BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 318 (January 2023)

Control System Laboratory

Final Project Report

Section: C2 Group: 08

Greenhouse Monitoring & Controlling System

Course Instructors:

Dr. Pran Kanai Saha, Professor
Azazul Islam, Part-Time Lecturer

Signature of Instructor: ____

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature. ***You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.***

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: _Nayem___________________________ **Full Name: Md. Nayem Hasan** **Student ID: 1906175**	**Signature:** __Tanvir__________________________ **Full Name: Tanvir Rahman** **Student ID: 1906176**
Signature: ____Samiul________________________ **Full Name: Samiul Hossain** **Student ID: 1906179**	**Signature:** __Tousif__________________________ **Full Name: Tousif Ansari** **Student ID: 1906190**

Table of Contents

1 Abstract

For improved plant growth, there are certain essential aspects that come into play such as temperature, humidity, light, and water. For this project we create a Greenhouse monitoring and controlling system which is totally autonomous – the Arduino will be programmed to switch the appliances on and off based on temperature, soil moisture, light intensity and humidity measurements here we use Node MCU to keep information on climatic factors. This project report presents the design, development, and implementation of a greenhouse controlling system using Arduino microcontrollers. The system's primary objective is to monitor and regulate four vital parameters crucial for optimal plant growth: temperature, humidity, soil moisture, and light intensity. Through an array of sensors and control components, the Arduino microcontroller collects real-time data and executes control strategies. This includes activating a light bulb for light intensity adjustment, a cooling system with a fan for temperature control, a nichrome coil for humidity management, and a heating coil for temperature elevation. By maintaining precise environmental conditions within the greenhouse, this project aims to significantly enhance crop yield and resource efficiency. The report delves into the system's hardware and software implementation, control algorithms, performance evaluation, and potential applications. Ultimately, this greenhouse controlling system represents a promising step towards sustainable and automated agriculture practices.

2 Introduction

The world's growing population places an increasing demand on agricultural production. In the quest to meet this demand sustainably, controlled environment agriculture (CEA) has emerged as a promising solution. Greenhouses, in particular, offer an enclosed and controlled environment that can optimize crop growth, minimize resource consumption, and mitigate the impact of unpredictable weather conditions. Within this context, our project focuses on the design and development of a greenhouse controlling system powered by Arduino microcontrollers.

3 Design

3.1 Problem Formulation

3.1.1 Identification of Scope

In this project report, we will focus on the development and implementation of a greenhouse controlling system using Arduino microcontrollers. The primary objectives include the monitoring and control of four critical parameters within the greenhouse environment: temperature, humidity, soil moisture, and light intensity. The system's core functionality involves real-time data acquisition from various sensors, such as temperature and humidity sensors, soil moisture sensors, and light intensity sensors. Upon data acquisition, the Arduino microcontroller will execute control strategies to regulate the greenhouse conditions. This control will encompass actions like activating a bulb for light intensity control, a cooler with a fan for temperature regulation, a nichrome coil for humidity adjustment, and a heating coil for temperature increment, all managed through relays. The project's significance lies in its potential to optimize and automate greenhouse conditions, thereby enhancing crop yield and resource efficiency. This scope identification sets the stage for a detailed exploration of the system's design, hardware and software implementation, control strategies, and performance evaluation in the subsequent sections of the report.

3.1.2 Literature Review

The literature review for this project underscores the significance of greenhouse automation systems in modern agriculture. It highlights the evolution of controlled environment agriculture (CEA) and the pivotal role of greenhouse automation in optimizing crop growth. Key findings from existing research emphasize the benefits of automating parameters like temperature, humidity, soil moisture, and light intensity for increased crop yields, resource efficiency, and environmental sustainability. However, it also identifies challenges, such as system complexity and cost, which need to be addressed. The literature review serves as the foundation for the development and evaluation of the greenhouse controlling system presented in this report.

3.1.3 Formulation of Problem

The challenge at hand is the inefficiency of manual greenhouse management, resulting in resource wastage, limited crop yields, labor-intensive operations, vulnerability to climate variability, and adverse environmental impacts. To address these issues, we need an automated greenhouse control system capable of real-time monitoring and adjustment of key parameters, promoting sustainable and efficient agriculture practices.

4.Analysis

Data Assessment:
Evaluating sensor data for trends and variations.

Control System Effectiveness:
Measuring how well the system maintains optimal conditions and its response time.

