Simulation of Arduino-Based Greenhouse Monitoring System Using TinkerCAD

Engr. Arvin C. Cabrera[1], Dr. Ma. Magdalena V. Gatdula[2], Engr. Michael Andre P. Guevarra[3], Engr. Christian Carr DG. Tac-an[4]

[1,4]Master of Science in Computer Engineering, Graduate School, Bulacan State University, Malolos City, 3000, Philippines

[2]Professor, Graduate School; University Registrar, Bulacan State University, Malolos City, 3000, Philippines

[3]Master of Engineering Program, Major in Computer Engineering, Graduate School, Bulacan State University, Malolos City, 3000, Philippines

DOI: https://dx.doi.org/10.47772/IJRISS.2025.91100367

Received: 26 November 2025; Accepted: 02 December 2025; Published: 11 December 2025

ABSTRACT

This study presents the Simulation of an Arduino-Based Greenhouse Monitoring System Using TinkerCAD, designed to demonstrate automated environmental monitoring and control within a greenhouse. The project aims to maintain optimal growing conditions by continuously measuring temperature, ambient lighting, and soil moisture through a temperature sensor, LDR, and soil moisture sensor. System responses are displayed via an LCD, while actuators—such as a fan, shade mechanism, and a light bulb representing a solenoid valve—operate based on threshold values.

The system architecture and block-based code were developed using TinkerCAD, incorporating timers and flag variables to mimic non-blocking execution despite platform limitations. The algorithm cycles through sensing, displaying, controlling, and reset phases, computing average sensor readings and activating actuators accordingly. Results confirmed correct detection and response across various environmental conditions, including temperature classification, light intensity interpretation, and soil moisture levels.

Overall, the simulation successfully achieved its objectives by demonstrating how automated greenhouse control can be implemented using Arduino components in a virtual environment. Future improvements may focus on enhanced synchronization of sensor data and more advanced control algorithms to better replicate real-world greenhouse automation.

Keywords: E-learning, Quiz Management, Web Application

INTRODUCTION

Crops such as leafy vegetables, root crops, and fruits are one of the major nutrient sources of humans making it a valuable produce of farms all over the Philippines. As an agricultural country, farmers need to devise solutions to cater for the needs of their crops. For example, in highland areas such as Benguet, Mountain Province, and Tagaytay, farmers needed a way to keep their crops warm enough to survive cold nights, especially during colder seasons. On the other hand, in lowland areas such as Central Luzon or Mindanao plains, farmers need to limit the exposure of crops to extreme heat to avoid withering and to minimize drying of soil. According to published research by Dolom et. Al (2023) [1], building greenhouse structures is one of the adaptation strategies to combat continuous changes in the climate in Benguet. To aid the farmers in maintaining the health of the crops, the developers developed an Arduino-Based Greenhouse Monitoring System which not only measures the temperature, ambient lighting, and soil moisture but also automatically regulates the greenhouse by turning on sprinklers and fans. By controlling the crops environment, we can make sure that the crops are in optimal condition thus increasing the chances of producing bountiful crops.

Problem Statement

- Due to the country's geophysical location, the Philippines is prone to severe weather conditions making it harder to grow crops.
- Greenhouses need constant monitoring especially for crops that are environmentally sensitive.
- Operating greenhouses includes manual labor which takes time especially if manpower is limited.

Project Objectives

The general problem of this research is to be able to simulate monitoring and to automate environment control of a greenhouse to maintain optimal conditions for the crops with adjustable thresholds to cater for the needs of various crops through TinkerCAD block-based programming.

Specifically, the team aims to execute the following:

- To measure the greenhouse temperature, ambient lighting, and soil moisture for monitoring and control using temperature sensors, light-dependent resistors (LDR), and soil moisture sensors.
- To notify the user of current temperature, ambient lighting, and soil moisture of the greenhouse.
- To optimize the greenhouse environment by turning on the fan or the sprinkler, and provide extra shade depending on the monitoring values with the use of servo motors and LEDs representations.
- To be able to control thresholds with the use of buttons and potentiometers.

RELATED WORKS

With the advancements and recent trends on Internet-of-Things, greenhouse systems that apply this kind of technology emerge and already shown promise, in the field of agricultural practices. Akpulonu et. al (2024) [2] proposed an IoT-based greenhouse system which measures different conditions, that typically determines crop cultivation efficiency. To check and to control the environment of the crop yield that they are studying with, they utilized sensors and actuators for different parameters like temperature, humidity, sodium, potassium, light, phosphorus, and soil moisture. In this way, their system can regulate these environmental parameters - making the crop cultivation more efficient. Furthermore, one feature that they also have is the real-time data recorder, to have live updates regarding the environment and reactively execute and perform regulation methods that they have designed. With their result showing a 20% crop yield increase, it shows how microcontroller-based systems can really impact the quality of life - for this is for plants.

Patil et. al (2024) [3] developed an IoT-based greenhouse monitoring and control system as well, wherein they mainly focused on measuring and controlling lighting, watering, and aeration factors. In their process, they send the data collected on ThingSpeak, a cloud-based server. Here the data is now displayed which they analyze and create further solutions to help in crop cultivation. With this study, the current researchers obtain one way of recording information received from Arduino Uno and the sensors integrated with it, and also how to still measure parameters accurately, even when regulators like motors, starts to spin or perform their programmed tasks.

Hoque et. al (2020) [4] proposed an automated greenhouse monitoring and controlling system, incorporating sensors for measuring temperature, humidity, light, and soil moisture. Furthermore, they integrated tools like Arduino Uno R3, GSM module, a solar power system, and IoT. Aside from these, they included the cost analysis with their developed prototype, to show how cost-effective and economical it is, especially with their target users - which are farmers or agricultural workers. In their results, they have stated that what they did is effective, with the effective functioning of their features like monitoring and controlling the light intensity, air humidity, inside temperature, and soil moisture level. Also, the integration of GSM module is properly implemented, as users can text a specific SMS message, and then the system will give what the user have asked for. With these, the researchers can have a guide on the process that they will perform on their methodology, and also on how can they discuss and explain the results, since the materials and the objectives of both researches are aligned.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

RESEARCH METHODOLOGY

To build the system, the researchers have identified and listed the components that will be used, as well as on how these parts will function as one – generating the design as shown on Figure 1.

