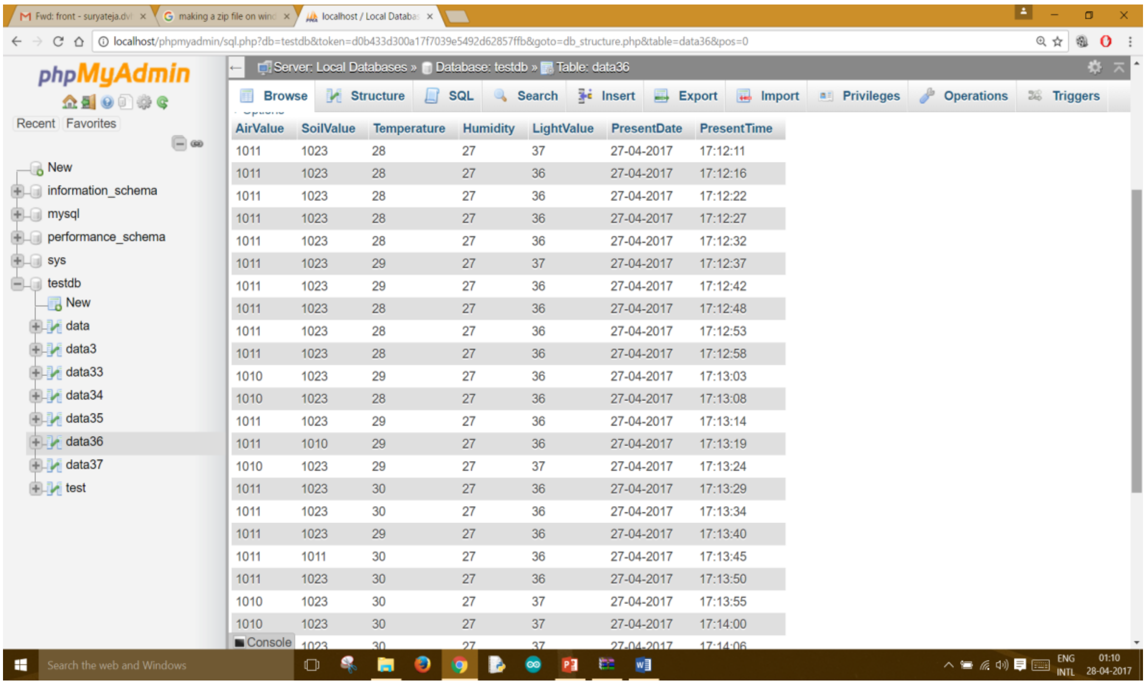
	A	B	C	D	E	F	G	H	I
5		191	1009	59	28	65	75	11:08:26	18-04-2017
6		190	1009	59	28	65	75	11:08:31	18-04-2017
7		190	1009	59	28	65	75	11:08:36	18-04-2017
8		190	1009	59	28	65	75	11:08:41	18-04-2017
9		191	1009	59	28	65	75	11:08:47	18-04-2017
10		191	1009	59	28	65	75	11:08:52	18-04-2017
11		190	1009	59	28	66	75	11:08:57	18-04-2017
12		191	1009	59	28	66	75	11:09:02	18-04-2017
13		191	1009	59	28	66	75	11:09:07	18-04-2017
14		191	1009	59	28	66	75	11:09:13	18-04-2017
15		191	1009	59	28	66	75	11:09:18	18-04-2017
16		191	1009	59	28	66	75	11:09:23	18-04-2017
17		191	1009	59	28	66	75	11:09:28	18-04-2017
18		191	1009	59	28	66	75	11:09:33	18-04-2017
19		191	1009	59	28	66	75	11:09:39	18-04-2017
20		191	1009	59	28	66	75	11:09:44	18-04-2017
21		191	1009	59	28	66	75	11:09:49	18-04-2017
22		191	1009	59	28	66	75	11:09:54	18-04-2017
23		191	1009	59	28	66	75	11:09:59	18-04-2017
24		191	1009	59	28	66	75	11:10:05	18-04-2017
25		191	1009	59	28	66	75	11:10:10	18-04-2017
26		191	1009	59	28	66	75	11:10:15	18-04-2017
27		191	1009	59	28	67	74	11:10:20	18-04-2017
28		191	1009	59	28	67	75	11:10:25	18-04-2017
29		191	1009	59	28	67	74	11:10:31	18-04-2017
30		191	1009	59	28	67	74	11:10:36	18-04-2017
31		191	1009	59	28	67	74	11:10:41	18-04-2017
32		191	1009	59	28	67	74	11:10:46	18-04-2017
33		191	1009	59	28	67	74	11:10:51	18-04-2017
34		191	1009	59	28	67	74	11:10:57	18-04-2017

Figure 6.1 Sensor Data in Excel file

Discussion

The data from the sensors about various parameters is collected and stored in an excel file. The data is displayed at different timestamps.