Crop Yield and Efficiency:
Quantifying impact on crop yield and resource usage, comparing with traditional methods.

Energy Usage:
Calculating energy consumption and compare it with conventional methods.

Labor Reduction:
Analyzing the system's impact on labor requirements and costs.

Environmental Benefits:
Determining water and energy savings and potential emissions reductions.

Challenges:
Identifying limitations and factors affecting system performance.

Comparative Analysis:
Evaluating results against industry standards and similar projects.

Future Implications:
Discussing how findings may shape the future of greenhouse agriculture and suggest areas for improvement.

3.2 Design Method

Requirements Analysis: We had to buy the sensors and actuator first. Then for coding, an Arduino was required with a computer. These were the primary requirements.

System Architecture: The sensors(DHT22 , XULCMA , LDR) were connected to Arduino input pin with power supply. Then output pin went to the each actuators through relay.

Sensor Integration: DHT22 was for temperature and humidity sensing . And XULCMA was for soil moisture detecting as well. We have used 3 LDR for light intensity measuring.

Control Strategy:

- Was enable to read the temperature or not
- If it was below the range , nichrome coil should go on.
- When temp was increasing for the coil, DHT22 was sensing properly or not
- When temp was above the range, fan should be on.

Thus other parameters controlling were ensured as above.

User Interface:A 16*2 LCD display was bought. It shows the all parameter output.

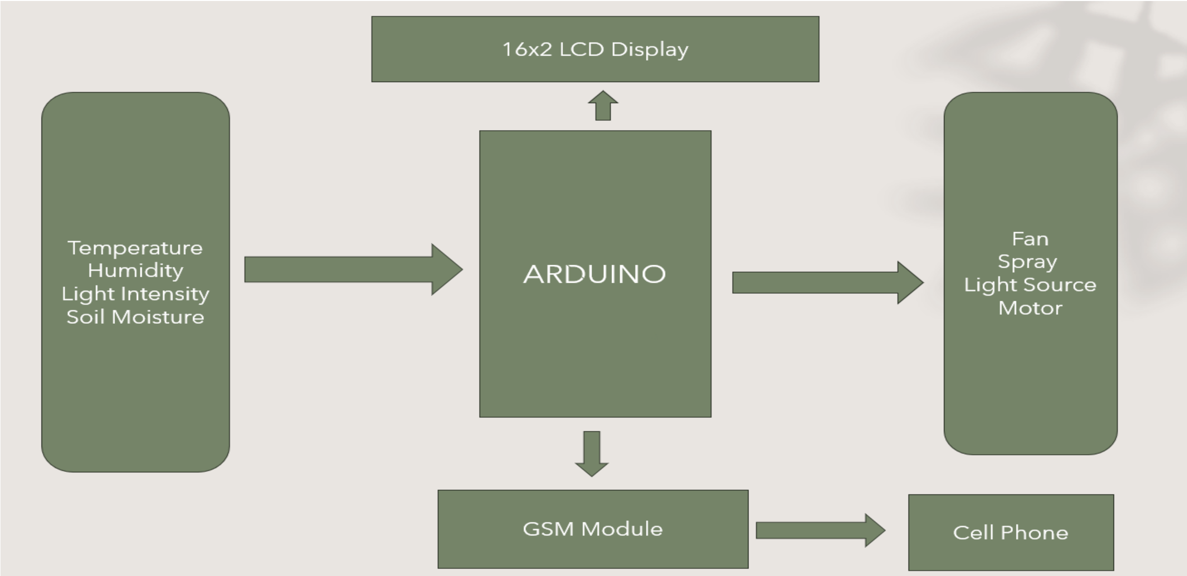
Hardware Setup: A glass-like plastic was designed for the greenhouse setup. Also there were two plants. Overall, the setup was well-furnished.

Software Development: All coding was done in Arduino IDE.

Performance Evaluation: We had to monitor system performance with proper surrounding condition.

Feedback and Iteration: For continuous improvement, we had to iterate same process many times with patience.

