VISVESVARAYA TECHNOLOGICAL UNIVERSITY

MINI PROJECT REPORT ON

"AUTOMATED HYDROPONICS NUTRITION PLANT SYSTEM"

SUBMITTED BY:

Srikakulam Hari Venkata Siva Naga Mani Sandeep **1NH18EC109**
Yeswanth Kumar **1NH18EC125**
Sai Kalyan Naidu **1NH18EC146**
Mahidhar Kumar Peddisetty **1NH18EC085**

Under the guidance of

ASSISTANT Prof. MONIKA GUPTA

Assistant Professor, Dept. of ECE, NHCE, Bengaluru.

NEW HORIZON COLLEGE OF ENGINEERING

(ISO-9001:2000 certified, Accredited by NAAC 'A', Autonomous college permanently affiliated to VTU), Outer Ring Road, Panathur Post, Near Marathalli, Bengaluru-560103.

NEW HORIZON COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the mini project work entitled "**AUTOMATED HYDRPONICS NUTRITION PLANTS SYSTEM**" carried out by **Srikakulam Hari Venkata Siva Naga Mani Sandeep (1NH18EC109) Yeswanth Kumar (1NH18EC125) ,Sai Kalyan Naidu (1NH18EC146) ,Mahidhar Kumar Peddisetty (1NH18EC085)** bonafide students of Electronics and Communication Department, New Horizon College of Engineering, Bangalore.

The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

Project Guide HOD ECE

------------------------- -------------------------

External Viva

Name of Examiner Signature with Date

1.

2.

ACKNOWLEDGEMENT

The satisfaction that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement helped us succeed.

We thank **Dr. Mohan Manghnani**, Chairman of **New Horizon Educational Institution**, for providing necessary infrastructure and creating good environment.

We also record here the constant encouragement and facilities extended to us by **Dr. Manjunatha**, Principal, NHCE and **Dr. Sanjeev Sharma**, head of the department of Electronics and Communication Engineering. We extend sincere gratitude to them.

We sincerely acknowledge the encouragement, timely help and guidance to us by our beloved guide **Assistant Prof. Monika Gupta** to complete the project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of electronics and communication department for their co-operation extended to us, who helped us directly or indirectly in this successful completion of mini project.

Srikakulam Hari Venkata Siva Naga Mani Sandeep (1NH18EC109)

Yeswanth Kumar (1NH18EC125)

Sai Kalyan Naidu (1NH18EC146)

Mahidhar Kumar Peddisetty (1NH18EC085)

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 01

INTRODUCTION

Hydroponics is a branch of hydroculture. It is the process of cultivating plants without soil. The plants receive nourishment through mineral nutrients dissolved in water, instead of soil. In my previous tutorial, I have already explained how to monitor the soil nutrients like Nitrogen, phosphorus, and Potassium using the NPK Sensor, Arduino, and an Android cell phone application.

Hydroponics

Hydroponics is basically growing plants without soil. It is a more efficient way to provide food and water to your plants. Plants don't use soil – they use the food and water that are in the soil. Soil's function is to supply plants nutrients and to anchor the plants' roots. Hydroponic gardeners may use different types of inert media to support the plants, such as rockwool, coconut fibre, river rock, Styrofoam, or clay pellets.

Hydroponics is a Latin word meaning "working water." In the absence of soil, water goes to work providing nutrients, hydration, and oxygen to plant life. From watermelons to jalapeños to orchids, plants flourish under the careful regimen of hydroponics. Using minimal space, 90% less water than traditional agriculture, and ingenious design, hydroponic gardens grow beautiful fruits and flowers in half the time. If you want to study more about the Hydroponics system then I highly recommend read the Fresh Water Systems article.

Hydroponic makes it easier to measure and fill the exact amounts of nutrients in the water solutions, since each plant requires different nutrients. This can be really a tedious job if you do it manually. Well, I am not a Hydroponics expert and I really don't know how much nutrients should be added in the water solution. But I can help you in designing and making your own Hydroponics system which you can use to measure and fill the exact amount of nutrients in the water using Solenoid valves which can be controlled as per the measured values.