Table 1 Components and Variables of the System

Component	Type	Function	Description
Arduino Uno	Controller	•Serves as the main controller of the system.	•Reads all sensor data (soil moisture, temperature, light). •Controls actuators such as the fan, sprinkler, LED, and relay. •Displays data and settings on the LCD.
Push Button	Input	•Used for user input to adjust system settings.	•SET button: cycles through adjustable settings. •UP button: increases selected value. •DOWN button: decreases selected value.
Voltage Multimeter	Input/ Output	•Ensures correct operation and safe voltage levels.	•Used during testing to measure voltage from sensors, solar cell, or power lines.
LCD 16x2	Output	•Shows system mode and configuration when adjusting settings.	•Displays real-time sensor readings (moisture, temperature, light).
250 kΩ Potentiometer	Input	• Used for user input to adjust a system setting.	•Used to change the threshold values of greenhouse.
220 Ω & 10kΩ Resistors	Input	•Regulates voltage flow.	•Current-limiting resistor for the red LED. •Prevents damage to the LED.
Soil Moisture Sensor	Input	•Measures soil water content.	•Provides an analog signal used to control the sprinkler motor.
Relay SPDT	Output	•Used to control higher voltage or current loads safely.	•Can be used to switch external devices such as pumps or lights.
DC Motor	Output	•Represents actuators in the system.	•Fan motor: turns ON when temperature is high. •Sprinkler motor: turns ON when soil moisture is low.
12 V, 1 A Solar Cell	Power	•Acts as both a power source and a light detector for the system.	•Demonstrates renewable energy integration. •Works with the photoresistor to simulate daytime detection.
H-bridge Motor Driver (L293D)	Output	•Interfaces the motors safely with Arduino outputs.	•Drives the two DC motors (fan and sprinkler). •Allows bidirectional control of each motor.
Temperature Sensor [TMP36]	Input	•Measures ambient temperature.	•Provides analog output proportional to temperature. •Used to trigger the fan when temperature exceeds a threshold.
Photoresistor	Input	•Detects ambient light levels to determine day or night.	•Works with a 10 kΩ resistor as a voltage divider. •Affects moisture and temperature behavior during simulation.
Red LED	Output	•Serves as an indicator for daytime.	•Turns ON when light intensity (from the photoresistor) exceeds a threshold.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 1 Architecture Diagram of the System

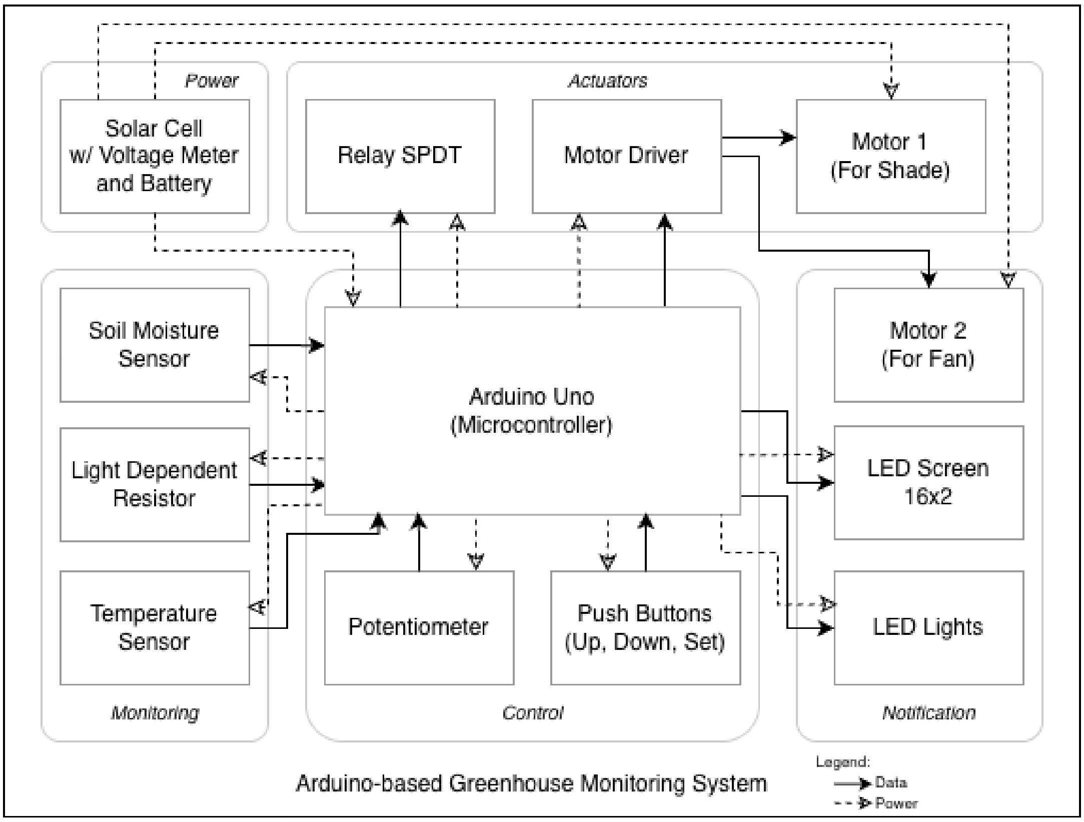
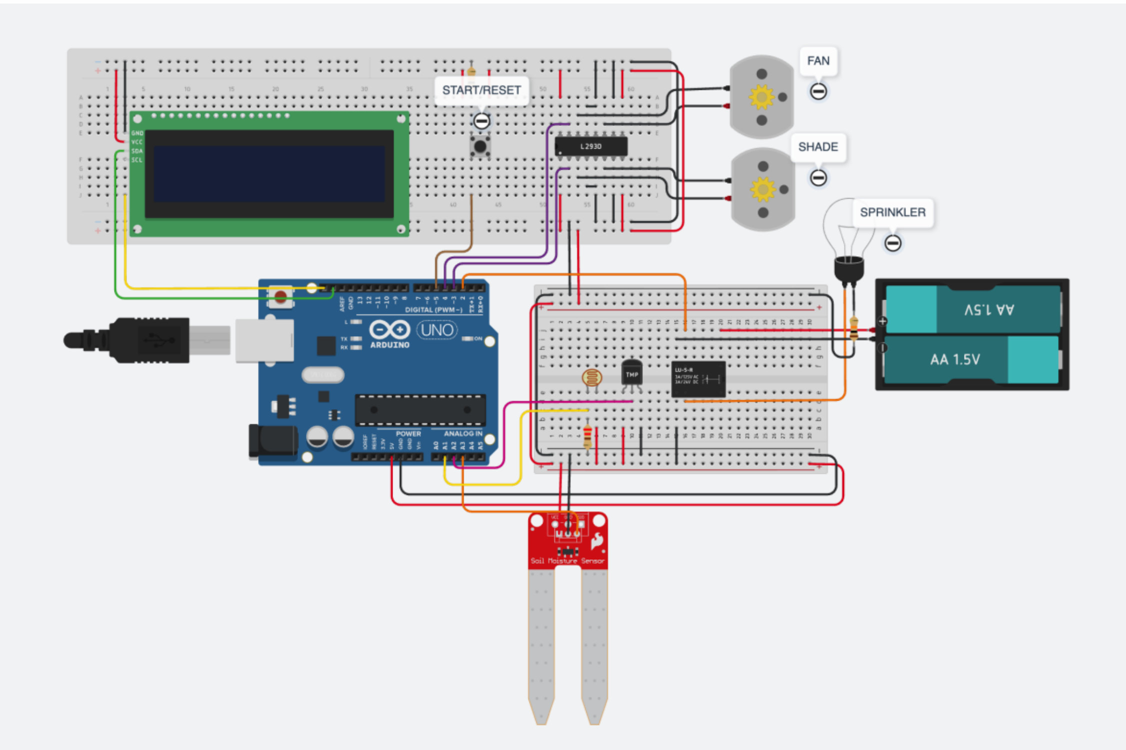


