**SRI SHANMUGHA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(Approved By AICTE, Accredited by NAAC, Affiliated to Anna University)**

*Tiruchengode – Sankari Mani Rd, Pullipalayam, Morur (PO), Sankari (Tk), Salem 637304.*

# AI8713-ICT IN AGRICULTURAL ENGINEERING LAB EXCERCISES

## DEPARTMENT OF AGRICULTURE ENGINEERING

**Anna University - Regulation: 2017**

**B.E AGRICULTURE ENGINEERING – VII SEMESTER**

**AI8713-ICT IN AGRICULTURAL ENGINEERING LAB EXCERCISES**

# GENERAL INSTRUCTIONS

- All the students are instructed to wear protective uniform and shoes before entering into the laboratory.
- Before starting the exercise, students should have a clear idea about the principles of that exercise
- All the students are advised to come with completed recorded and corrected observation book of previous experiments, defaulters will not allowed to do their experiment.
- Don’t operate any instrument without getting concerned staff member’s prior permission.
- All the instruments are costly. Hence handle them carefully, to avoid fine for any breakage.
- Almost care must be taken to avert any possible injury while on laboratory work. In case, anything occurs immediately report to the staff members.
- One student from each batch should put his/her signature during receiving the instrument in instrument issue register**.**

**SRI SHANMUGHA COLLEGE OF ENGINEERING AND TECHNOLOGY,**

**SANKARI (TK), SALEM-637304.**

(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)

# RECORD NOTEBOOK

**REGISTER NO**

Certified that this is a bonafide record of practical work done by

Mr/Ms_____ of the _____

Semester _____branch during the academic year_____

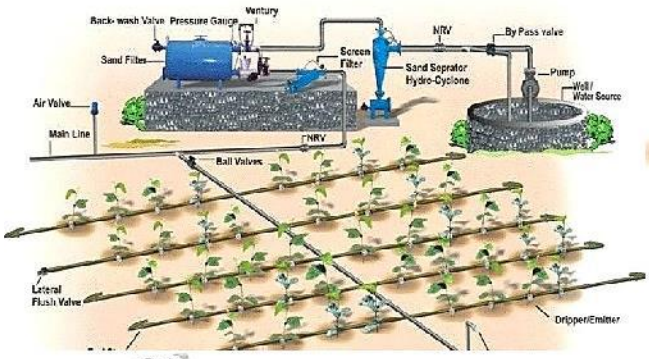
in the AI8713- ICT in Agricultural Engineering Lab Exercises

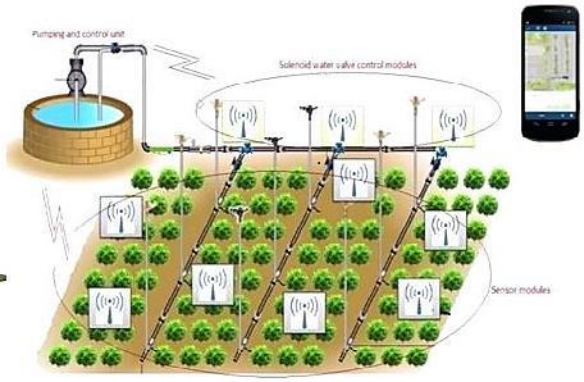
**Staff-in-charge** **Head of the Department**

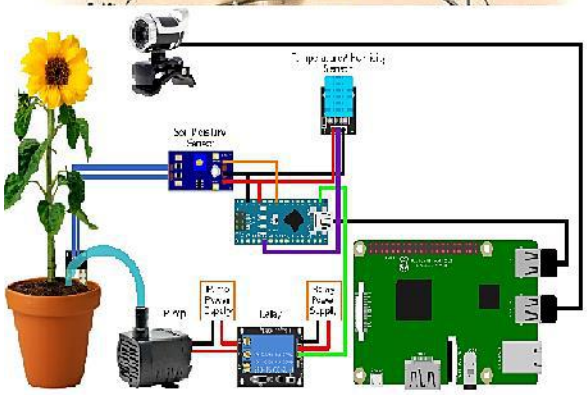
Submitted for the university practical examination held on_____

**Internal Examiner** **External Examiner**

# Table of Contents


Back-wash Valve
Pressure Gauge
Ventury
Sand Filter
Screen Filter
NRV
By Pass valve
Sand Seprator Hydro-Cyclone
Pump
Well / Water Source
Air Valve
Main Line
NRV
Ball Valves
Lateral Flush Valve
Dripper/Emitter


Pumping and control unit
Solenoid water valve control modules
Sensor modules


Soil Moisture Sensor
Pump
Relay

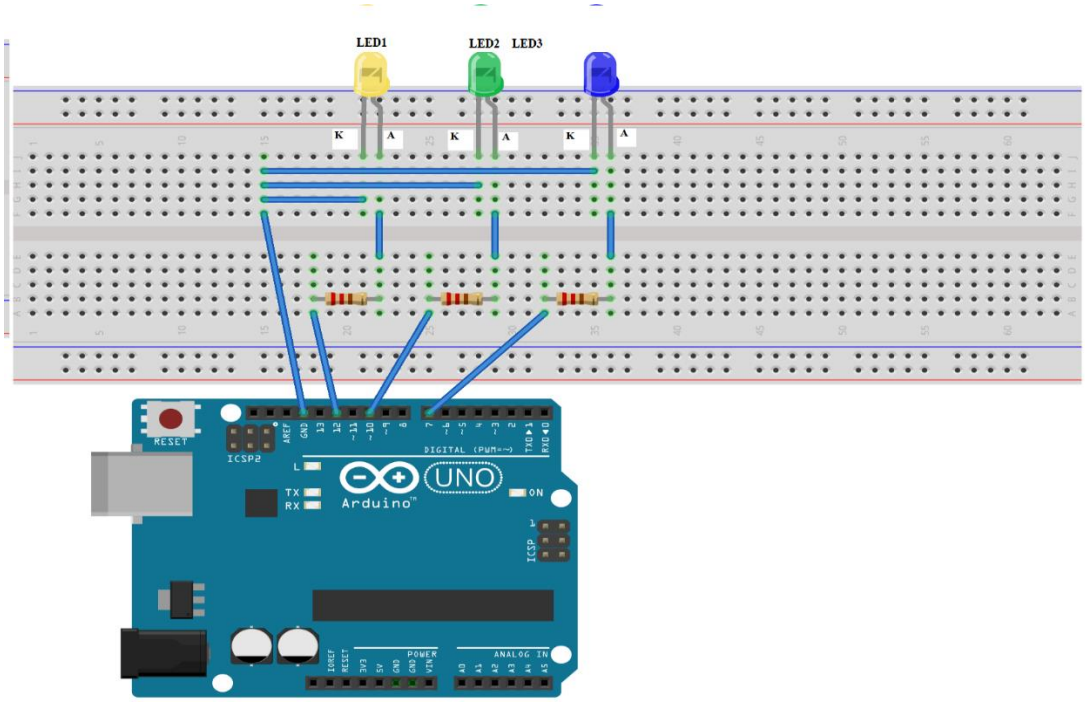
**ICT Laboratory for Agriculture Engineers**

