PROJECT TITLE: ***AUTOMATED IRRIGATION SYSTEM USING WASTE WATER***

DATE: March 13, 2022

ABSTRACT: In many parts irrigation is done manually. This requires more man power and more usage of water. This Project focuses on Developing an automatic and cost-effective Irrigation system by reusing waste water.Waste waste can be used in agriculture where there is water scarcity.

INTRODUCTION:

Here we have designed an Arduino- based drip irrigation system using various Sensors. This System allows irrigation to take place more efficiently, based on soil humidity, temperature and pH values sent to the microcontroller.

COMPONENTS USED:

1. <u>SOIL MOISTURE SENSOR:</u>

Soil moisture sensors measure the water content in the soil and can be used to estimate the amount of stored water in the soil. Soil moisture sensors do not measure water in the soil directly. Instead, they measure changes in some other soil property that is related to water content in a predictable way.

For a soil sensor to work, no matter the type, it must make contact with the soil. The highest accuracy will be obtained when the soil sensor is entirely surrounded by the soil, with no gaps between the probe and the soil.

Operating Voltage	3.3V to 5V DC

Operating Current:	15mA
Output Digital:	0V to 5V

CODE:

```
int sensorPin = A0;
int sensorValue;
int limit = 300;
void setup()
{
Serial.begin(9600);
pinMode(13, OUTPUT);
}
void loop() {
sensorValue = analogRead(sensorPin);

Serial.println("Analog Value : ");
Serial.println(sensorValue);
if (sensorValue<limit) {
digitalWrite(13, HIGH);
}
else {
digitalWrite(13, LOW);
 }
delay(1000); }
```

2. TEMPERATURE AND HUMIDITY SENSOR (DHT22):

The DHT22 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed).

Power supply	3.3-6V DC
Output signal	Digital signal via single-bus
Operating Range	Humidity 0-100%RH; Temperature -40~80 Celsius
Sensing Period Average	2 sec

CODE:

```
#include <dht.h>

#define dht_apin A0

 dht DHT;

 void setup(){

 Serial.begin(9600);

 delay(500);

 Serial.println("DHT11 Humidity & temperature Sensor\n\n");

  delay(1000);

 }//end "setup()"

void loop(){

DHT.read11(dht_apin);

Serial.print("Current humidity = ");

Serial.print(DHT.humidity);

Serial.print("%  ");

Serial.print("temperature = ");
Serial.print(DHT.temperature);

Serial.println("C  ");

 delay(5000);

}
```

3. **pH METER:**

A pH meter is a scientific instrument that measures the hydrogen-ion activity in water-based solutions, indicating its acidity or alkalinity expressed as ph. The pH meter measures the difference in electrical potential between a pH electrode and a reference electrode, and so the pH meter is sometimes referred to as a "potentiometric pH meter".

The difference in electrical potential relates to the acidity or pH of the solution.

The pH meter is used in many applications ranging from laboratory experimentation to quality control.

CODE:

```
const int analogInPin = A0;

int sensorValue = 0;

unsigned long int avgValue;

float b;

int buf[10],temp;

void setup() {

Serial.begin(9600);

}

void loop() {

 for(int i=0;i<10;i++) {

 buf[i]=analogRead(analogInPin);
```

```
  delay(100);
 }
 for(int i=0;i<9;i++){
  for(int j=i+1;j<10;j++){
   if(buf[i]>buf[j])
   {  temp=buf[i];
    buf[i]=buf[j];
    buf[j]=temp;
  } } }
 avgValue=0;
 for(int i=2;i<8;i++)
 avgValue+=buf[i];
 float pHVol=(float)avgValue*5.0/1024/6;
 float phValue = -5.70 * pHVol + 8.65;
 Serial.print("pH Value = ");
 Serial.println(phValue);
 delay(20);
}
```

4. <u>ARDUINO UNO:</u>

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

We can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software

(IDE), based on Processing. It is based on the ATMEL ATmega328p microcontroller.