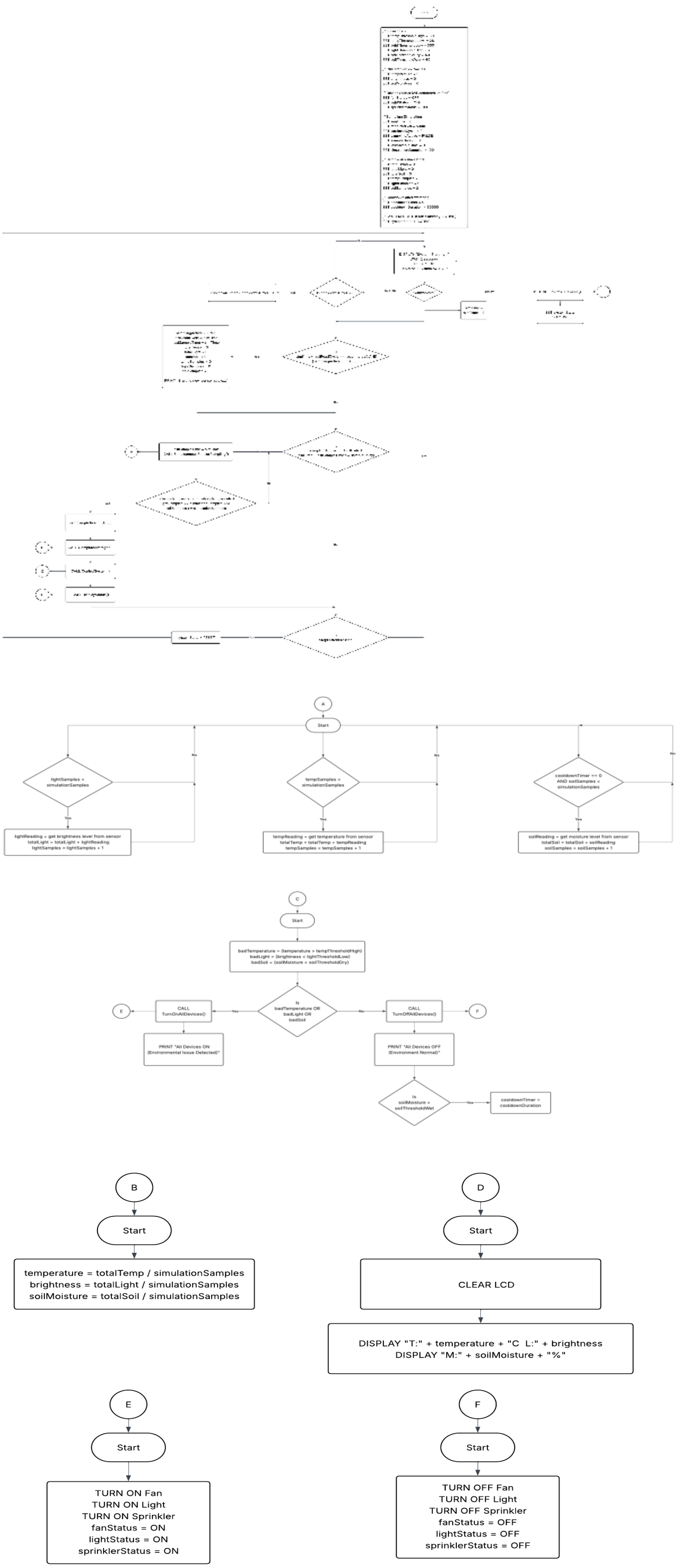
Figure 2 TinkerCAD Design of the System



As shown in Figure 2, the system features an LCD screen to display real-time status and sensor readings. It incorporates motors to control a fan and a shade mechanism, allowing dynamic adjustment of airflow and sunlight exposure. The system is equipped with sensors, including a temperature sensor, moisture sensor, and photoresistor, to continuously monitor environmental conditions. Additionally, a light bulb is utilized as a solenoid valve to regulate fluid flow when necessary. The entire setup is powered by two 1.5V AA batteries specifically to operate the light bulb, while the Arduino and other components are powered separately.

INTERNATIONAL JOURNAL OF RESEARCH AND INNOVATION IN SOCIAL SCIENCE (IJRISS)
ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 3 Flowchart of the System



ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

As shown on Figure 3, the process of the logic or programming side of the system is illustrated. It is composed of three images – to represent the functions in flowchart way. The first image is the main loop, where all the functionalities are summarized. On the second and third images are the functions that features on the first image utilizes. To elaborate, the flowchart shown on the last two charts include functions for simultaneous sensor sampling, for computing averages, for control system, for displaying the status, and for turning on and off of all devices, as configured.

Figure 4.1 SMART Greenhouse Monitoring System TinkerCAD Code Block, part 1

onStart()

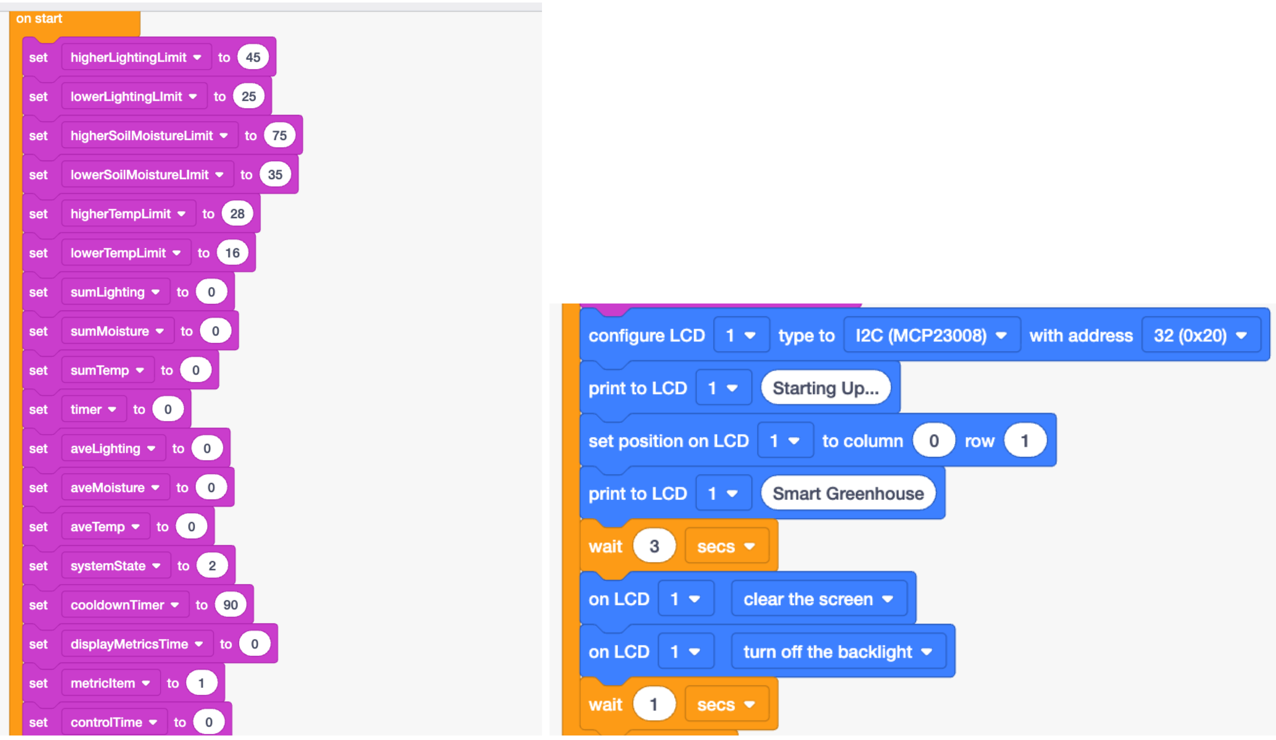


Figure 4.2 SMART Greenhouse Monitoring System TinkerCAD Code Block, part 2

Forever()