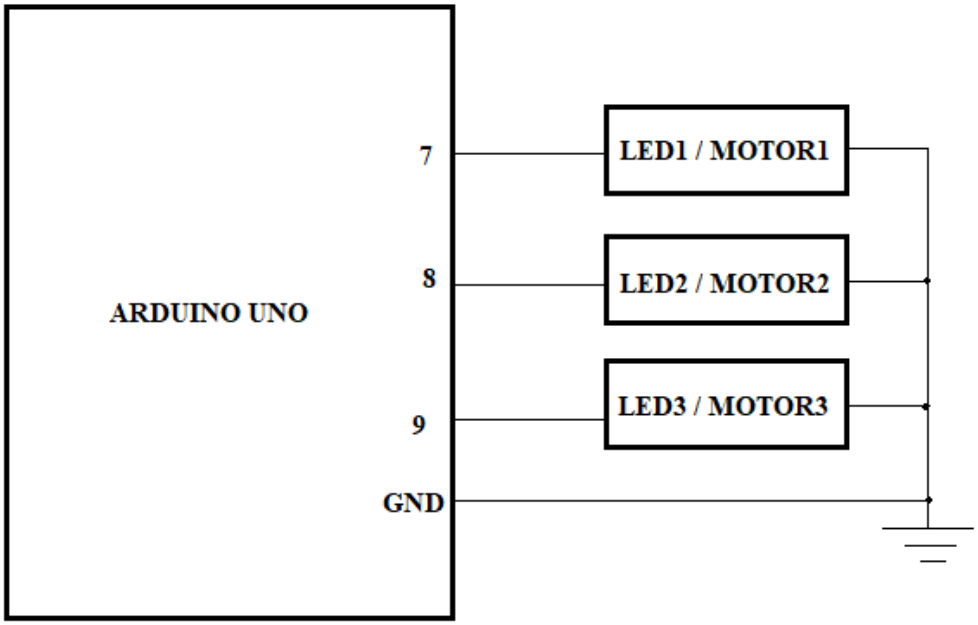
1. Configuring timers for automatic switching "on and off" of irrigation systems (With the available timing devices students should experience on how to configure and switch the small pumps or other devices. Requirements – timing devices and small pumps for simulations)
2. Experience with solenoid valves for pressurized irrigation (For distributing irrigation to different parts of the field in drip or sprinkler irrigation system solenoid valves need to be used for maintaining uniform pressure for better distribution of water. Student should be exposed how to use solenoid valves for this purpose. Requirements – Solenoid valves and layout of drip or sprinkler system)
3. Using sensors for Agro meteorological measurements (Mainly soil moisture and air temperature measurements. The Time Domain Reflectometer (TDR) can be used to measure soil moisture and any digital thermometer can be used to measure temperature. Making connections and measurements are important.)
4. Employing Printed Circuit Board (PCB) or Breadboard for controlling or triggering an agricultural system (Taken the measurements of temperature or soil moisture in the experiment No. 3 using a circuit board an event can be triggered ie if the soil moisture is below certain level the motor should be switched on. Similarly in a greenhouse if the temperature exceeds say 40ºC then fogger should be switched on. Requirements - Breadboards, relays etc.)
5. Use of mobile apps for controlling or triggering an agricultural system (An existing mobile application may be used to demonstrate this exercise. For example already applications are available for remotely switching motor or solenoid valves for irrigation.)
6. Construction of crop growth functions (best fit) for crop yields simulations (Data sets of x and y can be given to the students so that students can fit a best growth functions either using spread sheet application or MATLAB and draw graphs)
7. Image processing as tool for biotic and abiotic stress identification (Using the facilities available in MATLAB leaf images can be processed to identify degree

of damage including area calculation or to understand whether it is biotic or abiotic)

8. Experience with existing open source crop simulation models (Any one of the crop simulation models available as open source like DSSAT, InfoCrop, APSIM, EPIC may be demonstrated to the students to expose them crop growth simulation)
9. Exposing cloud resources for agricultural applications (The students should be demonstrated with use of clouds for exchanging information. For example the measured data like Temperature or soil moisture can be automatically send to clouds (a local cloud can be build for this purpose or existing cloud service can be used) for access and take decisions whether to irrigate or pay attention.)
10. Developing automated agro advisory systems (Based on observed and forecasted weather information agro-advisories can be developed automatically and advisory send to the farmers as SMS or by IVR. This exercise can be demonstrated based on the exercises already completed by the students in this lab.)

## CONNECTION DIAGRAM:



**EXP. NO:** **DATE:**

## CONFIGURING TIMERS FOR AUTOMATIC SWITCHING "ON AND OFF" OF IRRIGATION SYSTEMS

**AIM:**

To configuring timers for automatic switching "On and Off" of irrigation systems using Arduino development board

**COMPONENTS NEEDED:**

1. Arduino UNO development board
2. Motor (or) LED
3. USB cable
4. Breadboard
5. Connecting wires

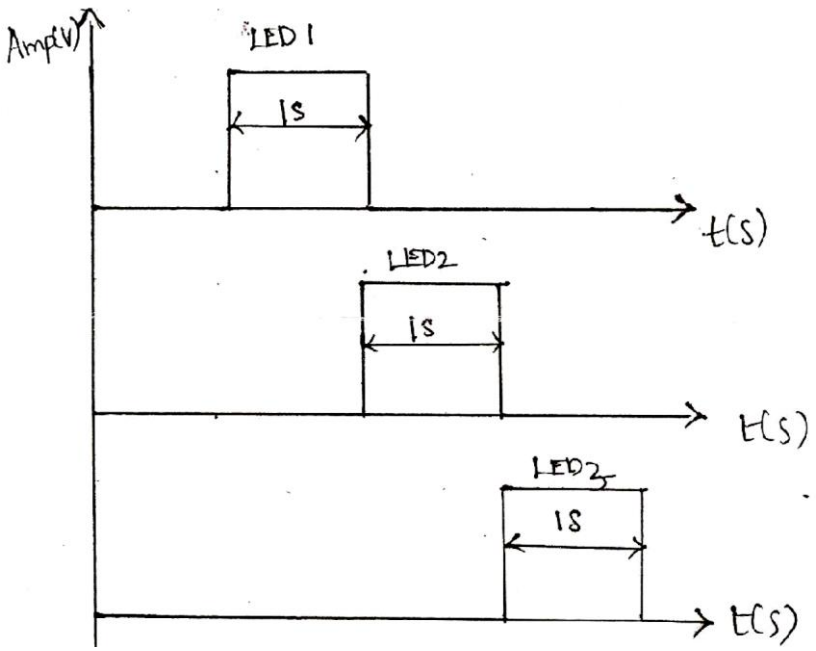
**PROGRAMMING:**

```
void setup() {
  // put your setup code here, to run once:
pinMode(7,OUTPUT);
pinMode(10,OUTPUT);
pinMode(12,OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(7,HIGH);
delay(1000);
digitalWrite(7,LOW);
digitalWrite(10,HIGH);
delay(1000);
digitalWrite(10,LOW);
digitalWrite(12,HIGH);
delay(1000);
digitalWrite(12,LOW);
}
```

**TECHNICAL SPECIFICATIONS OF ARDUINO UNO:**

| | | |
|---|---|---|
| Microcontroller | - | ATmega328 |
| Operating Voltage | - | 5V |
| Supply Voltage (recommended) | - | 7-12V |
| Maximum supply voltage (not recommended) | - | 20V |
| Digital I/O Pins | - | 14(of which 6 provide PWM o/p) |
| Analog Input Pins | - | 6 |

**MODEL GRAPH:**



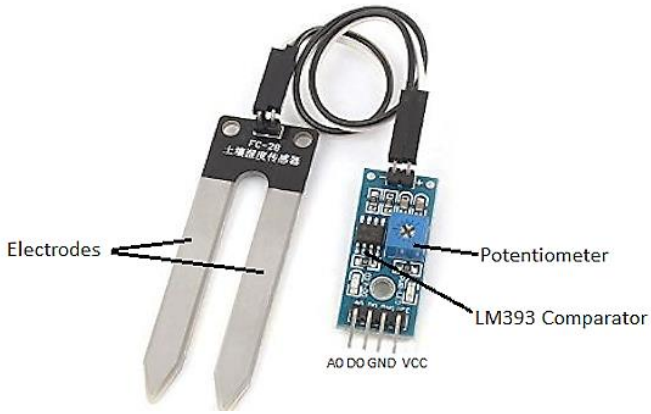
| | | |
|---|---|---|
| DC Current per I/O Pin | - | 40 mA |
| DC Current for 3.3V Pin | - | 50 mA |
| Flash Memory | - | 32 KB (ATmega328) of which 0.5KB Used by Bootloader |
| SRAM | - | 2 KB (ATmega328) |
| EEPROM | - | 1 KB (ATmega328) |
| Clock Speed | - | 16 MHz |

**PROCEDURE:**

1. Arduino development board is connected to PC through USB cable
2. Open "Arduino" icon from the computer desktop
3. Select **tools** menu→ choose Arduino board type
4. Select **tools** menu→ choose Arduino board COM port
5. Select **file** menu→ choose new and type program
6. Click **Verify icon→** once compilation done to upload the program to Arduino controller by clicking **Upload icon**
7. The circuit connections were given as shown in connection diagram.
8. When the given the LEDs are glows one by one sequentially.