Block Diagram of our setup—



3.3Circuit Diagram

If appropriate for your project. Otherwise, remove this section

3.4Full Source Code of Firmware

Arduino Nano Code

```
#include <DHT.h>

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define LUX A0
#define DHTPIN 2
#define DHTTYPE DHT22
#define light 10
#define fan 8
#define humidifier 11
#define pump 5
#define soil 12
#define nichrome 7
```

```
//#define select_pin 3

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);

float temperature, humidity;
float volt, lux, value;
int
checkFan,checkLight,checkPump,checkSpray,checkNichrome;

void setup() {
  pinMode(soil, INPUT);
  pinMode(LUX,INPUT);
  //pinMode(select_pin,INPUT);
  pinMode(light, OUTPUT);
  pinMode(fan, OUTPUT);
  pinMode(humidifier, OUTPUT);
  pinMode(pump, OUTPUT);
  pinMode(nichrome, OUTPUT);

  Serial.begin(9600);
  //lcd.backlight();
  lcd.begin();
  lcd.setCursor(0,0);
  dht.begin();
}

void loop() {
  //sensing temperature


    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Auto Mode ON");
    delay(1000);
    temperature = dht.readTemperature();
    humidity = dht.readHumidity();
    if (temperature>20){
    digitalWrite(fan,HIGH);
    checkFan = 1;
    }
    else {
```

```
    digitalWrite(fan,LOW);
    checkFan = 0;
    }
    if (temperature<20){
    digitalWrite(nichrome,HIGH);
    checkNichrome = 1;
  }
  else {
    digitalWrite(nichrome,LOW);
    checkNichrome = 0;
  }
  if (humidity<80){
    digitalWrite(humidifier,HIGH);
    checkSpray = 1;
  }
  else {
    digitalWrite(humidifier,LOW);
    checkSpray = 0;
  }
  if(digitalRead(soil)==1){
    digitalWrite(pump,HIGH);
    checkPump = 1;
  }
  else{
    digitalWrite(pump,LOW);
    checkPump = 0;
  }
  value=analogRead(LUX);
  volt=(value/1023.0)*5;
  lux=((2500/volt)-500)/3.3;
  if(lux<10){
    digitalWrite(light,HIGH);
    checkLight = 1;
  }
  else{
    digitalWrite(light,LOW);
    checkLight = 0;
  }

  lcd.clear();
  lcd.setCursor(0,0);
  if (checkFan==1){
```

```
    lcd.print("Fan ON,");
  }
  else{
    lcd.print("Fan OFF,");
  }
  if(checkPump==1){
    lcd.print("Pump On");
  }
  else{
    lcd.print("Pump OFF");
  }

  lcd.setCursor(0,1);
  if(checkSpray==1){
    lcd.print("Spray ON");
  }
  else{
    lcd.print("Spray OFF");
  }
  delay(2500);

  lcd.clear();
  lcd.setCursor(0,0);
  if(checkLight==1){
    lcd.print("Light ON");
  }
  else{
    lcd.print("Light OFF");
  }

  lcd.setCursor(0,1);
  if(checkNichrome==1){
    lcd.print("Heater ON");
  }
  else{
    lcd.print("Heater OFF");
  }
  delay(2500);

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print("C, ");
```

```
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print("%, ");
  Serial.print("Light Intensity: ");
  Serial.print(lux);
  Serial.print("lx, ");
  if(checkPump==1){
    Serial.println("LOW Moisture ");
  }
  else{
    Serial.println("HIGH Moisture ");
  }


}
```

NodeMCU Code

```
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>



#define     BOTtoken     "6510119034:AAFd6fqo47fxM0a6QSB-
j5TZAbMxvUR8tyM"
#define CHAT_ID "1595737437"

//float temperature,humidity,lux,value,volt;

//long myChannelNumber = 2237985;
//const char myWriteAPIKey[] = "V3ZHHDPXGY1TG225";
X509List cert(TELEGRAM_CERTIFICATE_ROOT);
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  WiFi.begin("Mishti Kumra", "begun123");
```

```
  while(WiFi.status() != WL_CONNECTED)
  {
    delay(200);
    Serial.print("..");
  }
  Serial.println();
  Serial.println("NodeMCU is connected!");
  Serial.println(WiFi.localIP());
  //dht.begin();
  //ThingSpeak.begin(client);

  //pinMode(LUX,INPUT);
  configTime(0, 0, "pool.ntp.org");      // get UTC time
via NTP
  client.setTrustAnchors(&cert); // Add root certificate
for api.telegram.org
  bot.sendMessage(CHAT_ID, "System started", "");
  delay(1000);
}


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

  if(Serial.available()>0){
    //char data = Serial.read();
    //Serial.readBytesUntil('\n',buffer,100);
    //Serisubstral.println(buffer);
    String data = Serial.readStringUntil('\n');
    Serial.println(data);
    //int t = data.indexOf("Temperature");
    //int t_end = data.indexOf("C");
    //int h = data.indexOf("Humidity");
    //int h_end = data.indexOf("%");
    //int l = data.indexOf("Light");
    //int l_end = data.indexOf("lx");

    //if((t!=-1)&&(h!=-1)){
     //temperature = data.substring(t+12,t_end).toFloat();
     // humidity = data.substring(h+10,h_end).toFloat();
      //lux = data.substring(l+17,l_end).toFloat();
   // }
```