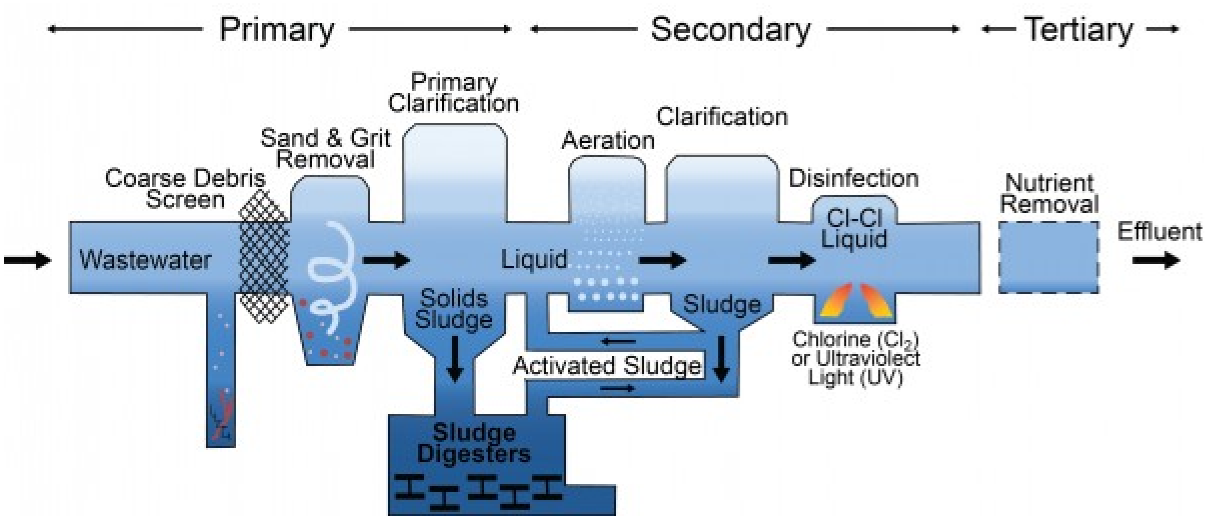
5. LCD DISPLAY:

The LCD (Liquid Crystal Display) is a type of display that uses the liquid crystals for its operation.

Here, we will accept the serial input from the computer and upload the sketch to the Arduino. The characters will be displayed on the LCD. The library that allows us to control the LCD display is called Liquid Crystal Library, which is discussed below:

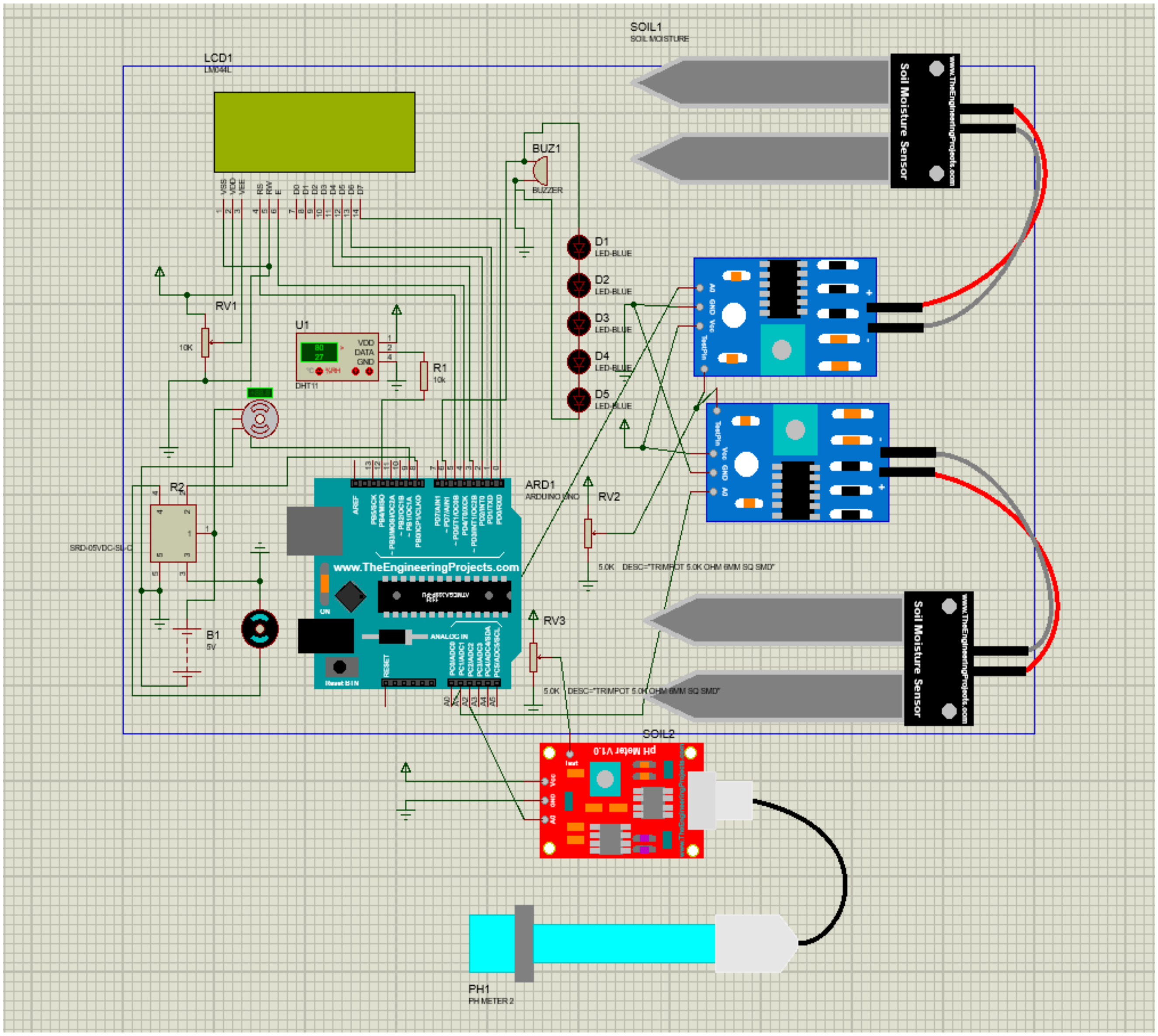
The library is declared as: #include<LiquidCrystal.h>.