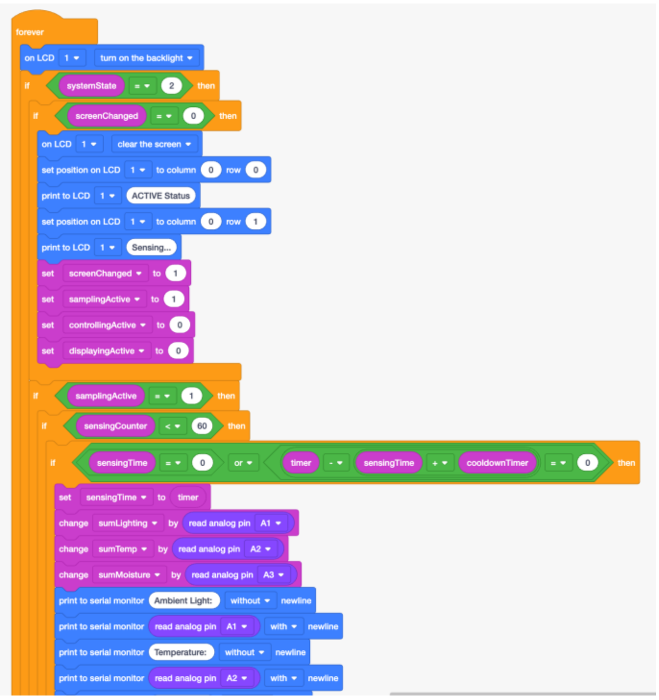


Figure 4.3 SMART Greenhouse Monitoring System TinkerCAD Code Block, part 3

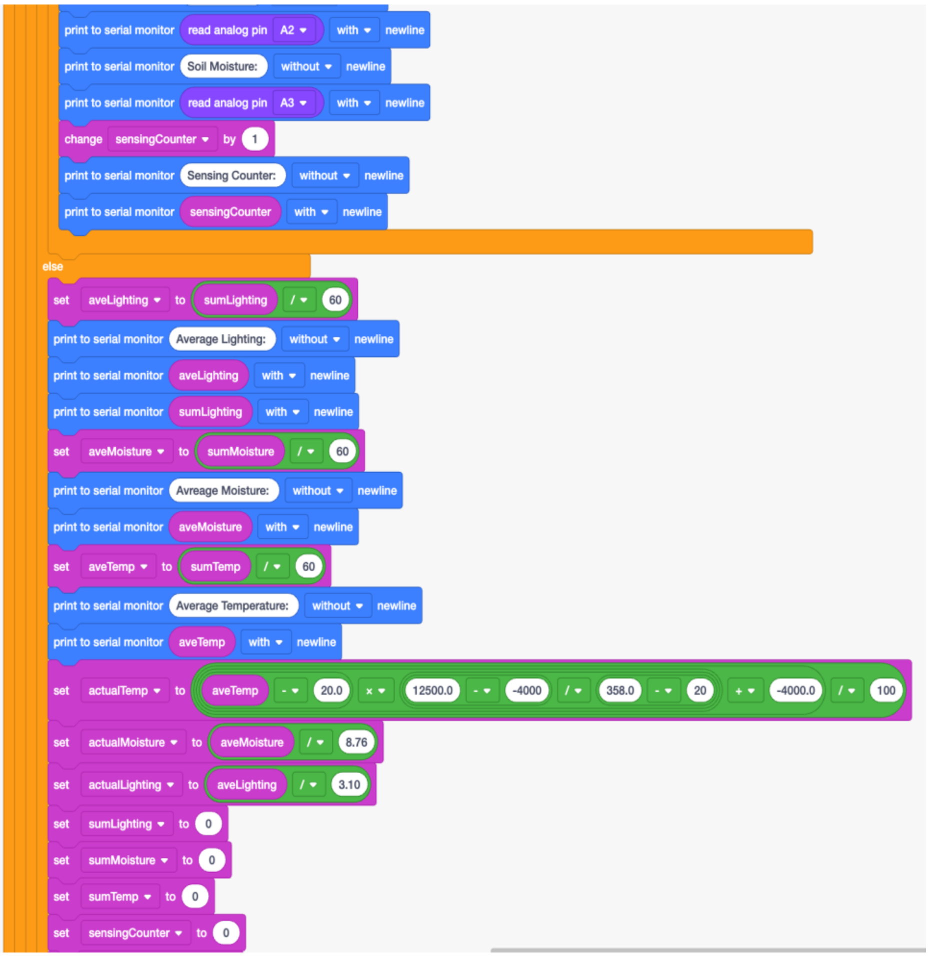
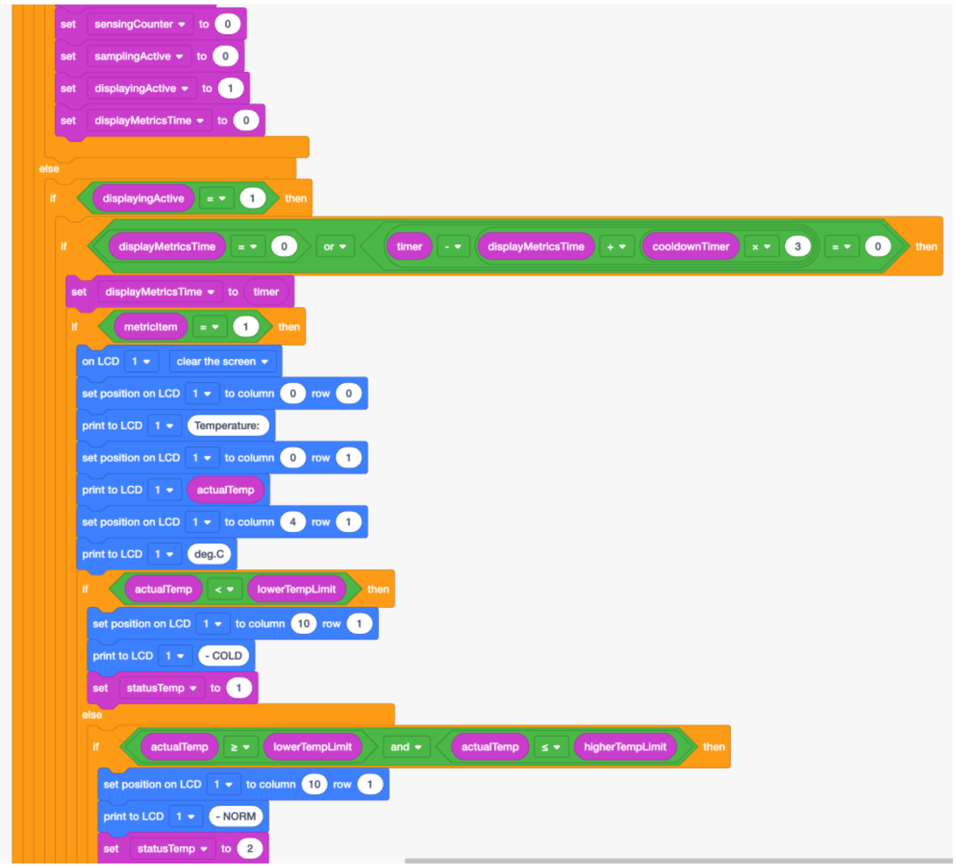