CHAPTER 2

LITERATURE SURVEY

IoT based Smart Agriculture identified with remote sensor organize, scientists measured soil related parameters, for example, temperature and stickiness.Sensors were set underneath the dirt which speaks with hand-off hubs by the utilization of compelling correspondence convention giving low obligation cycle and henceforth expanding the life time of soil observing framework. The framework was created utilizing microcontroller, widespread no concurrent beneficiary transmitter (UART) interface and sensors while the transmission was finished by hourly testing also, buffering the information, transmit it and after that checking the status messages. The downsides of the framework were its cost and organization of sensor under the dirt which causes weakening of radio recurrence (RF) signals. Hydroponics or soil-less culture is an innovation for developing plants in supplement arrangements that supply every supplement component required for ideal plant development with or without the utilization of a dormant medium, for example, rock,vermiculite, Rock wool, peat greenery, saw clean, coir tidy, coconut fiber, and so forth to give mechanical support. Field Monitoring and Automation utilizing IOT in Agriculture Area proposes the upsides of having Information and Correspondence Technology (ICT)in Indian agrarian area, which demonstrates the way for provincial ranchers to supplant a portion of the ordinary systems. Observing modules are exhibited utilizing different sensors for which the data sources are nourished from Knowledge base. A model of the system is done utilizing Launch pad interconnected sensors modules with other vital electronic devices. To develop our application, we studied two groups of mobile applications for gardening. First one is the group of garden planning applications that do not support automatic control functions. The second one is those that do support automatic control functions by using sensor technology that is similar to what we planned to implement.For the first group of applications, we reviewed their functions in order to understand the requirements of the features to be developed. For example, Garden Plan Pro is a garden planning application that allows users to plan the growing area and define area requirements. This application supports users to create a year plan for their gardening. More importantly, it uses location data to suggest users what to plant and when to harvest.

A writing study likewise portrays an overview of the past existing material on a subject of the report. Exploring the writing is a basic part when composing scholarly papers that utilization look into discoveries for thoughts and focuses they attempt to make. It is likewise a necessity for the venture report.

Writing studies give brief reviews or a synopsis of the momentum inquire about on points. The structure composed requires to be such that it appeared to be consistent. It needs to sequentially speak to an advancement of the thoughts in the field that is being looked into. The length of a writing overview depends much on whether the motivation behind the undertaking report is to finish a school task or submitting for diary production. It can survey a couple of research papers on a point or be a full-length dialog on the huge work in the field until that date. A portion of the targets required recorded as a hard copy a writing overview incorporate for the comprehension on a portion of the basics of learning the definitions and ideas that will help in finding subjects that depend on past research. Writing studies are utilized in guaranteeing that the utilized examinations, philosophies and analyses offer unwavering quality and legitimacy in the exploration being led. The studies need to demonstrate basic substance evading much translation. One's sentiments and ends require to be isolated from the substance in the referred to sources. The theme of writing study must be important and limited for it to be directly to the point. It recognizes the most significant research papers from an examination on the subject. Composing the instructional exercise regarding a matter requires referring to of distinguished writing however much as could be expected. Notwithstanding, it ought not present any new outcomes but rather give a synopsis of the current condition of information regarding a matter.

Using technologies in farming is the need of the decade. There are many papers that that says how we can use the available technologies in farming in various dimensions.

CHAPTER 03

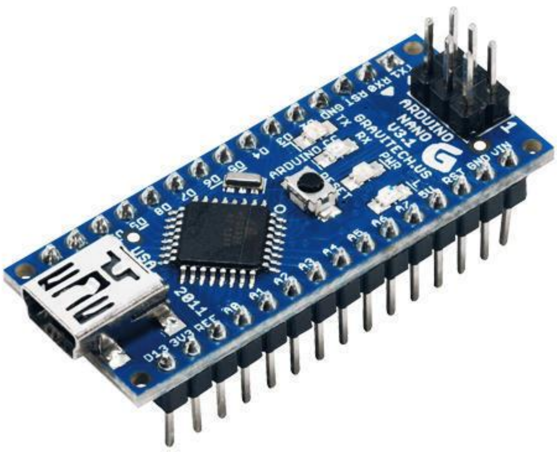
HARDWARE AND SOFTWARE SPECIFICATION

Software:

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. You can tell your board what to do by sending a set of instructions to the micro-controller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing

Hardware:

- **Arduino Nano Microcontroller:**



Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino received an Honory Mention in the Digital Communities section of the 2006 Ars Electronica Prix. Credits
Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a smallest, complete, and breadboard friendly. It has everything that Diecimila/Duemilanove has (electrically) with more analog input pins and onboard +5V AREF jumper. Physically, it is missing power jack. The Nano is automatically sense and switch to the higher potential source of power, there is no need for the power select jumper.

