Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A REPORT ON

"Automated Greenhouse"

Code No: COMP 306

Submitted by

Aakriti Banjara (05)

Jayash Banjara (08)

Regal Adhikari (64)

Submitted to

Prof. Dr. Sudan Jha

Department of Computer Science and Engineering

Table of Contents

Project Title: Automated Greenhouse Control System

Brief Description

This project involves designing an automated greenhouse system to monitor and control environmental conditions such as temperature, humidity, light intensity, and soil moisture. The system employs various sensors and an Arduino microcontroller to automate the control of fans, heaters, and a water pump, ensuring optimal conditions for plant growth.

Component Breakdown

1. **Microcontroller**
    - Arduino UNO
2. **Sensors**
    - Temperature Sensor: LM35
    - Light Intensity Sensor: LDR (Light Dependent Resistor)
    - Soil Moisture Sensor: Custom resistive sensor
    - Humidity Sensor: Analog humidity sensor
3. **Actuators**
    - Relay Modules: 5V relay modules (RL1, RL2, RL3)
    - Fans: 220V fans controlled via relay
    - Heater: 220V heater controlled via relay
    - Water Pump: 12V water pump controlled via relay
4. **Display**
    - LCD Display: 16x2 LCD (LM044L)

Simulation Details

Approach: The project was simulated using Proteus software. The Arduino UNO microcontroller was programmed to read data from the sensors and control the actuators accordingly. The simulation setup includes connecting each sensor to the appropriate analog input pins on the Arduino and configuring the LCD to display real-time sensor data and system status.

Proteus Schematic

Connections

- Temperature Sensor (LM35): Connected to analog pin A1.
- Light Intensity Sensor (LDR): Connected to analog pin A0 through a voltage divider circuit.
- Soil Moisture Sensor: Connected to analog pin A2.
- Humidity Sensor: Connected to analog pin A3.
- LCD Display: Connected to digital pins 7-12 for data lines and control pins.

Relay Control

- Fan: Controlled by relay RL1, activated by digital pin D2.
- Heater: Controlled by relay RL2, activated by digital pin D3.
- Water Pump: Controlled by relay RL3, activated by digital pin D4.