6. WASTE WATER TREATMENT:

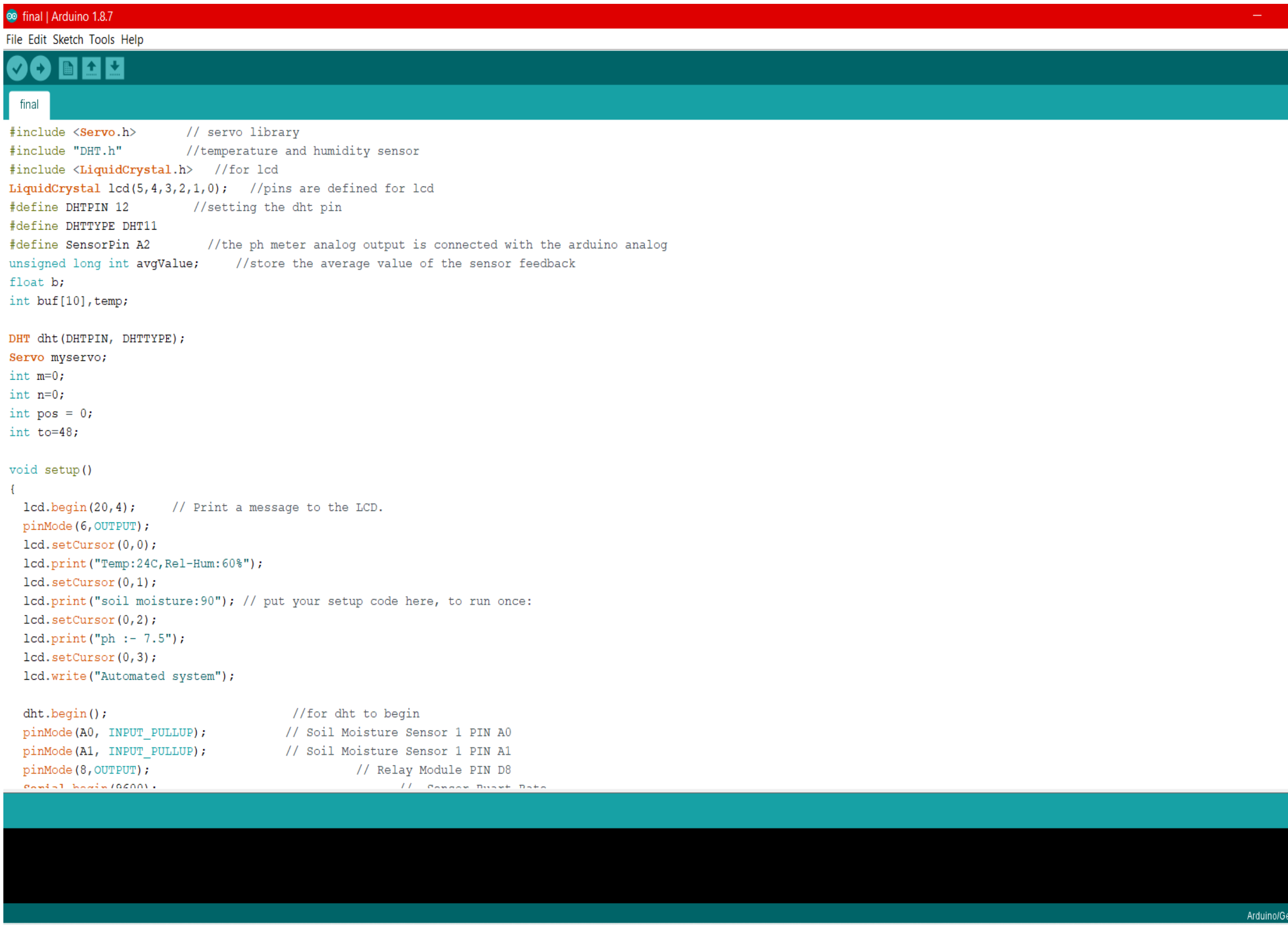


CIRCUIT DIAGRAM:

1.