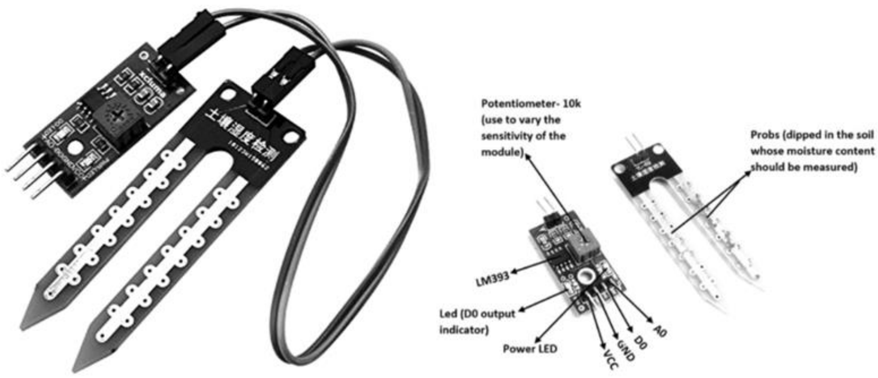
```
    bot.sendMessage(CHAT_ID, data , "");
  }
  //Serial.println("Temperature: " + (String) temperature
+ "C");
  //Serial.println("Humidity: " + (String) humidity + "%");
   //Serial.println("Light Intensity: " + (String)lux +
"lx");
  //ThingSpeak.writeField(myChannelNumber, 1, temperature,
myWriteAPIKey);
   //ThingSpeak.writeField(myChannelNumber, 2, humidity,
myWriteAPIKey);
     //ThingSpeak.writeField(myChannelNumber, 3, lux,
myWriteAPIKey);

  delay(5000);
}
```

3 Implementation

Description

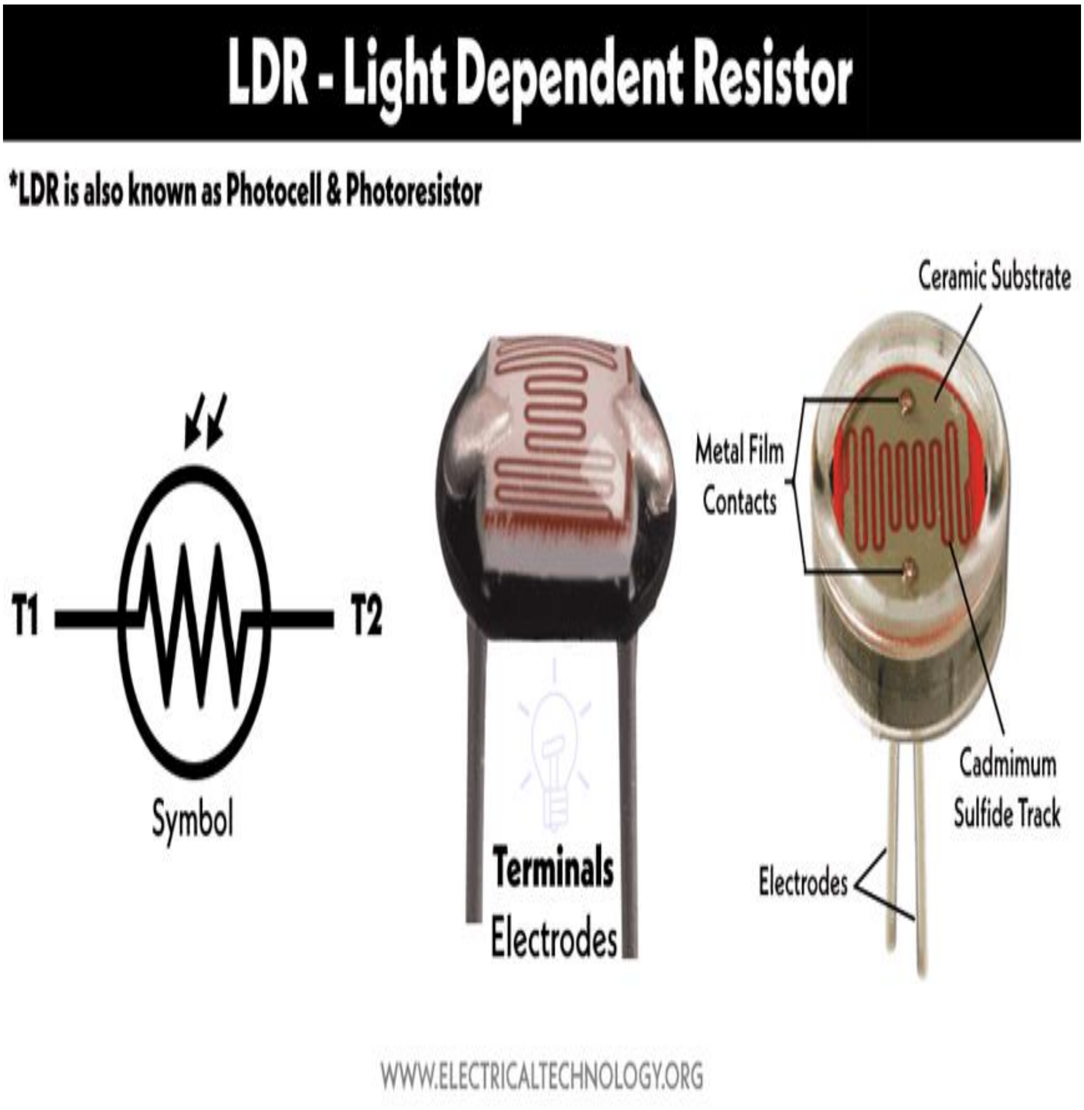
Soil Moisture Control:



The module determines the amount of soil moisture by measuring the resistor between two metallic probes that is inserted in the soil to be monitored.

A submersible DC pump is used to pump water in the soil is set up as the actuator. After measuring, if the moisture is found okay, the pump doesn’t turn on and if the soil moisture is found low, the pump flows water in the soil until a threshold moisture is reached.

Light Control:



- In our greenhouse prototype, a LDR (Light Dependent Resistor) is used to detect the light levels. Resistance of LDR decreases as light increases and vice versa. So. At high light, it shows low resistance with high current and in the dark at night, the resistance is high and little current flow.

- The LDR is connected to Arduino and the light (an LED) is put inside the prototype via a relay control mechanism.

Humidity Control:

DHT 22 is used to measure the humidity inside our greenhouse

We have used a water heater coil as humidifier. This water heater is put in water. The heater is operated by 220VAC. When the heater is on the water get boiled quickly and starts making vapor ,increasing the humidity.

Temperature Control:

Here we used a 9V DC fan for cooling the environment inside green-house . When the temperature is higher than the optimum value the fan with start and flow air to decrease the temperature

For sensing temperature, DHT22 is also used here. By flowing electricity through heating wire, heat is increased inside the prototype

3.5 Experiment and Data Collection

Experimental Setup:

A controlled greenhouse environment with defined sections for different crop types.