6.2 Database result



AirValue	SoilValue	Temperature	Humidity	LightValue	PresentDate	PresentTime
1011	1023	28	27	37	27-04-2017	17:12:11
1011	1023	28	27	36	27-04-2017	17:12:16
1011	1023	28	27	36	27-04-2017	17:12:22
1011	1023	28	27	36	27-04-2017	17:12:27
1011	1023	28	27	36	27-04-2017	17:12:32
1011	1023	29	27	37	27-04-2017	17:12:37
1011	1023	29	27	36	27-04-2017	17:12:42
1011	1023	28	27	36	27-04-2017	17:12:48
1011	1023	28	27	36	27-04-2017	17:12:53
1011	1023	28	27	36	27-04-2017	17:12:58
1010	1023	29	27	36	27-04-2017	17:13:03
1010	1023	28	27	36	27-04-2017	17:13:08
1011	1023	29	27	36	27-04-2017	17:13:14
1011	1010	29	27	36	27-04-2017	17:13:19
1010	1023	29	27	37	27-04-2017	17:13:24
1011	1023	30	27	36	27-04-2017	17:13:29
1011	1023	30	27	36	27-04-2017	17:13:34
1011	1023	29	27	36	27-04-2017	17:13:40
1011	1011	30	27	36	27-04-2017	17:13:45
1011	1023	30	27	36	27-04-2017	17:13:50
1010	1023	30	27	37	27-04-2017	17:13:55
1010	1023	30	27	37	27-04-2017	17:14:00

Figure 6.2 Data stored in database

Discussion:

The data stored in excel file is imported into database. Data will be stored in the form of tables in database.

6.3 Data Display

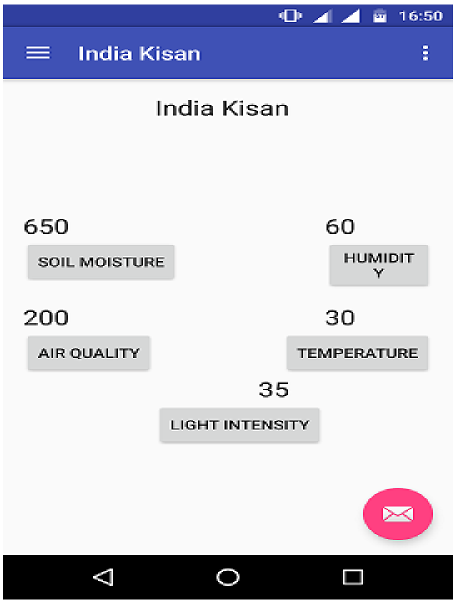


Figure 6.3 Android application home page

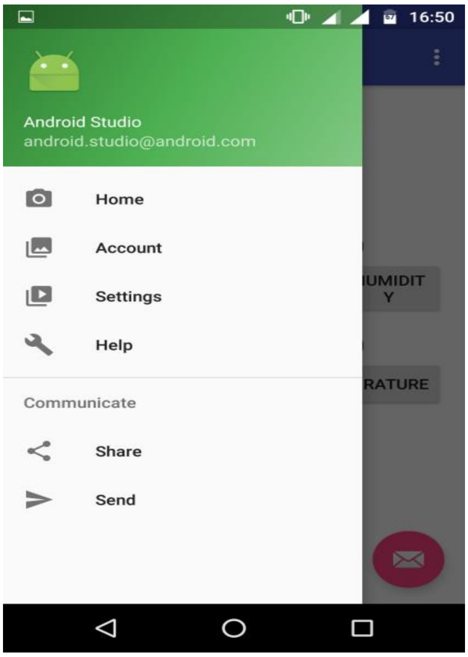


Figure 6.4 Android Application Menu

Discussion:

The data from database is sent to the web server. The data is then displayed in smart devices, which is then used to analyze to use the resources conservatively.