Figure 4.4 SMART Greenhouse Monitoring System TinkerCAD Code Block, part 4



ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 4.5 SMART Greenhouse Monitoring System TinkerCAD Code Block, part 5

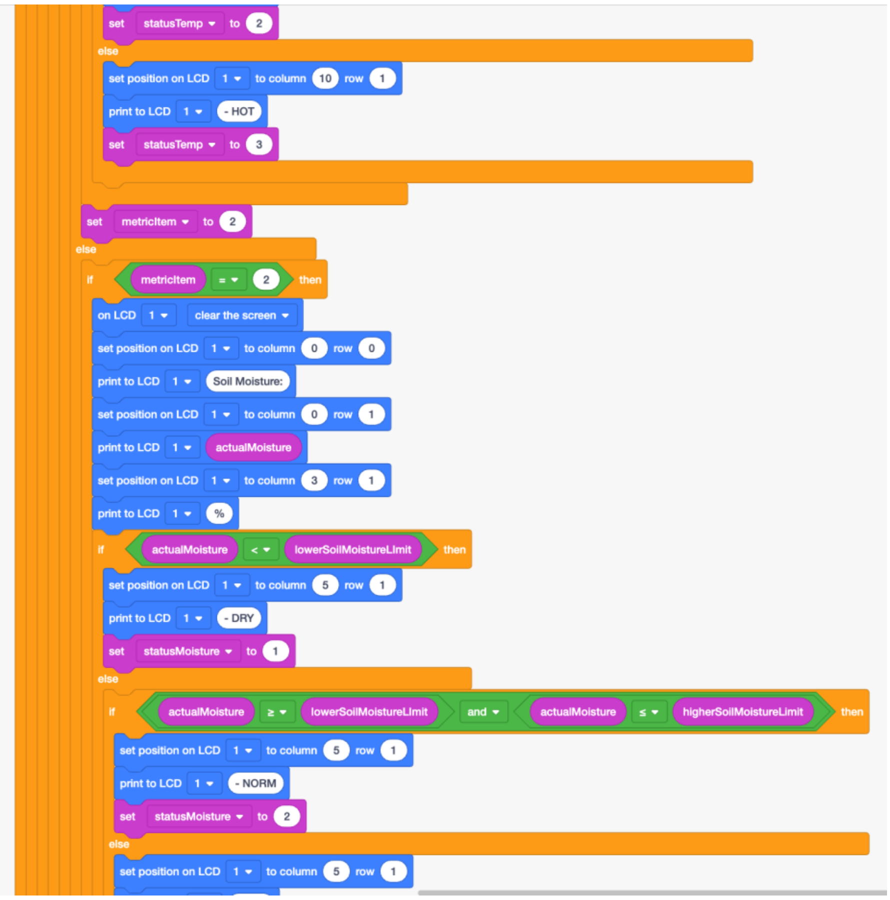
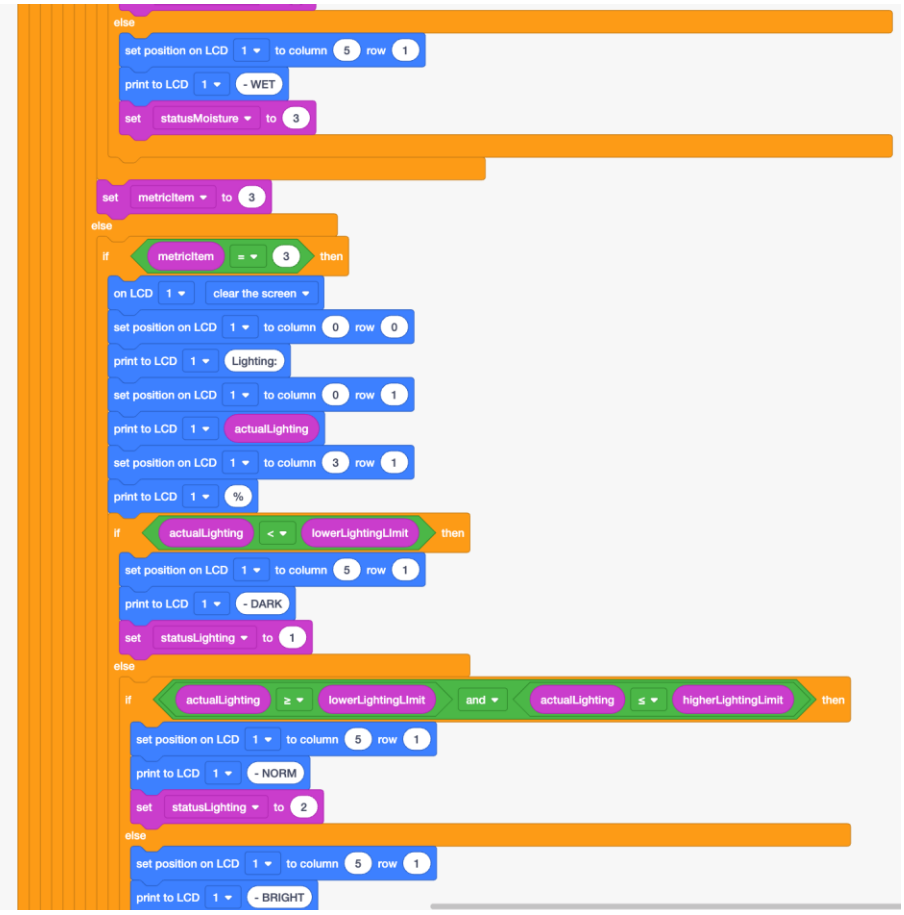


Figure 4.6 SMART Greenhouse Monitoring System TinkerCAD Code Block, part 6



INTERNATIONAL JOURNAL OF RESEARCH AND INNOVATION IN SOCIAL SCIENCE (IJRISS)
ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 4.7 SMART Greenhouse Monitoring System TinkerCAD Code Block, part 7