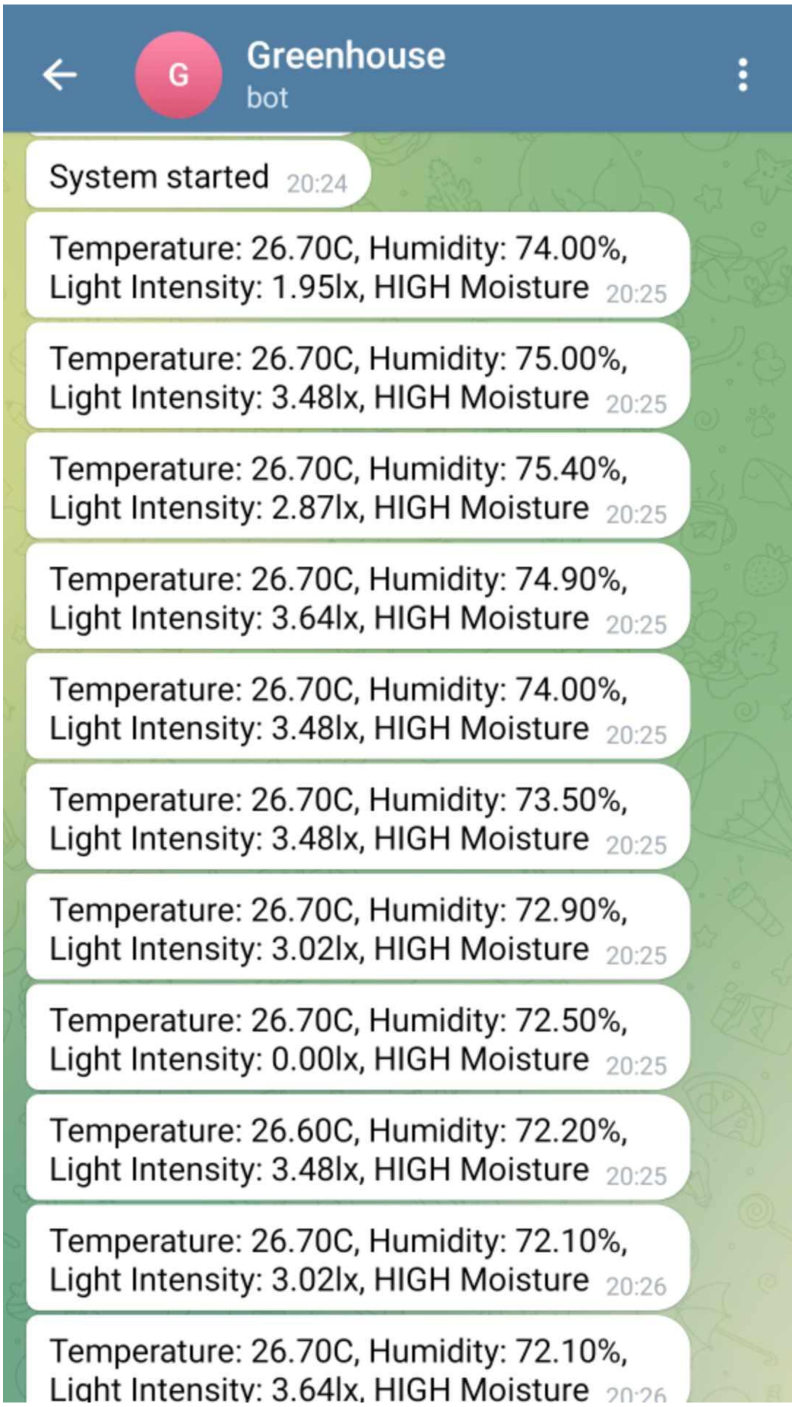
Arduino-based greenhouse controlling system equipped with temperature, humidity, soil moisture, and light intensity sensors.

Actuators including heating coils, fans, bulbs, and nichrome coils connected to the system via relays.

Data logging capabilities integrated into the Arduino code.

3.6 Data Analysis

Text sent to telegram app—



Here for first data, it is observed that temp is 26.70° C , so in this condition , fan will be on and heater will be off. As humidity is 60% above , so nichrome coil will also be producing heat. We further notice that light intensity is 1.95 lux which is lower than 10 lux giving in code. So light will be off.

3.7 Results

Overall result was satisfactory.

4 Design Analysis and Evaluation

4.1 Novelty

The novelty of this project is the system is fully automatic. Whenever any climatic factor is beyond its optimum level the system takes immediate action to keep that climatic factor under control.

This project confirms complete monitoring of the system. Any change in the climatic factor, information of any action always being sent to a mobile. So that information of the greenhouse can be obtained from another place.

Finally, our project is cost effective. This project shows a small size greenhouse control system at a comparatively low cost. Components used in this project are also of moderate price.

5.2 Design Considerations

5.2.1Considerations to public health and safety

This project is quite safe to public health and safety. The project is stable and easy to use so there is negligible chance for any accident to occur.

5.2.2 Considerations to environment

In our project we have addressed the issue of maintaining appropriate climatic factors for the survival of plants through an all-time running automated system. Plants are essential for the preservation of a balanced ecosystem. A healthy nature ensures the health of its people. So our project can ensure great service to the environment and therefore to public health and safety.

5.2.3Considerations to cultural and societal needs

Since our automated greenhouse system is relatively cheap, any person who has plans of making a small-scale garden can consider our proposed greenhouse model. So, our project has good social applications.

5.3 Limitations of Tools

Mostly the limitations we faced were due to hardware. But we faced some code related problems too. In the end, we were able to solve all the problems to make the project stable. We intend to use better sensors and closed environment for more precise control of our project.