Functionality Testing

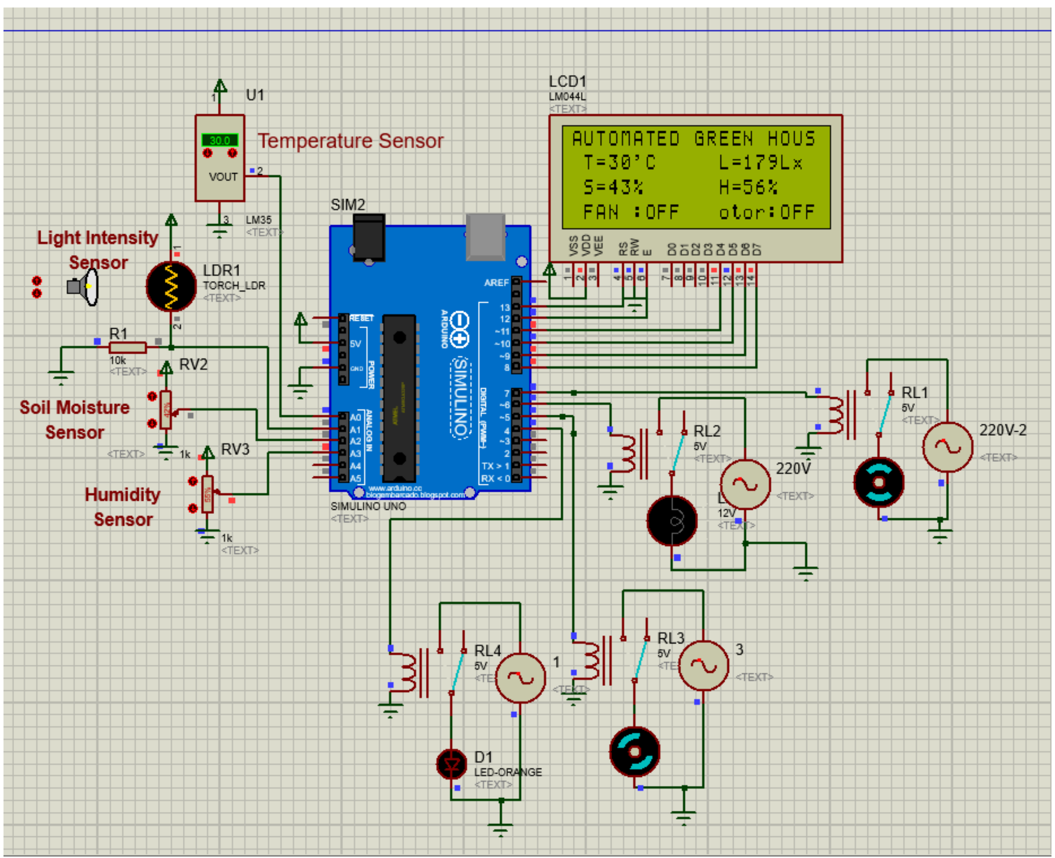
Scenarios Tested

1. **Temperature Control**
   - Scenario: If the temperature exceeds 30°C, the fan is activated. If the temperature drops below 18°C, the heater is activated.
   - Test Data: Adjusted the LM35 sensor output to simulate temperatures above and below the thresholds.
2. **Humidity Control**
   - Scenario: If humidity drops below 40%, the water pump is activated.
   - Test Data: Adjusted the humidity sensor output to simulate different humidity levels.

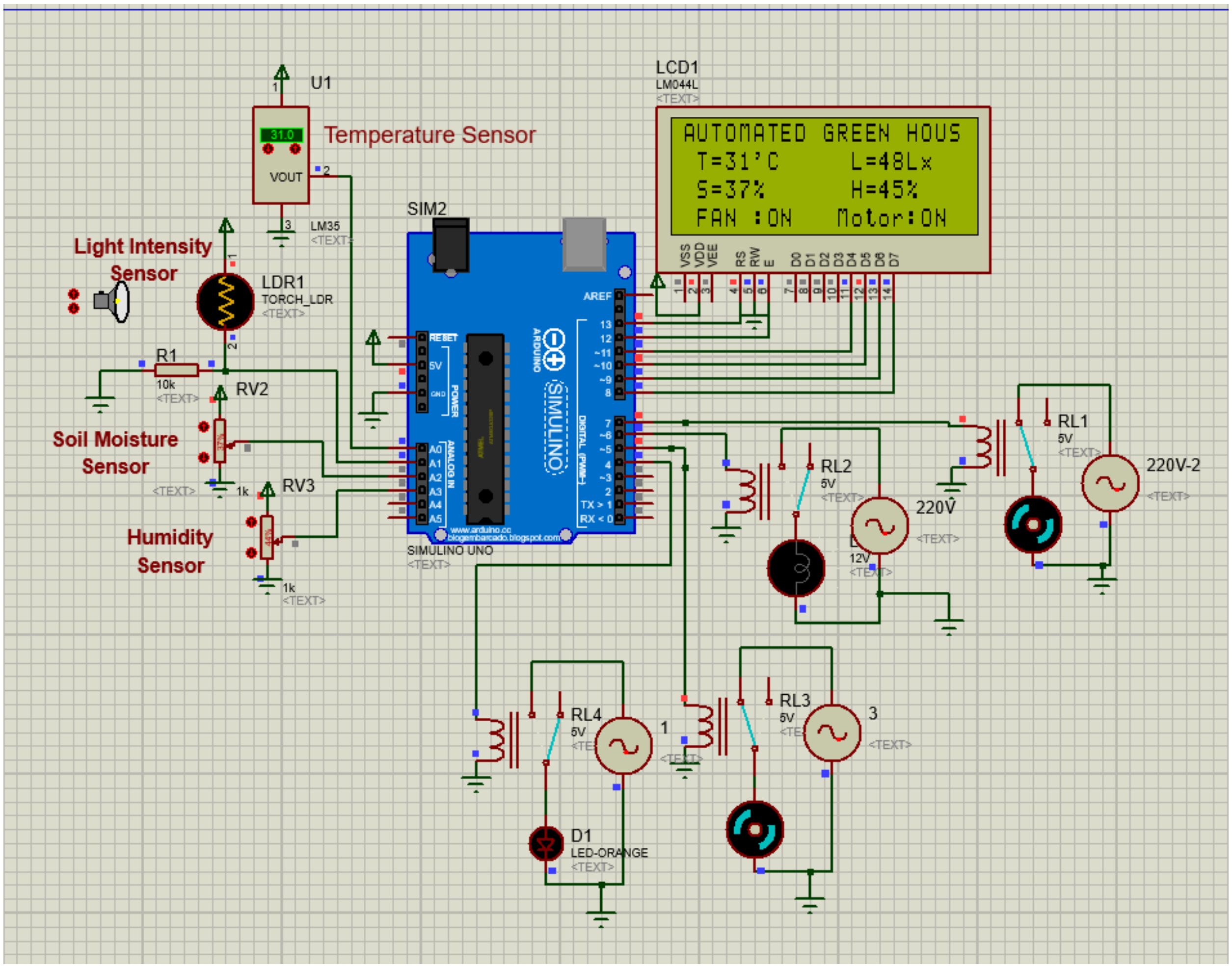
3. **Light Intensity Control**
    - Scenario: If light intensity is below 200 lux, an artificial light source is turned on.
    - Test Data: Adjusted the LDR output to simulate varying light conditions.
4. **Soil Moisture Control**
    - Scenario: If soil moisture is below a certain threshold, the water pump is activated.
    - Test Data: Adjusted the soil moisture sensor output to simulate dry soil conditions.