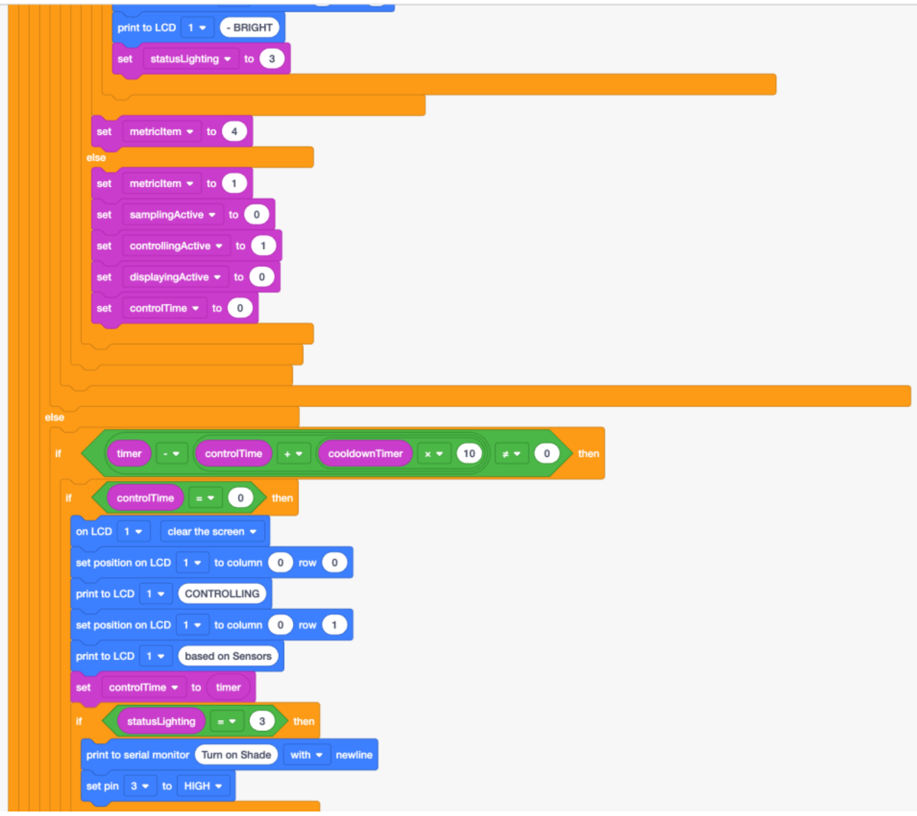
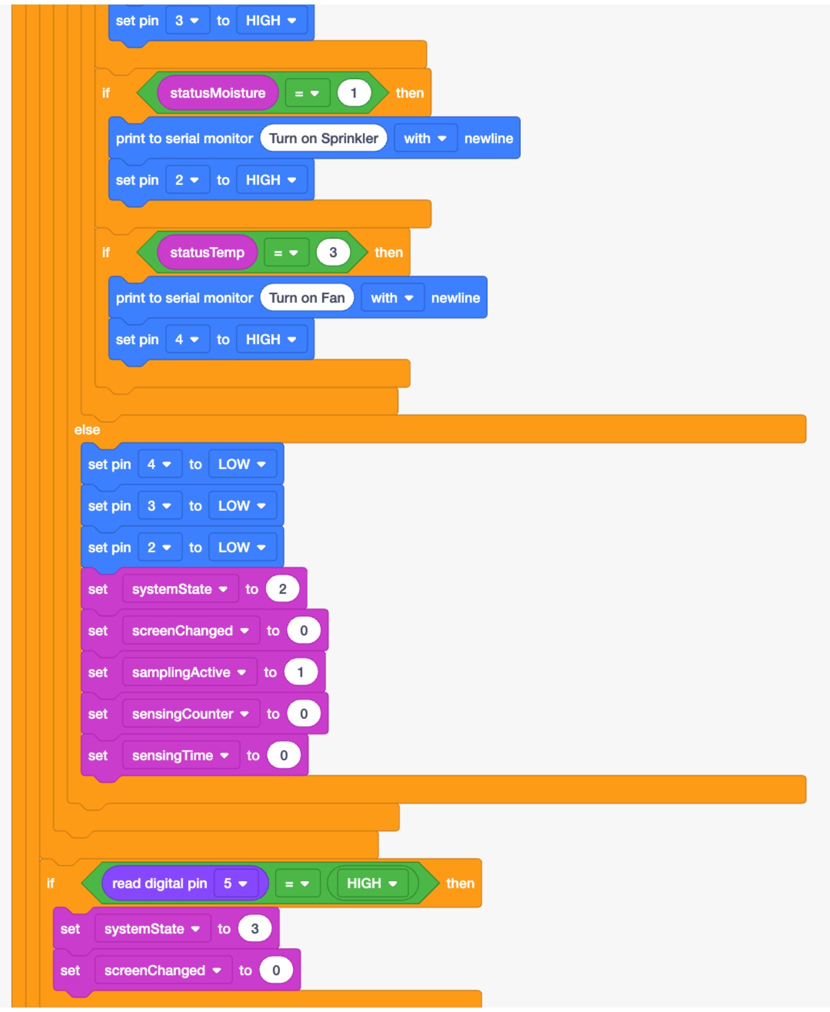
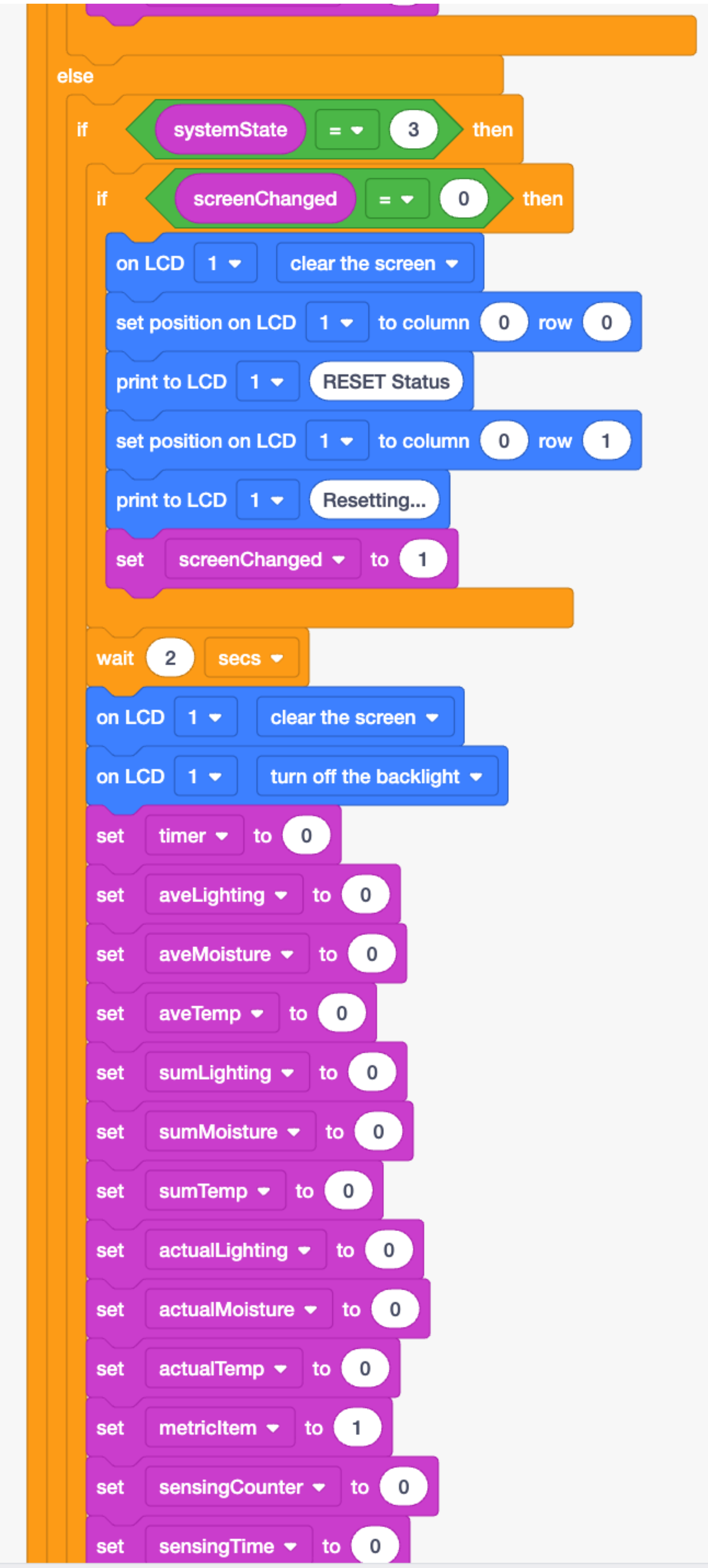