Nano's got the breadboard-ability of the Boarduino and the Mini+USB with smaller footprint than either, so users have more breadboard space. It's got a pin layout that works well with the Mini or the Basic Stamp (TX, RX, ATN, GND on one top, power and ground on the other). This new version 3.0 comes with ATMEGA328 which offer more programming and data memory space. It is two layers. That make it easier to hack and more affordable.

You end up paying less with Nano than Mini and USB combined.

PARAMETRICS:

Pin Category	Pin Name	Details
Power	**Vin, 3.3V, 5V, GND**	**Vin:** Input voltage to Arduino when using an external power source (6-12V). **5V:** Regulated power supply used to power microcontroller and other components on the board. **3.3V:** 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. **GND:** Ground pins.
Reset	**Reset**	Resets the microcontroller.
Analog Pins	**A0 – A7**	Used to measure analog voltage in the range of 0-5V
Input/Output Pins	**Digital Pins D0 - D13**	Can be used as input or output pins. 0V (low) and 5V (high)
Serial	Rx, **Tx**	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	**13**	To turn on the inbuilt LED.
IIC	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	**AREF**	To provide reference voltage for input voltage.

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage for Vin pin	7-12V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (2 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz
Communication	IIC, SPI, USART

OLED DISPLAY :

OLED is an abbreviation for “Organic Light Emitting Diode” which would seem to indicate that these devices are somehow related to traditional LED’s but differ in the sense that they are “organic”. This is actually true on both accounts.

The term “organic” to most people is used when describing the production of food but in this case, it has an entirely different meaning.The “organic” in OLED refers to the organic molecules used in creating these devices. Organic molecules are simply molecules that are based around lines or rings of carbon atoms. Examples of organic molecules include common items such as sugar, gasoline, alcohol, wood, and plastics.OLED displays typically come with either an SPI or an I2C interface. Some even have both interfaces, one of the ones we will look at today works like that.

To display data on an OLED you will first load it into the OLED buffer and then give it a command to write to the display. There are several libraries available for the Arduino that will simplify this, I’ll be showing you one from Adafruit that is very versatile.

As the OLED display is really a matrix of OLEDs you’ll need to address them individually to control them. The Adafruit library simplifies this, as do the other OLED libraries.

You can use font files to display characters, your library may also have these built-in.

PH SENSOR :

pH scale is used to measure the acidity and basicity of a liquid. It can have readings ranging from 1-14 where 1 shows the most acidic liquid and 14 shows the most basic liquid. 7 pH is for neutral substances that are neither acidic nor basic. Now, pH plays a very important role in our lives and it is used in various applications. For example, it can be used in a swimming pool to check the quality of water. Similarly, pH measurement is used in a wide variety of applications like agriculture, wastewater treatment, industries, environmental monitoring, etc.

In this project, we are going to make an Arduino pH Meter and learn how to measure the pH of a liquid solution using a gravity pH sensor and Arduino. A 16x2 LCD is used to show the pH value on the screen. We will also learn how to calibrate the pH sensor to determine the accuracy of the sensor. So let's get started.

The unit that we use to measure the acidity of a substance is called pH. The term "H" is defined as the negative log of the hydrogen ion concentration. The range of pH can have values from 0 to 14. A pH value of 7 is neutral, as pure water has a pH value of exactly 7. Values lower than 7 are acidic and values greater than 7 are basic or alkaline.

Analog pH sensor is designed to measure the pH value of a solution and show the acidity or alkalinity of the substance. It is commonly used in various applications such as agriculture, wastewater treatment, industries, environmental monitoring, etc. The module has an on-board voltage regulator chip which supports the wide voltage supply of 3.3-5.5V DC, which is compatible with 5V and 3.3V of any control board like Arduino. The output signal is being filtered by hardware low jitter.