6.4 Admin Display Panel

Discussion:

The data is continuously displayed on the admin panel so that any aberrations in the climatic conditions may be detected.

CHAPTER – 7

CONCLUSION AND FUTURE ENHACEMENT

7.1. Conclusion

In this project, sensors are used to collect the data related to the parameters - temperature, soil moisture, humidity, air quality and light intensity. In data acquisition unit the data is collected from the sensors and transmitted to the data relay unit using RF module. In data relay unit, the received data is stored in excel files. In data storage unit, the data obtained in the data relay unit is also stored into the database and web server simultaneously. In data display unit, the data from the storage unit can be displayed on the control panel of the system or on the web browser or inside an android application in smart phones. Using this feature the farmer can monitor these parameters by checking the data values when necessary. This data can be further used for analytical process and for prediction. The data analysis provides the relationship between the environmental parameters and the growth of plant. The data trends have been observed and are recorded accordingly.

7.2. Future Enhancement

This system can be further enhanced by adding CO_2 sensor which allows the farmer to monitor the Carbon dioxide levels inside the polyhouse. The data analysis can be further used for prediction of the environmental conditions like temperature or growth based upon the previous data values and using pattern matching techniques.

REFERENCES

[1] Prof. M. S. Prasad Babu "**Convergence of Digital Technologies for Transforming India into a World Power**" Presidential Address in the Section of ICT, Indian Science Congress, January 2007.

[2] Luigi Atzori a, Antonio Iera b, Giacomo Morabito "**The Internet of Things: A survey**" Computer Networks 54 (2010) 2787–2805.

[3] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusi , Marimuthu Palaniswami " **Internet of Things (IoT): A vision, architectural elements, and future directions**" Future Generation Computer Systems 29 (2013) 1645–1660.

[4] Naiqian Zhang, Maohua Wang, Ning Wang "**Precision agriculture- A Worldwide Overview**" Computers and Electronics in Agriculture, 36 (2002) 113-132.

[5] D.D.Chaudhary, S.P.Nayse, L.M.Waghmare, "**Application Of Wireless Sensor Networks for Greenhouse Parameter Control in Precision Agriculture**" International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 1, February 2011

[6] Jagannathan, S.; Priyatharshini, R. "**Smart Farming System Using Sensors for Agricultural Task Automation**", Proceedings of the IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR), Chennai, India, 10–12 July 2015; pp. 49–53.

[7] Blackmore, S. (1994). "**Precision Farming: An Introduction. Outlook on Agriculture**" 23(4) 4,275-280.

[8] IF Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). "**Wireless sensor networks: Survey on Computer Networks**" , 38, 393-422.

[9] Ning Wang, Naiqian Zhang, Maohua Wang, "**Wireless sensors in agriculture and food Industry —Recent Development and Future Perspective**", published in Computers and Electronics in Agriculture 50 (2006) 1–14.

[10] A Baggio, "**Wireless Sensor Networks in Precision Agriculture**," 2005.

[11] Yu-Pin Lin et al. "**Applications of Information and Communication Technology for Improvements of Water and Soil Monitoring and Assessments in Agricultural Areas—A Case Study in the Taoyuan Irrigation District**" Special Issue Information and Communication Technologies for Improving Monitoring and Management Schemes in Agricultural Environments, MDPI, 17 January 2017.

BIBILOGRAPHY

1. http://agritech.tnau.ac.in/horticulture/horti_Greenhouse%20cultivation.html
2. https://www.arduino.cc/en/main/arduinoBoardUno
3. http://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/
4. https://www.olimex.com/Products/Components/Sensors/SNS-MQ135/resources/SNS-MQ135.pdf
5. http://www.ti.com/product/LM35
6. https://www.engineersgarage.com/electronic-components/rf-module-transmitter-receiver
7. https://randomnerdtutorials.com/guide-for-soil-moisture-sensor-yl-69-or-hl-69-with-the-arduino/
8. http://www.electrodragon.com/analog-and-digital-sense-of-sensors-lm393-the-voltage-comparator/
9. https://www.jetbrains.com/pycharm/documentation/