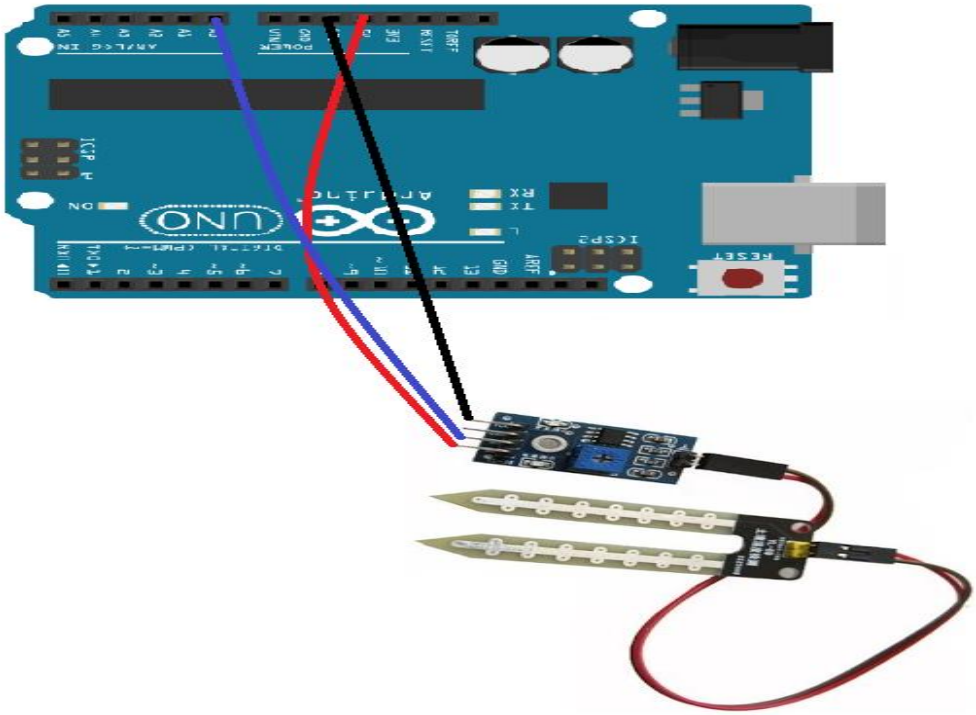
**RESULT:**

**PINOUT DIAGRAM:**




**Pin-out for Moisture Sensor**

**CONNECTION DIAGRAM:**



**EXP. NO:** **DATE:**

## INTERFACE MOISTURE SENSOR FOR AGRO METEOROLOGICAL MEASUREMENTS

**AIM:**

Program to interface moisture sensor for Agro meteorological measurements using Arduino board

**COMPONENTS NEEDED:**

1. Arduino board
2. Moisture sensor
3. Connecting wires

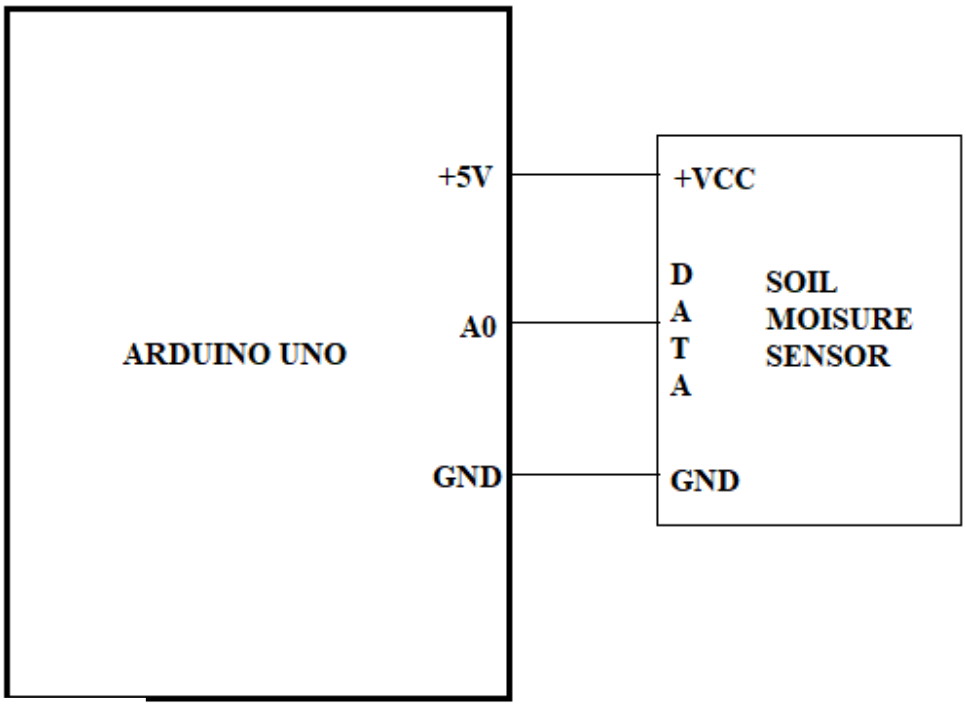
**HARDWARE CONNECTIONS:**

Moisture sensor To Arduino
Vcc --- 5v
GND --- GND
A0 --- A0 (Analog Pin of Arduino)

**PROGRAMMING:**

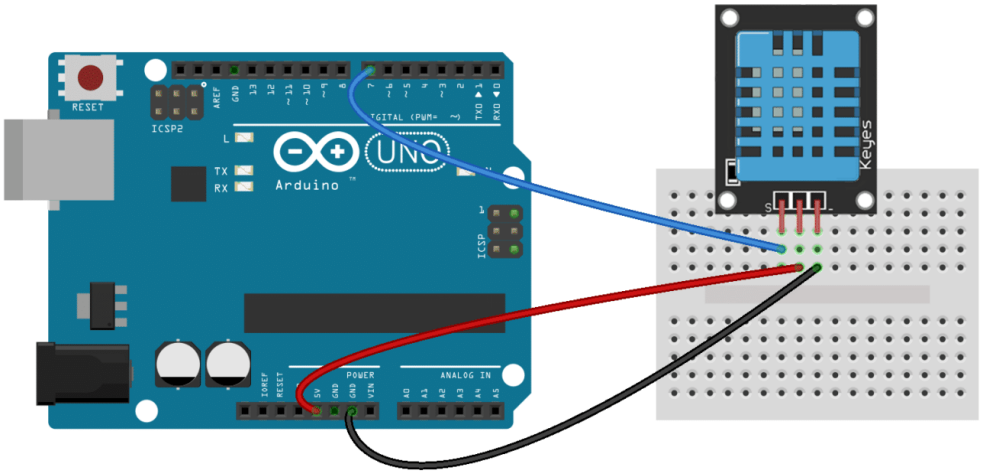
```
int sensor_pin = A0;
int output_value ;
void setup() {
  Serial.begin(9600);
 Serial.println("Reading From the Sensor ...");
  delay(2000);
  }
void loop() {
  output_value= analogRead(sensor_pin);
  output_value = map(output_value,550,0,0,100);
  Serial.print("Mositure : ");
  Serial.print(output_value);
  Serial.println("%");
  delay(1000);
  }
```

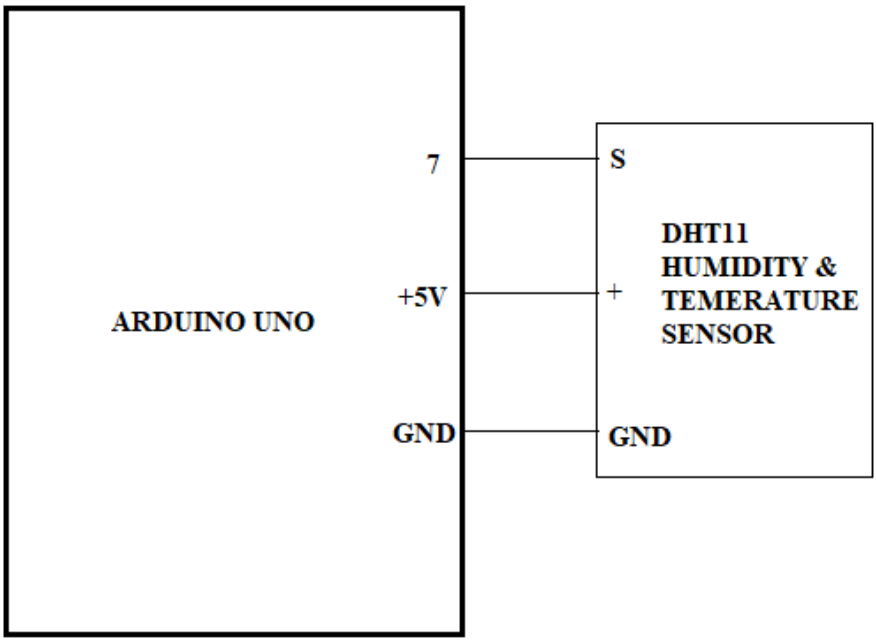
**CONNECTION DIAGRAM:**



**RESULT:**

**CIRCUIT DIAGRAM:**



**EXP. NO:** **DATE:**

## INTERFACE HUMIDITY AND TEMPERATURE SENSOR FOR AGRO METEOROLOGICAL

**AIM:**

Program to interface humidity and temperature sensor for Agro meteorological measurements using Arduino board

**COMPONENTS NEEDED:**

1. Arduino board
2. Breadboard
3. Humidity and Temperature Sensor DHT11
4. Connecting wires

**HARDWARE CONNECTIONS:**

DTH11 To Arduino

Vcc --- 5v

GND --- GND

Data Pin --- 7th Pin of Arduino

**PROCEDURE:**

- Please download the DHT library from the below link.

https://drive.google.com/file/d/0B1paTI5fzcHodno5azFOSVVDT0E/view?usp=sharing

- Go to Sketch--> Include Library --> Add Zip File

- As shown in the above screen shot please browse the ZIP file and include the library after including the library.