5.4 Sustainability and Environmental Impact Evaluation

While working on our project, the only issues we faced were working with the temperature sensor and moisture sensor. We initially used DHT11 sensor for temperature and humidity measurement. However, the sensor either showed very inaccurate values and sometimes didn’t work at all. So, we replaced it with a DHT22 sensor which worked properly.

Regarding the moisture sensor, it worked fine most of the time. However, there were unfortunately a few cases where the moisture sensor failed to detect moisture.

5 Reflection on Individual and Team work

5.1 Individual Contribution of Each Member

ID: 1906175 (Nayem Hasan) and ID: 1906176 (Tanvir Rahman) worked with the hardware setup like the relay connections and battery connections with the individual devices and the overall setup.

ID: 1906179 (Samiul Hossain) and ID: 1906190 (Tousif Ansari) worked with Arduino and NodeMCU interfacing with the setup, mainly the software portion of the project.

5.2 Mode of Team Work

We have started our project work from mid-break. We had to sit for 7 times. So mainly our mode was offline based. We also faced some difficulties regarding the sensors and with coding. But later with frequent group sitting, we were able to overcome the issues.

6 Communication

6.1 User Manual

Our project is actually quite easy to use. It only requires a power source to work. All the necessary parameters are already coded but if someone wants to change the paraments then it is still very easy. A laptop can be used to change the values and the code is easy to understand which shouldn't be a problem for anyone.

7 Project Management and Cost Analysis

16x2 Serial LCD Module Display for Arduino Assembled Reference: RBD-1349	1	BDT 340	BDT 340
DHT11 Digital Relative Humidity and Temperature Sensor Module Reference: RBD-2094	1	BDT 99	BDT 99
LDR 5 mm Light Depending Resistor Reference: RBD-0995	3	BDT 6	BDT 18
ULN2003 Driver Reference: RBD-1279	1	BDT 18	BDT 18
BC547 NPN Transistor Reference: RBD-0758	2	BDT 4	BDT 8
Kiloohm (KΩ) 1/4w Resistors - Pack of 5 - Resistor Pack of 5 : 1 KΩ Reference: RBD-0245	2	BDT 5	BDT 10
Kiloohm (KΩ) 1/4w Resistors - Pack of 5 - Resistor Pack of 5 : 27 KΩ Reference: RBD-0245	1	BDT 5	BDT 5
50K Ohm Potentiometer Reference: RBD-1890	1	BDT 25	BDT 25
10K Ohm Potentiometer Reference: RBD-0996	1	BDT 17	BDT 17
Male to Male Jumper Wires 40 Pin 30cm Reference: RBD-2105	2	BDT 190	BDT 380
5v Relay module for Arduino Reference: RBD-2074	3	BDT 75	BDT 225

5 Watt DC 12V SMD LED Bulb Reference: RBD-1595	1	BDT 20	BDT 20
DC 5V Brushless CPU Cooling Fan with Screws for Raspberry Pi 4 Reference: RBD-1623	1	BDT 190	BDT 190
YL-69 Soil Hygrometer Humidity & Soil Moisture Detection Sensor Reference: RBD-0145	1	BDT 120	BDT 120
Push Button Switch (2PIN) Reference: RBD-0761	2	BDT 6	BDT 12
Piezoelectric Vibration Sensor Module with Analog & Digital Output Reference: RBD-2229	1	BDT 180	BDT 180
5v DC Mini Water Pump Micro Submersible Motor Pump 2.5-6V 120L/H Reference: RBD-0949	1	BDT 145	BDT 145
Breadboard Jumper Wire Set Reference: RBD-1153	1	BDT 220	BDT 220

Extra Cost

Arduino Nano	–	645
5 V DC Fan	–	50
Nichrome Coil	–	140
Wire	–	40
Relay(2 pcs)	–	150
Plant	–	100
Bulb	–	30
Socket	–	45
Battery (2 pcs)	–	180
Glue stick	–	30

Total cost is 3445 Tk

9 Future Work

So, our project has many ways for future expansion. We can implement better sensors for more precise control of the parameters. We can also simulate the project in a closed environment to produce accurate and expected results.

10 References

- https://www.engineersgarage.com/green-house-monitoring-using-arduino/
- https://srituhobby.com/iot-based-home-security-system-using-nodemcu-esp8266-module-and-telegram-app/#google_vignette