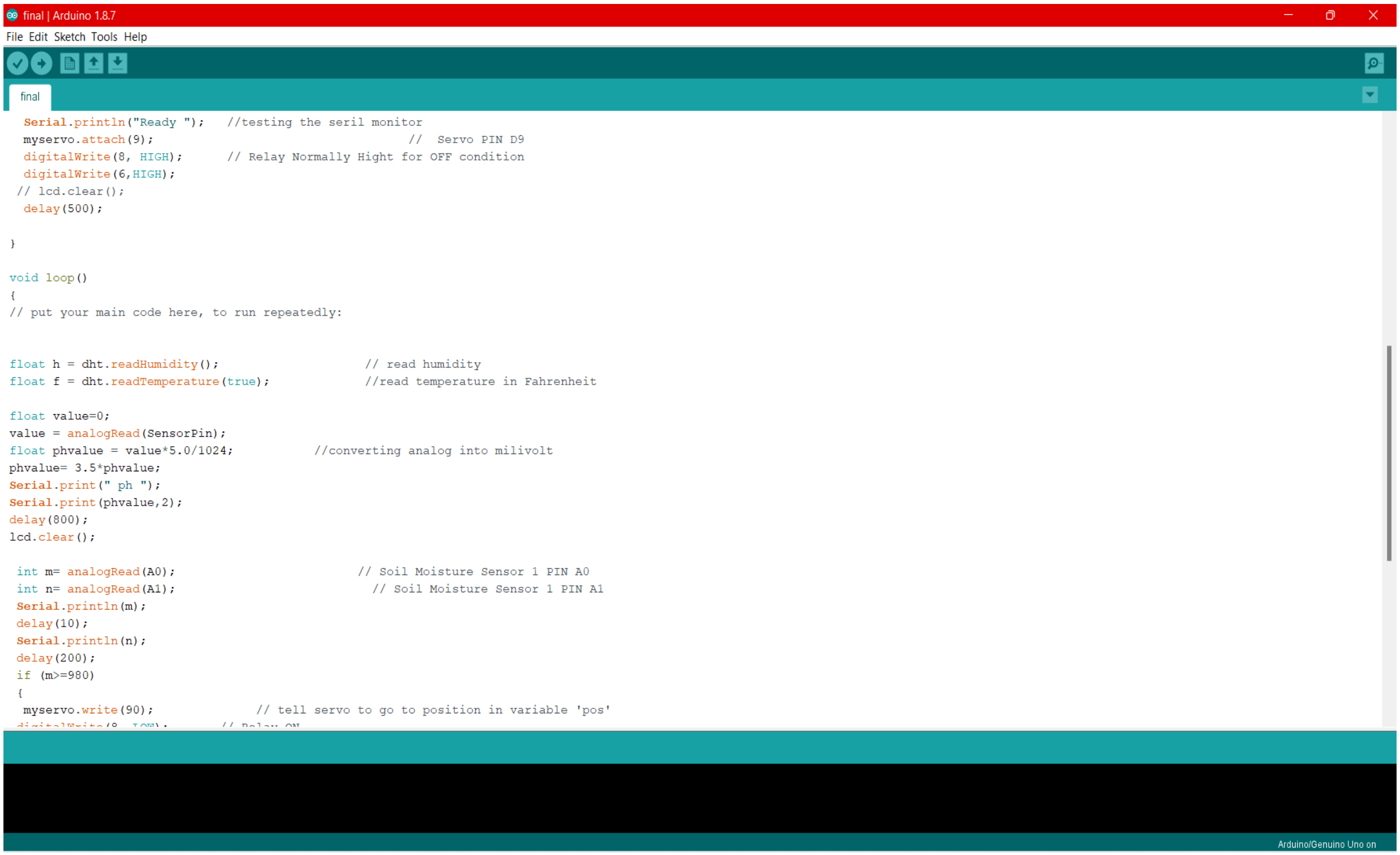


2.



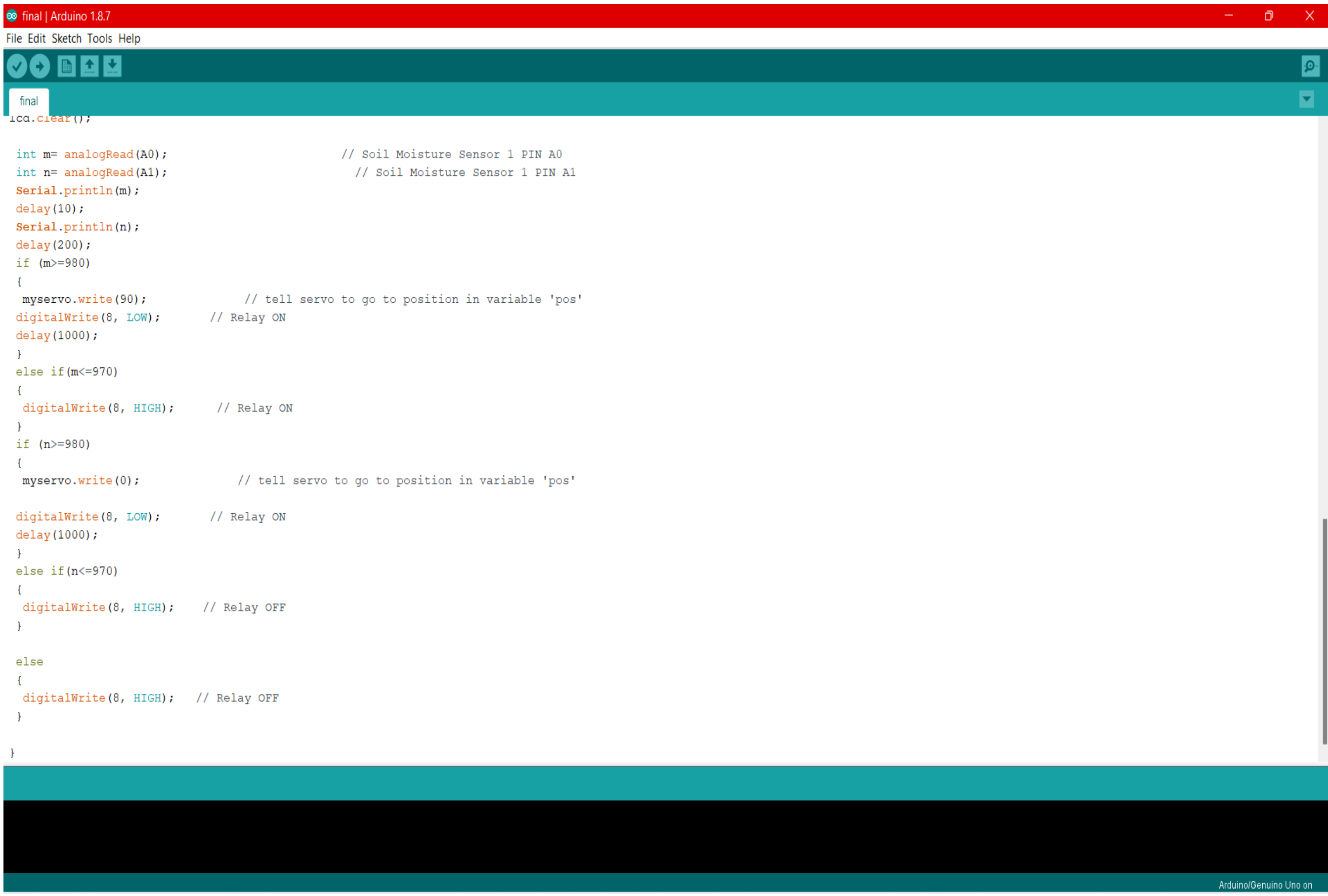
```
#include <Servo.h>       // servo library
#include "DHT.h"         //temperature and humidity sensor
#include <LiquidCrystal.h>   //for lcd
LiquidCrystal lcd(5,4,3,2,1,0);   //pins are defined for lcd
#define DHTPIN 12         //setting the dht pin
#define DHTTYPE DHT11
#define SensorPin A2        //the ph meter analog output is connected with the arduino analog
unsigned long int avgValue;     //store the average value of the sensor feedback
float b;
int buf[10],temp;

DHT dht(DHTPIN, DHTTYPE);
Servo myservo;
int m=0;
int n=0;
int pos = 0;
int to=48;

void setup()
{
  lcd.begin(20,4);     // Print a message to the LCD.
  pinMode(6,OUTPUT);
  lcd.setCursor(0,0);
  lcd.print("Temp:24C,Rel-Hum:60%");
  lcd.setCursor(0,1);
  lcd.print("soil moisture:90"); // put your setup code here, to run once:
  lcd.setCursor(0,2);
  lcd.print("ph :- 7.5");
  lcd.setCursor(0,3);
  lcd.write("Automated system");

  dht.begin();                          //for dht to begin
  pinMode(A0, INPUT_PULLUP);           // Soil Moisture Sensor 1 PIN A0
  pinMode(A1, INPUT_PULLUP);           // Soil Moisture Sensor 1 PIN A1
  pinMode(8,OUTPUT);                             // Relay Module PIN D8
  Serial.begin(9600);                                  //  Sensor Buart Rate
```