TDS METER :

Total dissolved solids (TDS) are measured in milligrams per unit volume of water (mg / l) or also referred to as parts per million (ppm). For drinking water, the maximum concentration level established by the EPA (the United States Environmental Protection Agency) is 500 mg / L. So TDS value is used as a reference in knowing the quality of water for drinking, Aquaponics, hydroponics and where the quality of water matters.

So taking this into consideration we are going to build an IOT TDS monitoring device with the help of Arduino with Gravity Analog TDS sensor which is designed for Arduino with simple plug and play use. As TDS is dependent on temperature we use a waterproof temperature sensor bundled with the TDS sensor probe.

TDS (Total dissolved solids), are inorganic compounds found in water, such as salts, heavy metals, and some traces of organic compounds that dissolve in water. Some of these compounds or substances can be essential for life, however, it can be harmful when taking more than the desired amount that the body needs.

This sensor is specially designed to be compatible with arduino for measuring water for drinking, aquaponics and more. it operates between input voltages 3.3v to 5v. So it can be uses with multiple compatible MCU's available, like Arduino, ESP32, Nodemcu, and more. It outputs an analog voltage from 0 to 2.3v. As the DC signal can add polarization errors to the value and corrode the probe pins used, this sensor uses an AC signal because of no polarity to get better accuracy and for long term usage of the waterproof TDS Probe. The probe has 2 metallic pins through which the electrical conductivity is calculated.
This is a very simple circuit among the other two just connect the electrode probe with the sensor chip on one side where the other side has 3 pins where one goes to 5v power, second goes to GND, and third goes to Analog pin on arduino A1. And we connect the temperature sensor to digital pin D7 with the help of a 4.7k ohm resistor.The TDS in water value as you can see on the LCD screen indicates how many milligrams of soluble solids are dissolved in one liter of water. Many TDS meters display the TDS value in ppm which stands for parts per million. In general, the higher the TDS value, the more soluble solids are dissolved in water, and the less clean the water is. Right now the water under test is excellent as the TDS value is less than 300.

DS18B20 DIGITAL TEMPERATURE SENSOR:

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements. It uses a 1 wire bus to communicate.Each device has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus.

It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C ove the range of -10°C to +85°C.The DS18B20 is available in a discrete component form or in a pre wired sealed or water proofed package. They are readily available from many suppliers.

In this hookup we are only connecting one device to the Arduino. We also have a hookup guide for connecting several DS18B20 to the Arduino board.

The DS18B20 can be wired up using 2 or three wires. Using three wires the DS18B20 is powered using the normal Vdd and GND pins. This is the preferred option. We can also use what is called Parasitic power which is where we get power off the DATA pin. Here we use only two wires. The Vdd is connected to GND. We are not using this power mode as it more complicated to implement.