Figure 4.8 SMART Greenhouse Monitoring System TinkerCAD Code Block, part 8



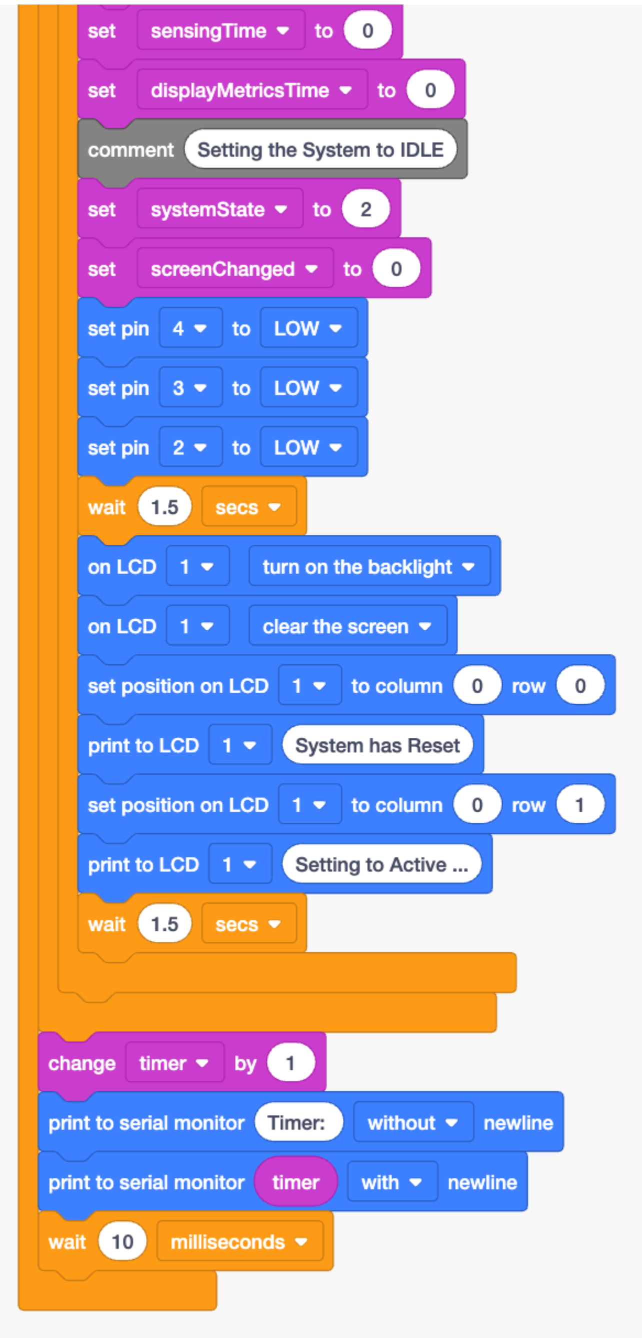
ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 4.9 SMART Greenhouse Monitoring System TinkerCAD Code Block, part 9



ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 4.10 SMART Greenhouse Monitoring System TinkerCAD Code Block, part 10



The Smart Greenhouse Monitoring System collects analog values from the Light-Dependent Resistor, Temperature Sensor, and Soil Moisture Sensor. After collecting enough data, the microcontroller analyzes it and controls the motor for fan, motor for shade, and the relay for sprinkler, accordingly. To do this, the developers thought of a way that the system can run in a non-blocking sequence, meaning all the process doesn't have to wait for the other process to be finished. However, TinkerCAD does not support asynchronous wait. That is why we used timers and flaggers to determine where the execution should be. As seen in the code blocks, the wait-block was only used when the system resets or starts while there is none during the sensing and controlling. The algorithm has 4 phases – System Start, Active (Sensing), Displaying, Controlling, and System Reset. After the System Start phase, it will transition to Active (Sensing) where it gathers data every second 60 times (equivalent to 1 minute). The microcontroller then computes the average measurement and assigns necessary flaggers depending on the set thresholds. Afterwards, it will go to the Displaying phase where the LCD displays all the metrics and their equivalent interpretation for low, normal, and high values. It will proceed to the Controlling phase where the actuators move depending on the flags set during the Active (Sensing) phase, which will last for 10 seconds. Automatically, the system will loop back to Active (Sensing) phase and will repeat the whole process unless the reset button is pressed entering the System Reset phase. In this phase, all

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

variables are set to their default value and will return to Active (Sensing) phase starting fresh. Anytime during the iterating processes, the reset button can be pressed which can be used in emergency use cases.

The algorithm utilizes variables for different purposes. For example, the cooldownTimer - as seen in Figure 1, is set to 90 which counts how many loops were done by the forever loop. By the timer variables (i.e. sensingTime, displayMetricsTime) reached the value equal to cooldownTimer, it has passed roughly 1 second in real time. If the cooldownTimer is paired with a multiplier (i.e. cooldownTimer x 10) as seen in Figure 7, we can now expand the time accordingly. The timer variable changes its value by 1 every iteration in the forever loop. It is also the basis of the value set in cooldownTimer where 90 iterations are roughly equivalent to 1 second. Flaggers (i.e. screenChanged, statusLighting, statusTemp) aid the system to know the next instructions to execute. As seen in Figure 9, the screenChanged flagger is used to keep the LCD display steady. When its value is 1, the LCD will not be commanded to display again, preventing unwanted flickering.

TinkerCAD Code Block feature is limited to fundamental blocks only. That’s why some of the features, such as storing data in an array or displaying floating point values, are not possible. The developers have just thought of a work-around to work in a similar way. Moreover, the measurements displayed and used in the computations are only estimations and cannot be interpreted as actual value.