- Close the Arduino IDE and open it again then you will find the library included.

- Compile the program and check out the temperature and humidity in serial monitor

**PROGRAMMING:**

```
#include<dht.h>

dht DHT;

// if you require to change the pin number, Edit the pin with your arduino pin.
```

```
#define sen 7
void setup() {
Serial.begin(9600);
Serial.println("The value of temperature and humidity is"); }
void loop() {
int value ;
value= DHT.read11(sen);
Serial.println(" Humidity " );
Serial.println(DHT.humidity, 1);
Serial.println(" Temparature ");
Serial.println(DHT.temperature, 1);
delay(2000);
}
```

**RESULT:**

**EXP. NO:** **DATE:**

## CONTROL THE SERVO MOTOR BY USING MOISTURE SENSOR FOR AGRO METEOROLOGICAL MEASUREMENTS

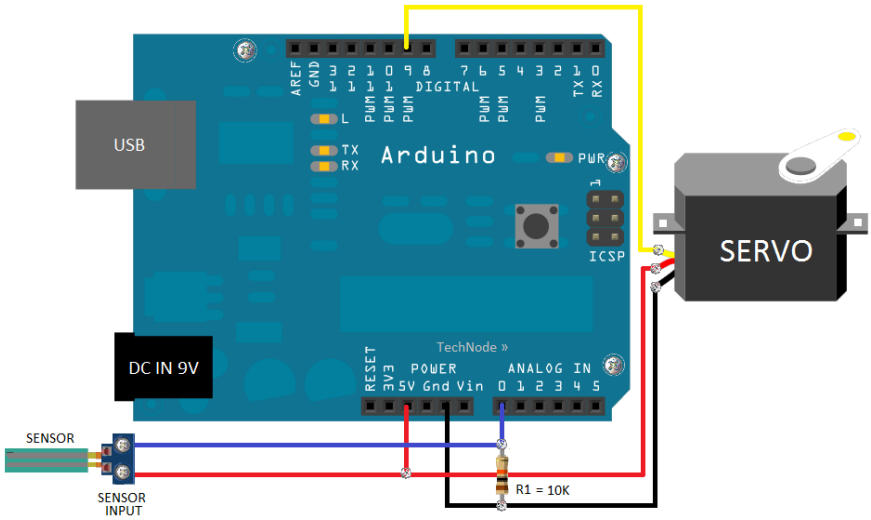
### AIM:

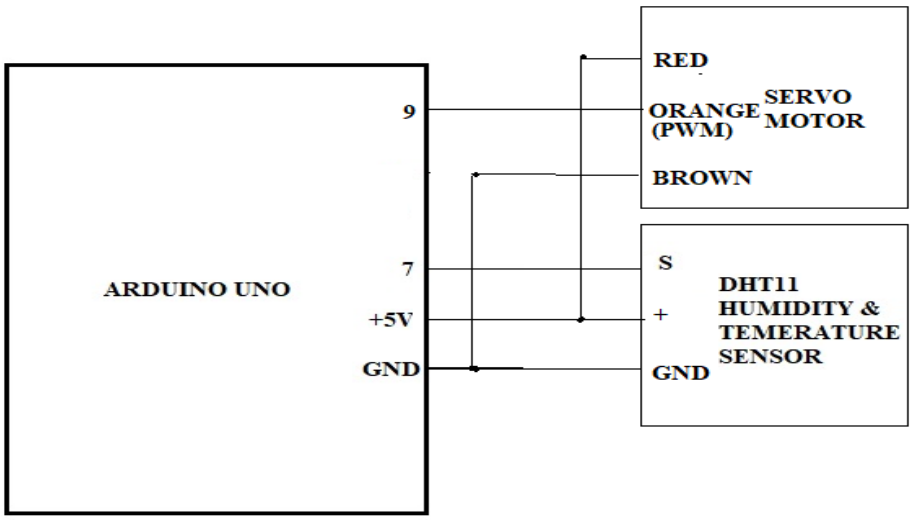
Program to control the servo motor by using moisture sensor for Agro meteorological measurements using Arduino board

### COMPONENTS NEEDED:

1. Arduino board
2. Moisture sensor
3. Servo motor
4. Connecting wires

### CIRCUIT DIAGRAM:



**HARDWARE CONNECTIONS:**

1. Moisture Sensor To Arduino

Vcc --- 5v

GND --- GND

Data Pin --- 7th Pin of Arduino

2. Servo motor To Arduino

Vcc --- 5v

GND --- GND

Pulse Pin --- 9th Pin of Arduino

**PROGRAMMING:**

```
// Include the Servo library
#include <Servo.h>
int sensor_pin = A0;
int output_value ;
int servoPin = 9;
// Create a servo object
Servo Servo1;
void setup() {
 Serial.begin(9600);
        Serial.println("Reading From the Sensor ...");
  delay(2000);
  Servo1.attach(servoPin);
}
void loop()
{
h:output_value= analogRead(sensor_pin);
  output_value = map(output_value,550,0,0,100);
  Serial.print("Mositure : ");
  Serial.print(output_value);
  Serial.println("%");
delay(1000);
if (output_value<28)
{
  // Make servo go to 0 degrees
```

```
  Servo1.write(0);
  delay(1000);
  // Make servo go to 90 degrees
  Servo1.write(90);
  delay(1000);
  // Make servo go to 180 degrees
  Servo1.write(180);
  delay(1000);
  Servo1.write(90);
  delay(1000);
}
else
goto h;
}
```

**RESULT:**

**EXP. NO:** **DATE:**

## CONTROL THE SERVO MOTOR BY USING HUMIDITY SENSOR FOR AGRO METEOROLOGICAL

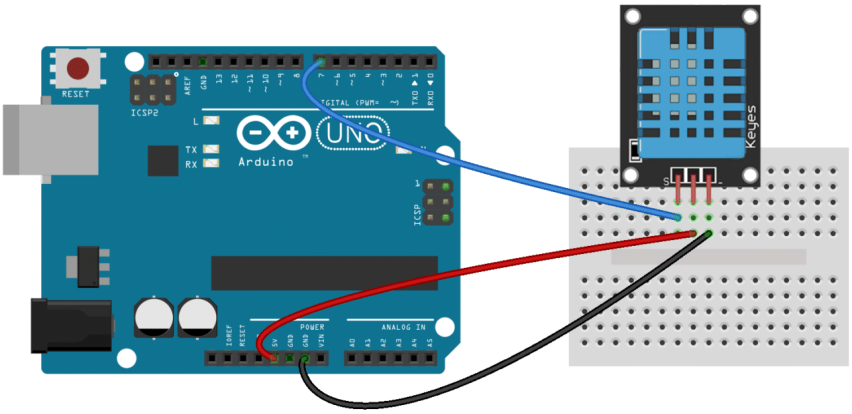
### AIM:

Program to control the servo motor by using humidity sensor for Agro meteorological measurements using Arduino board

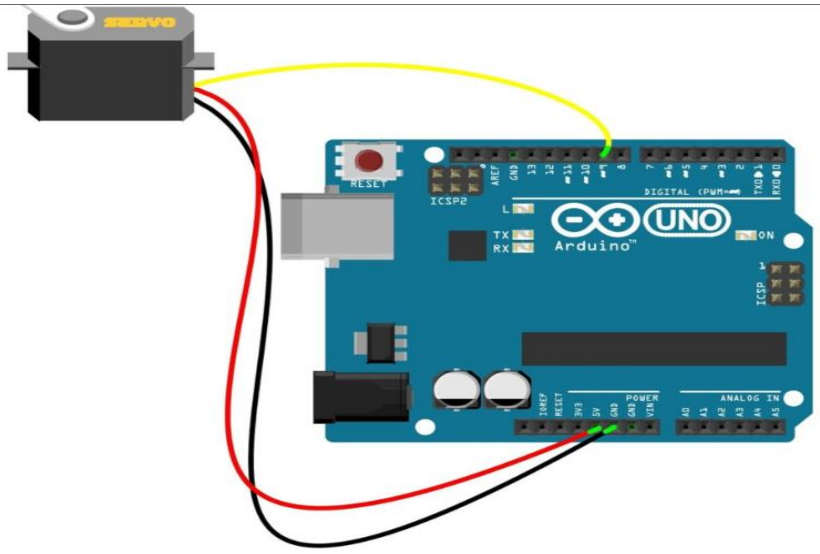
### COMPONENTS NEEDED:

1. Arduino board
2. Humidity sensor
3. Servo motor
4. Connecting wires

### CIRCUIT DIAGRAM




**Interfacing with DHT11 Sensor**




**Interfacing with servo motor**

**HARDWARE CONNECTIONS:**

1. DTH11 To Arduino

Vcc --- 5v

GND --- GND

Data Pin --- 7th Pin of Arduino

2. Servo motor To Arduino

Vcc --- 5v

GND --- GND

Pulse Pin --- 9th Pin of Arduino

**PROGRAMMING:**

```
#include <Servo.h>
// Declare the Servo pin
int servoPin = 4;
// Create a servo object
Servo Servo1;
#include<dht.h>
dht DHT;
#define DHT11_PIN 3
void setup() {
    Serial.begin(9600);
   Serial.println("Reading From the Sensor ...");
   delay(2000);
   Servo1.attach(servoPin);
Serial.begin(9600);
 }
void loop() { // READ DATA
h:int chk = DHT.read11(DHT11_PIN);
Serial.println(" Humidity " );
Serial.println(DHT.humidity, 1);
Serial.println(" Temparature ");
Serial.println(DHT.temperature, 1);
delay(2000);
```

```
if (DHT.humidity (<29)
{
  // Make servo go to 0 degrees
  Servo1.write(0);
  delay(1000);
  // Make servo go to 90 degrees
  Servo1.write(90);
  delay(1000);
  // Make servo go to 180 degrees
  Servo1.write(180);
  delay(1000);
  Servo1.write(90);
  delay(1000);
}
else
goto h;
}
```

**RESULT:**

**EXP. NO:** **DATE:**

## BIOTIC AND ABIOTIC STRESS IDENTIFICATION

**AIM:**

Program to construction of crop growth functions using MATLAB.

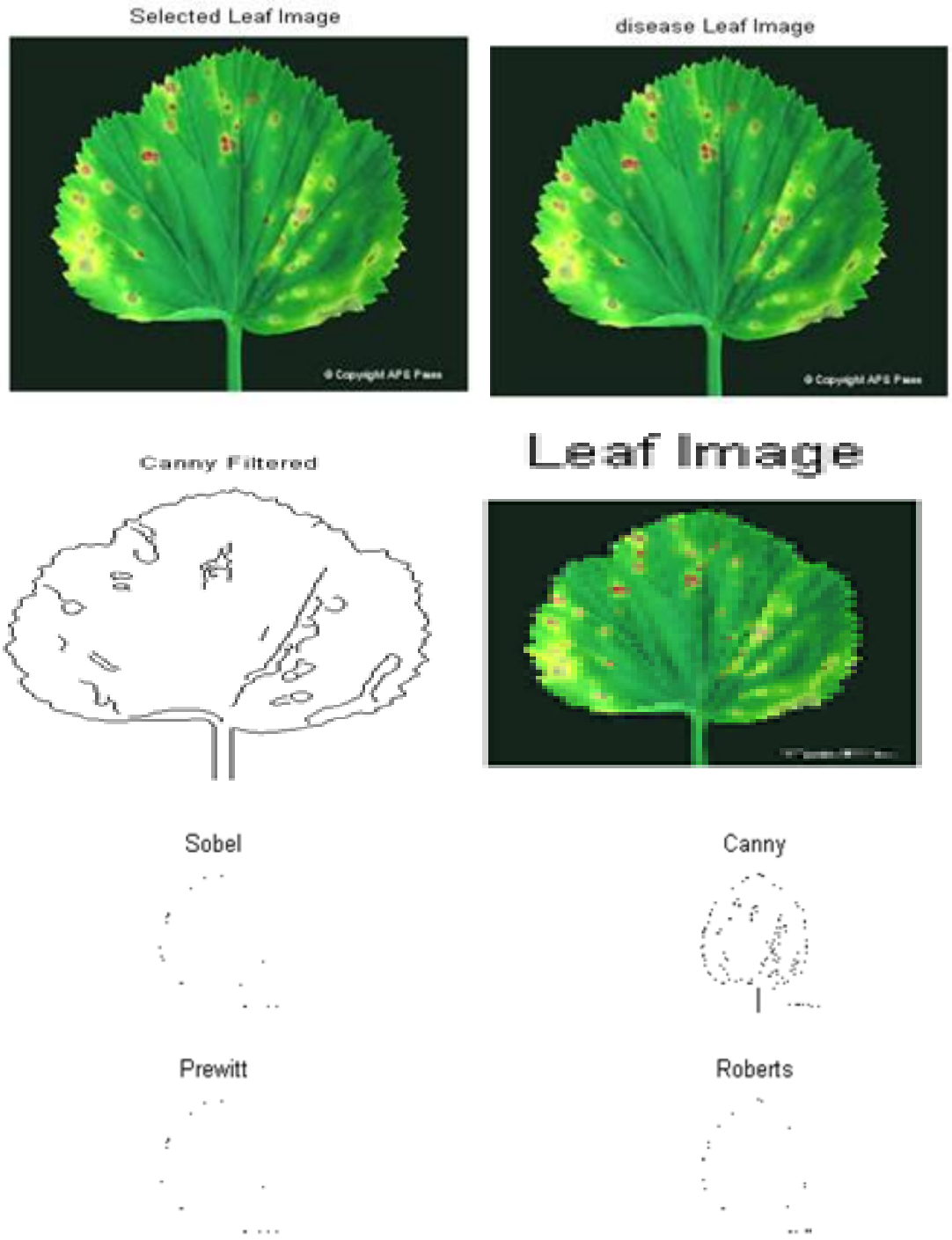
**SOFTWARE NEEDED:**

MATLAB 2011

**PROGRAM:**

```
clc
close all
clear all
[filename, pathname] = uigetfile('*.jpg','Pick a leaf Image' );
image = imread([pathname,filename]);
image = imresize(image,[256,256]);
figure, imshow(image);title(' Selected Leaf Image ');
I = rgb2gray(image);
% Apply Sobel filter with threshold 0.3
Filt_Sobel = edge(I,'sobel',0.3);
Filt_Sobel = imcomplement(Filt_Sobel);
figure, imshow(Filt_Sobel);title(' Sobel Filtered ');
% Apply Canny Edge
Filt_Canny = edge(I,'canny',0.3);
Filt_Canny = imcomplement(Filt_Canny);
figure, imshow(Filt_Canny); title(' Canny Filtered');
% Apply Prewitt Filter
Filt_Prewitt = edge(I,'prewitt',0.3);
Filt_Prewitt = imcomplement(Filt_Prewitt);
figure, imshow(Filt_Prewitt); title(' Prewitt Filtered');
% Apply Robert's Filter
Filt_Roberts = edge(I,'roberts',0.3);
Filt_Roberts = imcomplement(Filt_Roberts);
figure, imshow(Filt_Roberts); title(' Roberts Filtered');
```

```
figure,subplot(3,3,2);imshow(image);title(' Leaf Image ');
    subplot(3,3,4);imshow(Filt_Sobel);title(' Sobel ');
    subplot(3,3,6);imshow(Filt_Canny);title(' Canny ');
    subplot(3,3,7);imshow(Filt_Prewitt);title(' Prewitt ');
    subplot(3,3,9);imshow(Filt_Roberts);title(' Roberts ');
% PSNR Evaluation
im1 = double(I);
% PSNR Sobel
im2 = double(Filt_Sobel);
mse = sum((im1(:)-im2(:)).^2)/prod(size(im1));
psnr_Sobel = 10*log10(255*255/mse)
% PSNR Canny
im3 = double(Filt_Canny);
mse2 = sum((im1(:)-im3(:)).^2)/prod(size(im1));
psnr_Canny = 10*log10(255*255/mse2)
% PSNR Prewitt
im4 = double(Filt_Prewitt);
mse3 = sum((im1(:)-im4(:)).^2)/prod(size(im1));
psnr_Prewitt = 10*log10(255*255/mse3)
% PSNR Roberts
im5 = double(Filt_Roberts);
mse4 = sum((im1(:)-im5(:)).^2)/prod(size(im1));
psnr_Roberts = 10*log10(255*255/mse4)
figure, imshow(image);title(' disease Leaf Image ');
psnr_Sobel =    8.9115
psnr_Canny =    8.9087
psnr_Prewitt =    8.9115
psnr_Roberts = 8.9115
```


Selected Leaf Image
disease Leaf Image
Canny Filtered
Leaf Image
Sobel
Canny
Prewitt
Roberts

**RESULT:**

**EXP. NO:** **DATE:**

## CONSTRUCTION OF CROP GROWTH FUNCTIONS

**AIM:**

Program to construction of crop growth functions using MATLAB.