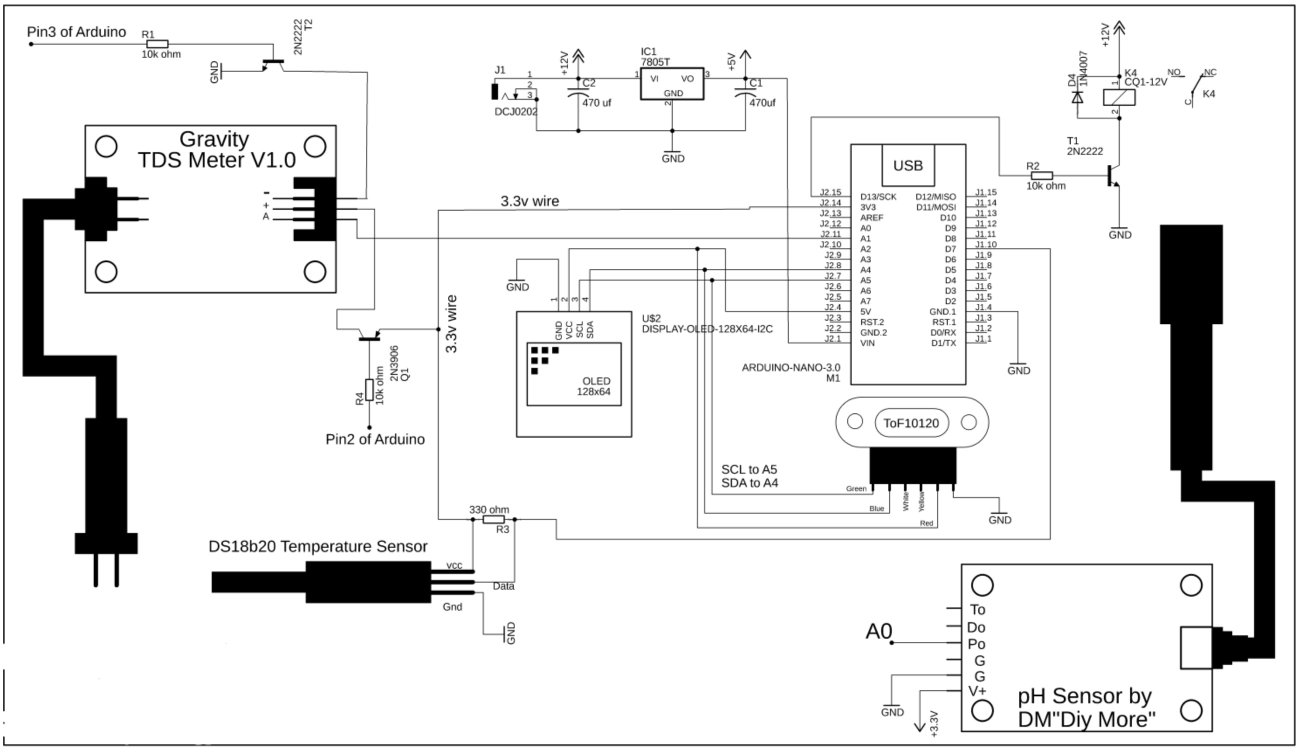
The bus master writes data to the DS18B20 during write time slots and reads data from the DS18B20 during read time slots. One bit of data is transmitted over the 1-Wire bus per time slot.The DS18B20 samples the 1-Wire bus during a window that lasts from 15µs to 60µs after the master initiates the write time slot. If the bus is high during the sampling wins written to the DS18B20. If the line is low, a 0 is written to the DS18B20.

TOF10120 RANGE FINDER :

TOF10120 range sensor provides accurate and repeatable long range distance measurement for high-speed autofocus (AF). The innovative time-of-flight technology allows performance independent of
object reflectance. TOF10120's time-of-flight sensing technology is realized by Sharp's original SPAD (Single Photon Avalanche Diodes) using low-cost standard CMOS process. It enables accurate ranging result, higher immunity to ambient light and better robustness to cover-glass optical cross-talk by special optical package design.
The module has 6 pins (GND-VCC) then (RX-TX) for UART and (SDA-SCL) for i²c, in this tutorial we gonna use the i²c interface to wire it with Arduino also I'll add a LCD i²c screen to see the measure in (mm) you can convert it if you want just simple calculations

CHAPTER 04

PROPOSED SYSTEM



CIRCUIT DIAGRAM

Connect the + pin of the TDS meter with 3.3V. Connect the Minus pin with the GND of the Arduino, and finally, connect the A pin of the TDS or EC sensor with the A1 pin of the Arduino.

Connect the VCC wire with the 3.3V of the Arduino. Connect the ground of the DS18b20 with the ground of the Arduino, and finally, connect the Data wire with the digital pin D7 of the Arduino.

For performing my initial tests I started with the Well water. First I started with the DS18b20 temperature sensor which is completely waterproofed and can be dipped in liquids without any problem. I have been using this sensor with Arduino, ESP32, ESP8266, and Raspberry Pi. This is seriously an amazing temperature sensor and as it's a waterproofed temperature sensor for this project it's just perfect. Anyhow after I immersed the Temperature sensor, the value on the Oled display module started to increase. For the practical demonstration watch video tutorial given at the end of this article. So the temperature sensor was working just fine.After adding the TDS meter although I was able to read the EC value but it really affected the pH value, as you can see on the display the pH value is changed, this is the kind of problem that most of the guys are complaining about that when we add the pH sensor and TDS meter in the same liquid container the pH value is greatly disturbed. This is a very serious kind of problem as we are not getting the actual pH valuethis problem needs to be solved before we can make an efficient Hydroponics system. Anyways, I continued with my test, to further ensure, that the TDS meter is creating this problem, I removed the TDS probe, and then in just 3 to 4 seconds, the pH value got normal. Now I know exactly the pH sensor behaves in an abnormal way when the TDS Meter or the EC sensor is added in the same container. So, I decided to check this one more time, I dipped the TDS meter probe in the water, waited for around 3 to 4 seconds and the pH value jumped. So, from this initial test, we know that when the EC sensor is not dipped in the same container we get pretty stable pH values, and when we add the EC sensor then the pH value is greatly disturbed.