Screenshots

- Initial Conditions: Display shows normal temperature, humidity, light intensity, and soil moisture levels with all actuators off.



- High Temperature: Display indicates temperature above 30°C, and the fan is turned on.
- Low Humidity: Display shows humidity below 40%, and the water pump is activated.



Results and Analysis

The simulation successfully demonstrated the system's ability to monitor environmental conditions and control the actuators based on sensor inputs. Key observations include:

- The fan and heater were correctly activated in response to temperature changes.
- The water pump responded appropriately to low humidity and dry soil conditions.
- The light control mechanism was functional, turning on artificial light when necessary.

Insights

- Accurate calibration of sensors is crucial for reliable system performance.
- Relay response time and power considerations must be factored into the physical implementation.

Code

```
// include the library code:
#include <LiquidCrystal.h>

//LiquidCrystal lll(rs, en, d4, d5, d6, d7);
  LiquidCrystal lll(13, 12, 11, 10,  9,  8);

const int P_Pin = 6;
const int Q_Pin = 7;
const int R_Pin = 5;
const int S_Pin = 4;
void setup()
{
  lll.begin(20, 4); // set up the lll's number of columns and rows:
  lll.setCursor(0,0); // set the cursor position:
  // lll.print("AUTOMATED GREEN HOUSE");
  pinMode(P_Pin,OUTPUT);
  pinMode(Q_Pin,OUTPUT);
  pinMode(R_Pin,OUTPUT);
  pinMode(S_Pin,OUTPUT);
}

void loop()
{
  //Temperature Sensing
  int S1=analogRead(A0);  // Read Temperature
  int Temp=(S1*500)/1023; // Storing value in Degree Celsius
  lll.setCursor(0,0);
  lll.print(" T=");
  lll.print(Temp);
  lll.print("'C     ");

  //Light Intensity
  int S2=analogRead(A1);  // Read Light Intensity
  int LI=S2/1.9;
  lll.setCursor(11,0);
  lll.print(" L=");
  lll.print(LI);
```

```
lll.print("Lx        ");

//Soil Moisture
int S3=analogRead(A2);  // Read Soil Moisture
int SM=S3/10;
lll.setCursor(0,1);
lll.print(" S=");
lll.print(SM);
lll.print("%      ");

//Air Humidity
int S4=analogRead(A3);  // Read Air Humidity
int H=S4/10;
lll.setCursor(11,1);
lll.print(" H=");
lll.print(H);
lll.print("%   ");

if(LI<30)
{
  digitalWrite(P_Pin,HIGH);
  lll.setCursor(0,2);
  lll.print(" Light:ON   ");
}
else
{
  digitalWrite(P_Pin,LOW);
  lll.setCursor(0,2);
  lll.print(" Light:OFF   ");
}

if(SM<40)
{
  digitalWrite(Q_Pin,HIGH);
  lll.setCursor(10,2);
  lll.print(" Motor:ON    ");
}
else
{
```

```
    digitalWrite(Q_Pin,LOW);
    lll.setCursor(10,2);
    lll.print(" Motor:OFF    ");
  }
  if(Temp>30)
  {
    digitalWrite(R_Pin,HIGH);
    lll.setCursor(0,3);
    lll.print(" FAN :ON   ");
  }
  else
  {
    digitalWrite(R_Pin,LOW);
    lll.setCursor(0,3);
    lll.print(" FAN :OFF   ");
  }
  if(Temp<20)
  {
    digitalWrite(S_Pin,HIGH);
    lll.setCursor(11,3);
    lll.print(" HEATER :ON   ");
  }
  else
  {
    digitalWrite(S_Pin,LOW);
    lll.setCursor(11,3);
    lll.print(" HEATER :OFF   ");
  }
}
```

Conclusion

The automated greenhouse control system effectively maintains optimal growing conditions by utilizing sensors and actuators controlled by an Arduino microcontroller. The simulation validated the system's functionality, providing a strong foundation for developing a physical prototype.

Future Steps

- Develop a physical prototype to test the system in a real greenhouse environment.
- Integrate wireless connectivity for remote monitoring and control.
- Enhance the system with additional sensors for CO_2 levels and pH monitoring.

Contribution

- Project Lead and Circuit Design: Aakriti Banjara
- Programming and Simulation: Regal Adhikari
- Documentation and Report Preparation: Jayash Bhattarai

This project report outlines the implementation of an automated greenhouse control system, detailing the components used, simulation approach, functionality testing, and results obtained. The project demonstrates a successful proof of concept, with potential for further development and physical deployment.