**SOFTWARE NEEDED:**

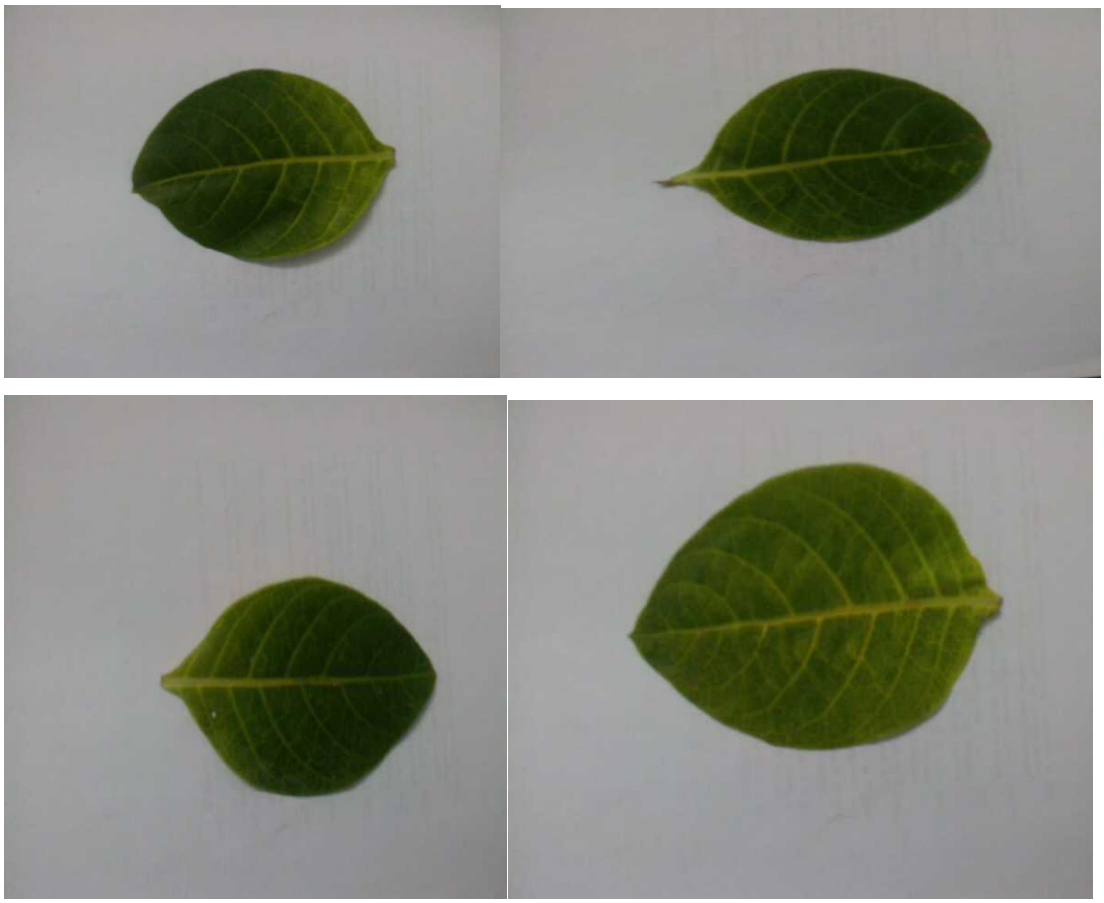
MATLAB 2011

**PROGRAM:**

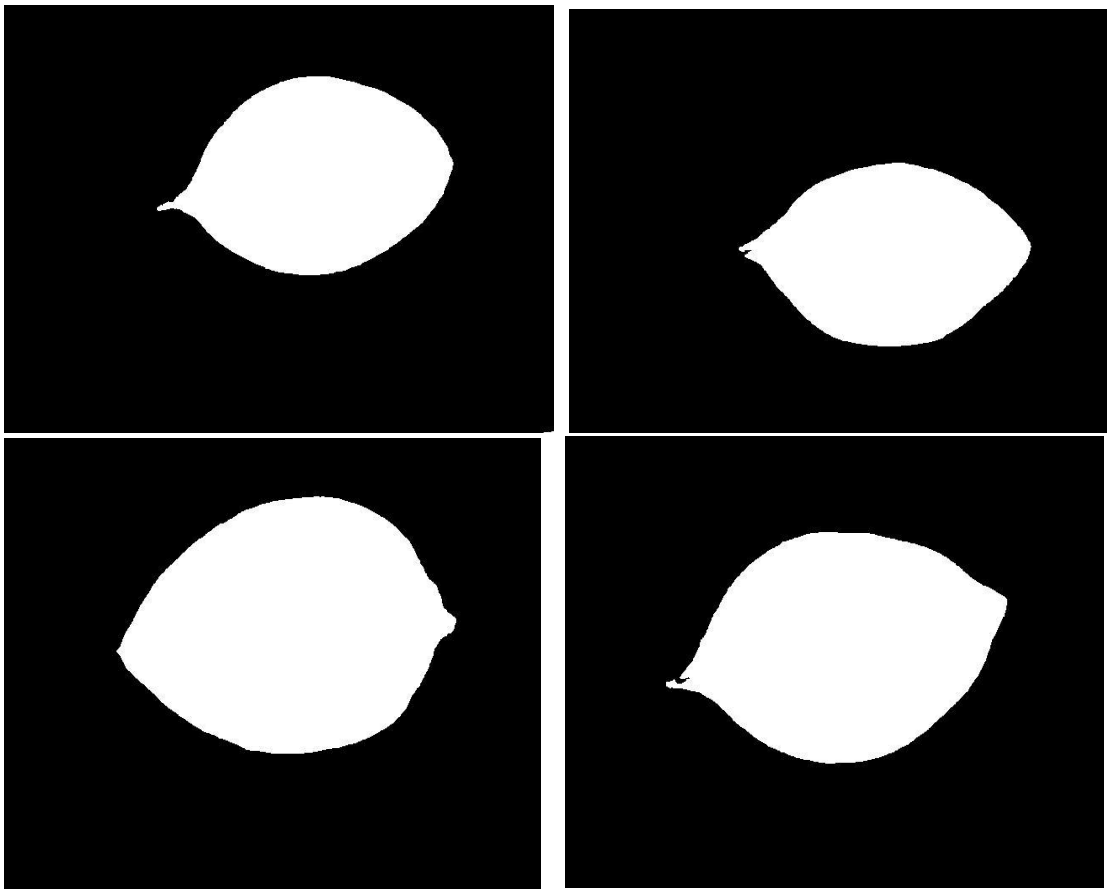
```
function training()
addpath(genpath('.\predict_leaf'));
%% edit this link
data=xlsread('.\predict_leaf\training\training.xlsx');
%%
[l br]=size(data);
obs=data(:,1:(br-1));
[r1 c1]=size(obs);
maxi=max(obs);
mini=min(obs);
for i=1:r1
   for j=1:c1
      obs(i,j)=(obs(i,j)-mini(1,j))/(maxi(1,j)-mini(1,j));
   end
end
group=data(:,br);
c=cvpartition(group,'HoldOut',0.2);
idx1=training(c);
idx2=test(c);
datatrain=obs(idx1,:);
grptrain=group(idx1,:);
testdata=obs(idx2,:);
testgrp=group(idx2,:);
X=datatrain;
Y=grptrain;
```

```
num_labels=10;
lambda = 0.01;
[all_theta] = oneVsAll(X, Y, num_labels, lambda);
pred = predictOneVsAll(all_theta, testdata);
result=pred;
count=(result==testgrp);
right=sum(count==1);
wrong=size(testgrp,1)-right;
percentage=(right/size(testgrp,1))*100
normalize=[mini;maxi];
%% edit these links
xlswrite('.\predict_leaf\predict\theta.xlsx',all_theta);
xlswrite('.\predict_leaf\predict\normalization_para.xlsx',normalize);
end
```

Input Image :



**AFTER_EXTRACTION**



**RESULT**

**EXP. NO:** **DATE:**

# EXPERIENCE WITH EXISTING OPEN SOURCE CROP SIMULATION MODELS

(Any one of the crop simulation models available as open source like DSSAT, InfoCrop, APSIM, EPIC may be demonstrated to the students to expose them crop growth simulation )

**Model**

A model is a simplified representation of a real system.