To solve this problem we will need to isolate the power wires of the EC sensor. I will use two transistors to control the EC Sensor, so the EC sensor will remain OFF during the times when Arduino will be reading the pH Sensor. After Arduino reads the pH Sensor, then again the EC sensor will be turned ON. This is the next test that I am going to perform and let's see if the problem will be solved.

To isolate the voltage and GND wires of the TDS Sensor I am using these two general purpose PNP and NPN Transistors. The one on the right side is the 2N3906 PNP type transistor. The emitter is connected with 3.3 volts which goes to the TDS meter through the collector of the transistor. So 3.3V to the TDS sensor are supplied by turning on this transistor. The other transistor is the 2n2222 NPN transistor and I am using this transistor to control the GND. The emitter is connected with the Arduino's GND while the collector of the 2n2222 NPN transistor is connected with the GND of the TDS meter. While the base of the transistor is connected with the digital pin 3 of the Arduino. I will explain the final connections in the circuit diagram after I am done with all the testing.

Program Code:

```
#include <Wire.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <SimpleTimer.h>


SimpleTimer timer;

float calibration_value = 21.34 + .06; //
int phval = 0;
unsigned long int avgval;
int buffer_arr[10],temp;

float ph_act;
// for the OLED display

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);


namespace pin {
const byte tds_sensor = A1;
const byte one_wire_bus = 7; // Dallas Temperature Sensor
}

namespace device {
float aref = 4.3;
}

namespace sensor {
float ec = 0;
unsigned int tds = 0;
float waterTemp = 0;
```

```
float ecCalibration = 1;
}

OneWire oneWire(pin::one_wire_bus);
DallasTemperature dallasTemperature(&oneWire);

// EC isolator

// EC isolator
int EC_Isolator = 2; // 3906 PNP TYPE TRANSISTOR THIS is used to connect and disconnect the 3.3V wire
int EC_GND_Wire = 3; // 2N2222 NPN TRANSISTOR THIS IS USED TO CONNECT AND DISCONNECT THE GND WIRE

// for the TOF10120 distance range sensor
unsigned char ok_flag;
unsigned char fail_flag;

unsigned short lenth_val = 0;
unsigned char i2c_rx_buf[16];
unsigned char dirsend_flag=0;

int x_mm; // distance in millimeters
float y_inches; // distance in inches

// to control pump
int relay = 9;
int relay_flag = 0;

void setup()
{
  Wire.begin();
 Serial.begin(9600);
  dallasTemperature.begin();
    pinMode(relay, OUTPUT);
  digitalWrite(relay, LOW);
  printf_begin();
pinMode(EC_Isolator, OUTPUT);
  pinMode(EC_GND_Wire, OUTPUT);
  digitalWrite(EC_Isolator, HIGH);
  digitalWrite(EC_GND_Wire, LOW);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setTextColor(WHITE);


timer.setInterval(500L, display_pHValue);
}
void loop() {
timer.run(); // Initiates SimpleTimer
```

```
tofsensor();



digitalWrite(EC_Isolator,HIGH);
digitalWrite(EC_GND_Wire, LOW);
ph_sensor();
digitalWrite(EC_Isolator,LOW);
digitalWrite(EC_GND_Wire, HIGH);
delay(1000);
 dallasTemperature.requestTemperatures();
  sensor::waterTemp = dallasTemperature.getTempCByIndex(0);
  float rawEc = analogRead(pin::tds_sensor) * device::aref / 1024.0; // read the analog value more stable by the median filtering algorithm, and convert to voltage value
  float temperatureCoefficient = 1.0 + 0.02 * (sensor::waterTemp - 25.0); // temperature compensation formula: fFinalResult(25^C) = fFinalResult(current)/(1.0+0.02*(fTP-25.0));