3.



```
  Serial.println("Ready ");   //testing the seril monitor
  myservo.attach(9);                               //  Servo PIN D9
  digitalWrite(8, HIGH);      // Relay Normally Hight for OFF condition
  digitalWrite(6,HIGH);
 // lcd.clear();
  delay(500);

}

void loop()
{
// put your main code here, to run repeatedly:


float h = dht.readHumidity();                    // read humidity
float f = dht.readTemperature(true);             //read temperature in Fahrenheit

float value=0;
value = analogRead(SensorPin);
float phvalue = value*5.0/1024;           //converting analog into milivolt
phvalue= 3.5*phvalue;
Serial.print(" ph ");
Serial.print(phvalue,2);
delay(800);
lcd.clear();

 int m= analogRead(A0);                         // Soil Moisture Sensor 1 PIN A0
 int n= analogRead(A1);                           // Soil Moisture Sensor 1 PIN A1
 Serial.println(m);
 delay(10);
 Serial.println(n);
 delay(200);
 if (m>=980)
 {
  myservo.write(90);               // tell servo to go to position in variable 'pos'
 digitalWrite(8, LOW);       // Relay ON
```

4.



```
lcd.clear();

 int m= analogRead(A0);                          // Soil Moisture Sensor 1 PIN A0
 int n= analogRead(A1);                            // Soil Moisture Sensor 1 PIN A1
 Serial.println(m);
 delay(10);
 Serial.println(n);
 delay(200);
 if (m>=980)
 {
  myservo.write(90);              // tell servo to go to position in variable 'pos'
 digitalWrite(8, LOW);       // Relay ON
 delay(1000);
 }
 else if(m<=970)
 {
  digitalWrite(8, HIGH);      // Relay ON
 }
 if (n>=980)
 {
  myservo.write(0);              // tell servo to go to position in variable 'pos'

 digitalWrite(8, LOW);       // Relay ON
 delay(1000);
 }
 else if(n<=970)
 {
  digitalWrite(8, HIGH);    // Relay OFF
 }

 else
 {
  digitalWrite(8, HIGH);   // Relay OFF
 }

}
```

CONCLUSION: From above we conclude that the irrigation process is done better than before to yield the proper production with limited usage of water.