**Forms of Models:**

1. pictorial (e.g., illustrations, diagrams and flowcharts)
2. conceptual or verbal (descriptions in a natural language)
3. physical (replica of the system such as airplane model and helical, double-stranded structure of DNA molecule)
4. mathematical

**Mathematical model**

A mathematical model is one type of simplified representation of a real system and describes the system using mathematical principles in the form of an equation or a set of equations. Ex. In a controlled field experiment the maize yield (Y; kg ha-1) responds to N rate (N; kg ha-1) by the following Y = 1143+31.7N − 0.084N2 increasing N rates, maize yield increases but at diminishing returns, and a certain threshold of N the maize yield will instead start to decrease

**Crop simulation model**

A Crop Simulation Model (CSM) is a simulation model that describes processes of crop growth and development as a function of weather conditions, soil conditions, and crop management. Typically, such models estimate times that specific growth stages are attained, biomass of crop components (e.g., leaves, stems, roots and harvestable products) as they change over time, and similarly, changes in soil moisture and nutrient status

**Major types of crop simulation models**

**Mechanistic**

A mechanistic or process-based model is built from our knowledge of the physical, chemical or biological underlying process that governs the phenomenon under study. Also known as an explanatory model because it represents the cause-and-effect relationships among the variables. Ex. Newtonian physics to describe the motion of objects or Darcy's law for the flow of soil water

**Empirical model**

An empirical model (correlative or statistical model), describes the relationships among the variables but offers little or no insight to the underlying, cause-and-effect process of the phenomenon.

**Commonly used crop simulation models**

**APSIM** (http://www.apsim.info)

APSIM, the Agricultural Production Systems sIMulator is a highly advanced simulator of agricultural systems. APSIM was created by CSIRO, the State of Queensland (through its Department of Agriculture Fisheries and Forestry) and The University of Queensland in Australia.

**CROPSYST** (http://modeling.bsyse.wsu.edu/CS_Suite/cropsyst/index.html)
CropSyst, a multi-year multi-crop daily time-step crop simulation model developed by a team at Washington State University's Department of Biological Systems Engineering

**DSSAT** (Decision Support System for Agro Technology Transfer) https://dssat.net
DSSAT is a software application program that comprises crop simulation models for over 42 crops (as of Version 4.7) as well as tools to facilitate effective use of the models. The tools include database management programs for soil, weather, crop management and experimental data, utilities and application programs. The crop simulation models simulate growth, development and yield as a function of the soil-plant-atmosphere dynamics. The core development team of DSSAT currently includes a small group of scientists from the University of Florida, the International Fertilizer Development Center, and the USDA-Agricultural Research Service.

**Environmental Policy Integrated Climate (EPIC) Model**
(https://epicapex.tamu.edu/ epic/)
EPIC model is a cropping systems model that was developed to estimate soil productivity as affected by erosion as part of the Soil and Water Resources Conservation Act analysis for 1980, which revealed a significant need for improving technology for evaluating the impacts of soil erosion on soil productivity. EPIC simulates approximately eighty crops with one crop growth model using unique parameter values for each crop. It predicts effects of management decisions on soil, water, nutrient and pesticide movements, and their combined impact on soil loss, water quality, and crop yields for areas with homogeneous soils and management.

**INFOCROP**
Model developed at Indian Agricultural Research Institute, Indian Council of Agricultural Research, New Delhi. It is daily time step, multi year model and simulates around 20 crops.

**DSSAT model for demonstration**
Download the model from their web pages "https://dssat.net/downloads/dssat-v47" by providing required information. The current version is DSSAT4.7. After downloading it can be installed in windows system by clicking appropriate executables. DSSAT model can be invoked by clicking the Desktop icon which got created during installation. At the first instance you need to provide license key provided you in your email while sending the download link. The DSSAT Cropping System Model (CSM) is the most important component of DSSAT Version 4.7. It is important to understand the operation and performance of the crop models in the CSM before they can be applied to real world situations.

**RESULT**

**EXP. NO:** **DATE:**

# RUNNING A CROP MODEL IN DSSAT VERSION 4.7

## AIM

By completing this exercise, you will be able to:

- 30 Run a crop simulation model.
- 30 Use a graphics program to analyze the simulations.

## Running a crop model

Click the Desktop DSSAT icon for starting the model. From the DSSAT v4.7 shell select Crops, then expand the files under Cereals and select Maize. You will see a list of experiments (real and hypothetical) to the right of the selection tree. For example, select the experiment UFGA8201.MZX (the 11th & last). This is an experiment conducted in Gainesville, Florida in 1982. The list of the treatments that were in the experiment is listed below. In this case, there were six treatments: three water levels and two nitrogen fertilizer levels. Select all six treatments with the mouse and then select the Run button in the tool bar line. Another screen will appear with summary information of the run you are about to make. Select Run Model, and the model will run with all six treatments in BATCH mode as background job,. A small Information screen will appear indicating "Simulations are complete!". You can close the screen by clicking OK.

## Viewing the results

Select the Analysis tab from the window, and then select the OVERVIEW.OUT file, and press the View button. Scroll down in this file to see results from each treatment. One section has simulated results and observed values for model evaluation. Compare simulated and observed yields and try to explain the difference. Also look at some of the other variables. After viewing this file, you can also select outputs from individual modules to see time series simulation results. For example, the PlantGro.OUT file has simulated crop growth variables over time; SoilWat.OUT has simulated soil variables, etc.

## Graphing Outputs.

In the DSSAT47 simulation window, select the dynamic results file that you want to graph (i.e., the PlantGro.OUT, SoilWat.OUT, SoilN.OUT, etc.). You may select more than one by holding down the CTRL key while selecting the files. The output files have extensions ‘OUT’. For example, select PlantGro.OUT and push the Plot button to plot results for plant growth variables. Develop various graphs30under PlantGro Variables, plot growth stage, leaf area index (LAI), tops weight and seed/grain weight for all treatments. Also, use the Statistic option to see how much error the model has in simulating each variable for each treatment. Export the graph to a spreadsheet to view numerical results. Note: this requires Excel to be operational on your computer. Also under PlantGro Variables, plot simulated vs. observed values for seed/grain yield, LAI, and tops weight (find this capability under OPTIONS in graphics program).

For Water Variables (SoilWat.OUT), plot irrigation and total extractable soil water. For Weather Variables (Weather.OUT), plot minimum and maximum temperature, precipitation, and solar radiation. Under Nitrogen Variables (SoilN.OUT and PlantN.OUT) plot total NO3 in soil, N-leaching, organic matter N, and plant N-uptake. For evaluation of yield and yield components, as well as the main developmental stages (Evaluate.OUT), plot observed final yield versus simulated final yield and other relevant variables.

**RESULT:**

**EXP. NO:** **DATE:**

## EXPOSING CLOUD RESOURCES FOR AGRICULTURAL APPLICATIONS

(The students should be demonstrated with use of clouds for exchanging information. For example the measured data like Temperature or soil moisture can be automatically send to clouds (a local cloud can be build for this purpose or existing cloud service can be used) for access and take decisions whether to irrigate or pay attention.)

In exercise 3 you have measured the weather data like temperature using sensors. To know the actual weather measured, one has to visit the field where the sensor is installed to see actual measurements when the display is embedded along with sensor. If there is no display then you need to use laptops (or some reading device) and physically connect the board to know the weather information. However, with the help of ICT, it is possible to visualize such information from wherever you are located that too in real time. It can be done in two ways,

a. Using a GSM-GPRS modem installed along with sensor. The sensor will collect the data and GSM-GPRS modem will be made to communicate to the required system. It can send the information as SMS to your mobile phone routinely when you do not require storing of the weather data. As storing the data is essential for future, the system should send it to a server or cloud resource according to our requirement. The communicated data get stored in the server or cloud and can be visualized then and there on real time basis.

Setting up of GSM-GPRS modem is important because it can be configured to your requirement. Modem can be configured in lot of ways you want it to communicate and we need to make sure the data is transmitted and got stored. For example it can be configured so that it can keep on trying to send the data to a server or cloud till it got received by it ie till it gets feedback answer from the server or cloud resource. When the GSM modem signal is poor, in that case also it can be made to wait till the time the signal got resumed.

b. The data can also be send to cloud if the sensor recording system has Wi-Fi facility built within. Then the system may be configured through codes to send the data to the cloud. Here, we need to have an live Internet connection for the Wi-Fi to be made functional and for communicating the data to the cloud resources.