  sensor::ec = (rawEc / temperatureCoefficient) * sensor::ecCalibration; // temperature and calibration compensation
  sensor::tds = (133.42 * pow(sensor::ec, 3) - 255.86 * sensor::ec * sensor::ec + 857.39 * sensor::ec) * 3.3; //convert voltage value to tds value
  Serial.print(F("TDS:")); Serial.println(sensor::tds);
  Serial.print(F("EC:")); Serial.println(sensor::ec, 2);
  Serial.print(F("Temperature:")); Serial.println(sensor::waterTemp,2);
  delay(1000);



}

void display_pHValue()
{
     // display on Oled display

   // Oled display
  display.clearDisplay();
  display.setTextSize(2);
  display.setCursor(0,0); // column row
  display.print("pH:");

  display.setTextSize(2);
  display.setCursor(55, 0);
  display.print(ph_act);


    display.setTextSize(2);
  display.setCursor(0,20);
  display.print("Temp:");
```

```
 display.setTextSize(2);
 display.setCursor(60, 20);
 display.print(sensor::waterTemp);


   display.setTextSize(2);
 display.setCursor(0,40);
 display.print("EC:");

 display.setTextSize(2);
 display.setCursor(60, 40);
 display.print(sensor::ec);



display.display();
}

void ph_sensor()
{
 for(int i=0;i<10;i++)
 {
 buffer_arr[i]=analogRead(A0);
 delay(30);
 }
 for(int i=0;i<9;i++)
 {
 for(int j=i+1;j<10;j++)
 {
 if(buffer_arr[i]>buffer_arr[j])
 {
 temp=buffer_arr[i];
 buffer_arr[i]=buffer_arr[j];
 buffer_arr[j]=temp;
 }
 }
 }
 avgval=0;
 for(int i=2;i<8;i++)
 avgval+=buffer_arr[i];
 float volt=(float)avgval*3.3/1024/6; // the original was float volt=(float)avgval*5.0/1024/6; when its
connected with arduino's 5v
  ph_act = -5.70 * volt + calibration_value;

 Serial.println("pH Val: ");
 Serial.print(ph_act);
 delay(1000);
}
```

```
int serial_putc( char c, struct __file * )
{
  Serial.write( c );
  return c;
}

void printf_begin(void)
{
  fdevopen( &serial_putc, 0 );
}



void SensorRead(unsigned char addr,unsigned char* datbuf,unsigned char cnt)
{
  unsigned short result=0;
  // step 1: instruct sensor to read echoes
  Wire.beginTransmission(82); // transmit to device #82 (0x52), you can also find this address using
the i2c_scanner code, which is available on electroniclinic.com
  // the address specified in the datasheet is 164 (0xa4)
  // but i2c adressing uses the high 7 bits so it's 82
  Wire.write(byte(addr));      // sets distance data address (addr)
  Wire.endTransmission();      // stop transmitting
  // step 2: wait for readings to happen
  delay(1);                   // datasheet suggests at least 30uS
  // step 3: request reading from sensor
  Wire.requestFrom(82, cnt);    // request cnt bytes from slave device #82 (0x52)
  // step 5: receive reading from sensor
  if (cnt <= Wire.available()) { // if two bytes were received
    *datbuf++ = Wire.read();  // receive high byte (overwrites previous reading)
    *datbuf++ = Wire.read(); // receive low byte as lower 8 bits
  }
}

int ReadDistance(){
    SensorRead(0x00,i2c_rx_buf,2);
    lenth_val=i2c_rx_buf[0];
    lenth_val=lenth_val<<8;
    lenth_val|=i2c_rx_buf[1];
    delay(300);
    return lenth_val;
}

void tofsensor()
{
      x_mm = ReadDistance();
   Serial.print(x_mm);
   Serial.println(" mm");
```

```
  // You can convert millimeters to inches in one of two ways: divide the number of millimeters by
25.4, or multiply the number of millimeters by 0.0394
  y_inches = x_mm * 0.0394;
  Serial.print(y_inches);
  Serial.println(" inches");

  if( (y_inches > 10 ) && (relay_flag == 0))
{
 digitalWrite(relay, LOW);
 relay_flag = 1;
}

if( (y_inches <= 5 ) && (relay_flag == 1))
{
 digitalWrite(relay, HIGH);
 relay_flag = 0;
}
}
```