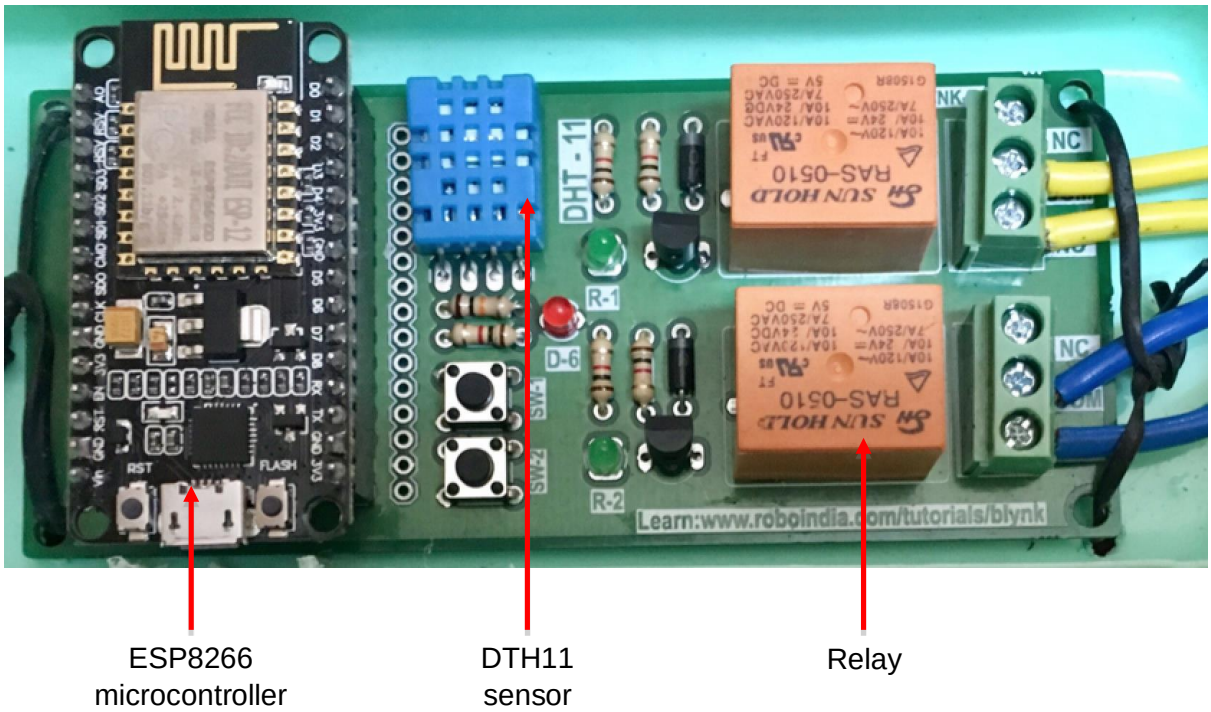
**Demonstration using NodeMCU microcontroller board**

**Description of the board**

NodeMCU board is fitted with ESP8266 which is a low-cost Wi-Fi microchip with full TCP/IP stack and micro-controller capability. The specifications are

| | |
|---|---|
| Manufacturer | Espressif Systems |
| Type | 32-bit microcontroller |
| CPU | @ 80 MHz (default) or 160 MHz |
| Memory | 32 KiB instruction, 80 KiB user data |
| Input | 16 GPIO pins |

Besides the microcontroller, the board has DHT11 temperature and Humidity sensor. It has two relays with LED bulbs, which can be used for automatic on/off of devices.First we need to fetch the temperature and RH through separate codes and store it as a variable.




ESP8266 microcontroller

DTH11 sensor

Relay

**Steps for sending weather data from NodeMCU to Cloud**

- According to requirement it can be made to read the temperature and RH for time interval like every hour or once in day etc.
- Then the data stored in the variable can be sent to the cloud using the IP details of the cloud.
- For receiving the data a file in the cloud's web server has to be configured in advance
- The data sent thus can be received by the cloud's web server file, which in turn can be made to write in to the database (MySql) already built in the Cloud resource.
- Database and browser compatibility is important and for the demonstration we use PHP and MySql, which has this type of compatibility.
- The sensor data from the NodeMCU is sent to php file in the cloud account and in turn the values are written in the MySql database.
- Through another php file from the web server the data can be fetched from the database and can be displayed through web pages on real time basis.

**Demonstration of NodeMCU**

1. Check for USB connection and open the controller and observe successful connection
2. Open “dht11.lua” and run the code for depicting the Temperature and Humidity

3. Then run “getip.lua” so that the system connected to Wi-Fi of the existing connection so that, it can communicate the weather information to the required server or cloud.
4. Run mysql.lua for sending the observed Temperature and RH via writer.php file
5. User web browser to connect to cloud for reading the data last inserted (52.74.171.62/reader.php)
6. View the writer.php and reader.php for its contents

**RESULT:**

**EXP. NO:** **DATE:**

# DEVELOPING AUTOMATED AGRO ADVISORY SYSTEMS

(Based on observed and forecasted weather information agro-advisories can be developed automatically and advisory send to the farmers as SMS or by IVR. This exercise can be demonstrated based on the exercises already completed by students in this lab)

## Prelude

Agro advisories need to be provided to the farmers in order to reduce the risks due to weather. Weather plays a major role in influencing the crop production. It has been observed that more than 50% of the crop production depended on weather prevailed during crop growth. Hence, weather based Agro Advisories gains importance in this context. The National Centre for Medium Range Weather Forecast (NCMRWF) was started during 1988 after visualizing the success of European Centre for Medium Range Weather Forecast. The NCMRWF developed Medium Range Weather Forecast of five days for each of the agro climatic zone. Now district level forecast being issued. This formed the basis for the utilization of these forecast for producing the weather based agro advisories.

## Requirements for weather based agro advisories

1. Past few days of observed weather for four to six days (Available weather information from India Meteorological Department or from Automatic Weather Station installed in Tamilnadu by Tamil Nadu Agricultural University can be used)

2. Weather for next four to six days (Medium Range Forecast) available from India Meteorological Department or from Tamilnadu Agricultural Weather Network hosted by Tamil Nadu Agricultural University can be utilized.

3. Crop stage of the crop for which advisories required

4. How this observed and forecasted weather is likely to influence the crop in question or one should have a best understanding of crop-weather relationships. For this purpose TNAU has identified 54 weather scenario (Attached separately) that is prevalent in different blocks of Tamilnadu. For these identified scenarios of observed and forecasted weather and according to different growth stages of crop, agro advisories were developed for 106 crops normally grown in Tamilnadu.

## Vulnerable crop stages identified for Rice

Rice 1 - Nursery Preparation
Rice 2 - Nursery two leaf stage
Rice 3 - Nursery two to four leaf stage
Rice 4 - Main field preparation and transplanting

Rice 5 - Transplanting and tillering
Rice 6 - Active tillering to panicle initiation
Rice 7 - Flowering
Rice 8 - Milking and dough stage
Rice 9 - Maturity and Harvest

**Example advisories**

a. Past few days weather was dry without any rainfall and the forecast for next few days also expected to be dry and if the cotton crop is at vegetative stage (30-45 days) then this kind of weather is likely to invite sucking pests in cotton crop. Then farmers need to be advised to take appropriate control measures.

b. If the forecast is indicating the chances of rainfall then the farmers can be advised to postpone the irrigation to any crop.

**Example for observed & forecasted weather and agro advisory**

**Past weather (one week)**

| | |
|---|---|
| Cloud cover | : Clear sky |
| Rainfall | : 0 mm |
| Maximum temperature | : 28.0° to 30.0°C |
| Minimum temperature | : 18.5°C to 19.7°C |
| Morning RH | : 70 to 75% |

**Weather Forecast**

| Day | Rainfall (mm) | Cloud cover (Okta) | Wind direction | Wind speed (KMPH) | Maximum Temp. (°C) | Minimum Temperature (°C) |
|---|---|---|---|---|---|---|
| 1 | 0 | 2 | 90 | 5 | 32.0 | 15.0 |
| 2 | 0 | 2 | 90 | 5 | 32.0 | 15.5 |
| 3 | 0 | 2 | 45 | 5 | 31.0 | 16.0 |
| 4 | 0 | 4 | 90 | 3 | 32.0 | 16.5 |
| 5 | 0 | 4 | 45 | 8 | 33.0 | 16.0 |

**First week of December** **Coimbatore region**

# Agro-advisory

**General**

- Due to the dry weather for next five days, irrigation to gardern land crops is to be arranged
- Stored produces may be dried using the expected dry weather

**Crop specific**

- Nursery for Navarai rice may be prepared
- Inputs may be arranged for Thai pattam maize/sorghum
- Since dry weather is expected, Puratasi pattam crops (red gram, sorghum, ragi etc.) to be harvested and dried
- Night temperature is very low. This condition retard the bud opening of rose crops. Water may be sprinkled
- Dry weather may favours for sucking pests. Watch the incidence and if needed spray may be taken up (Phosphamidon-1000ml/ Chlorpyriphos -1250 ml/Monocrotophos -1250 ml).
- Low temperature coupled with dew may favour for fungal diseases. Take appropriate measures.
- Margali pattam groundnut may sown.

**RESULT**