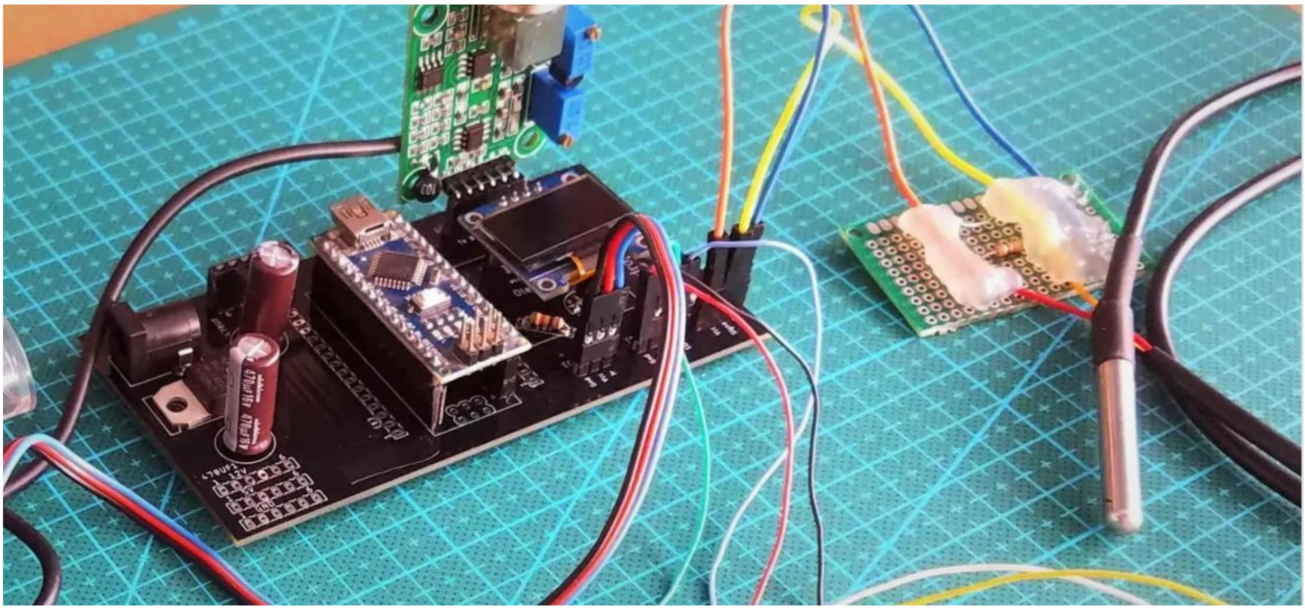
CHAPTER 05

ADVANTAGES AND APPLICATION

- Optimal Use of Location.
- Complete Control Over Climate
- Production increases 3 to 10 times in the same amount of space.
- pH and EC of the nutrients is maintained using this system.
- There is no need to go to the farm to measure pH & EC.
- The meters provide numerical value of the pH directly.
- pH meters are very accurate and provide exact pH value with the help of pH sensors
- It helps in determining how much acidic or basic any substance is.

CHAPTER 06

RESULT AND DISCUSSIONS



CHAPTER 07

FUTURE SCOPE

The project will be centered around the creation of a small scale automated hydroponics system with the use of mechatronics. The system itself shall be designed to house and maintain six medium sized sweet basil plants. Sweet basil will be suitable for this research because it is easy to maintain and
grows fast compared to other plants .
The system is intended to keep track of following parameters:

- pH concentration
- Nutrient solution concentration

If the system recognizes that the concentration of pH and/or nutrient level is not optimal, it will automatically compensate by adding more of either nutrient- and/or pH solution to the water solvent.
The system will be placed indoors by a window. This will give the system sufficient sunlight and ambient temperature, hence an external heater and light source will not be included in the system for regulation.
An air pump will be added to the water solution. This will provide the water solvent with a constant rate of oxygen. The air pump will not be directly regulated.
The applications can be made in regional languages with voice input and output to make them farmer friendly. For automating maintenance software's should be customized as per the individual’s requirement.