Table 2 Table of Variables

Variable / Component	Type (Input / Output)	Parameter Measured / Controlled	Condition or Range	System Response / Action
LDR (Light-Dependent Resistor)	Input	Light Intensity	0% – 24% = Low	Display “Dark”
			25% - 45% = Normal	Display “Normal”
			46% - 100% = High	Display “Bright” Open motor (shade) for 10 seconds
Soil Moisture Sensor	Input	Soil Moisture	0% - 34% = Low	Display “Dry” Open relay (sprinkler) for 10 seconds
			35% - 75% = Normal	Display “Normal”
			76% - 100% = High	Display “Wet”
Temperature Sensor	Input	Temperature – Celsius	15 °C and below = Low	Display “Cold”
			16 °C - 28 °C = Normal	Display “Normal”
			29 °C and up = High	Display “Hot” Open motor (fan) for 10 seconds
Button	Input	Manual Override	If pressed	Stop controlling, sensing, and displaying. Reset flaggers and variables to default values.
Motor Driver	Output	Motor Activity	If average light intensity is High	Motor for shade rotates
			If average temperature High	Motor for fan rotates
Relay	Output	Light Bulb activity (in replace of sprinkler)	If average soil moisture is high	Light Bulb turns on (in replace of Sprinkler)
LCD Screen	Output	Text Display	Logic-controlled	Displays status, system mode, and metric values
Arduino	Controller	Input / Output Processing	Logic-controlled	Execution is based on loops and if-else conditions.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Table 3 Test Cases

Test Num	Input Condition	Expected Output
1	Temp1 = 12 Temp2 = 10 Temp3 = 14 Temp4 = 13 Temp5 = 11 Temp6 = 10 Temp7 = 12 Temp8 = 14 Temp9 = 15 Temp10 = 17	The LCD displays "Temperature 12 deg. C – COLD"
2	Temp1 = 25 Temp2 = 28 Temp3 = 23 Temp4 = 18 Temp5 = 20 Temp6 = 25 Temp7 = 26 Temp8 = 25 Temp9 = 24 Temp10 = 18	The LCD displays "Temperature 20 deg. C – NORMAL"
3	Temp1 = 45 Temp2 = 50 Temp3 = 80 Temp4 = 85 Temp5 = 75 Temp6 = 125 Temp7 = 124 Temp8 = 125 Temp9 = 120 Temp10 = 120	LCD displays "Temperature 94 deg. C – HOT" Motor for Fan turns on for 10 seconds
4	Lighting1 = 7 Lighting2 = 12 Lighting3 = 3 Lighting4 = 9 Lighting5 = 0 Lighting6 = 14 Lighting7 = 6 Lighting8 = 10 Lighting9 = 4 Lighting10 = 13	The LCD displays "Lighting 7% - DARK"
5	Lighting1 = 33 Lighting2 = 40 Lighting3 = 27 Lighting4 = 36 Lighting5 = 42 Lighting6 = 25 Lighting7 = 38 Lighting8 = 31 Lighting9 = 44 Lighting10 = 29	The LCD displays "Lighting 34% - NORMAL"

6	Lighting1 = 73 Lighting2 = 91 Lighting3 = 58 Lighting4 = 84 Lighting5 = 46 Lighting6 = 100 Lighting7 = 67 Lighting8 = 79 Lighting9 = 52 Lighting10 = 95	The LCD displays “Lighting 74% - BRIGHT” Motor for Shade turns on for 10 seconds
7	SoilMoisture1 = 12 SoilMoisture2 = 27 SoilMoisture3 = 5 SoilMoisture4 = 30 SoilMoisture5 = 18 SoilMoisture6 = 9 SoilMoisture7 = 22 SoilMoisture8 = 16 SoilMoisture9 = 33 SoilMoisture10 = 11	The LCD dislplays “Soil Moisture 18% - DRY” Relay for light bulb (a.k.a. sprinkler) turn on
8	SoilMoisture1 = 63 SoilMoisture2 = 47 SoilMoisture3 = 70 SoilMoisture4 = 55 SoilMoisture5 = 41 SoilMoisture6 = 68 SoilMoisture7 = 60 SoilMoisture8 = 39 SoilMoisture9 = 75 SoilMoisture10 = 52	“The LCD displays Soil Moisture 57% - NORMAL”
9	SoilMoisture1 = 92 SoilMoisture2 = 81 SoilMoisture3 = 99 SoilMoisture4 = 84 SoilMoisture5 = 77 SoilMoisture6 = 95 SoilMoisture7 = 88 SoilMoisture8 = 79 SoilMoisture9 = 100 SoilMoisture10 = 93	“The LCD displays Soil Moisture 88% - NORMAL”
10	Reset Button is pressed	System resets The LCD Displays “System Reset” The system returns to Active status

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

RESULTS AND DISCUSSIONS

Table 4 Test Cases Results

Test Num	Input Condition	Expected Output	Observed Output	Remarks
1	Temp1 = 12 Temp2 = 10 Temp3 = 14 Temp4 = 13 Temp5 = 11 Temp6 = 10 Temp7 = 12 Temp8 = 14 Temp9 = 15 Temp10 = 17	The LCD displays “Temperature 12 deg. C – COLD"	The LCD displayed “Temperature 12 deg. C – COLD"	When the sensor reads ≤17°C, it displays “Temperature: 12°C – COLD,” showing correct temperature detection, and the Fan remains off
2	Temp1 = 25 Temp2 = 28 Temp3 = 23 Temp4 = 18 Temp5 = 20 Temp6 = 25 Temp7 = 26 Temp8 = 25 Temp9 = 24 Temp10 = 18	The LCD displays “Temperature 20 deg. C – NORMAL"	The LCD displayed “Temperature 28 deg. C - NORMAL”	When the sensor reads between 18°C and 28°C, it displays “Temperature – NORMAL,” indicating correct temperature detection, and the Fan remains off
3	Temp1 = 45 Temp2 = 50 Temp3 = 80 Temp4 = 85 Temp5 = 75 Temp6 = 125 Temp7 = 124 Temp8 = 125 Temp9 = 120 Temp10 = 120	LCD displays “Temperature 94 deg. C – HOT" Motor for Fan turns on for 10 seconds	The LCD displayed “Temperature 48 deg. C – HOT"	When the sensor reads 29°C or higher, it displays “Temperature – HOT,” confirming accurate temperature detection, and the Fan turns on.
4	Lighting1 = 7 Lighting2 = 12 Lighting3 = 3 Lighting4 = 9 Lighting5 = 0 Lighting6 = 14 Lighting7 = 6 Lighting8 = 10 Lighting9 = 4 Lighting10 = 13	The LCD displays “Lighting 7% - DARK"	The LCD displayed “Lighting: 12% – DARK"	When the light level is below 25%, the LCD displays “Lighting – DARK,” confirming accurate light detection,and the Shade remains off
5	Lighting1 = 33 Lighting2 = 40 Lighting3 = 27 Lighting4 = 36	The LCD displays “Lighting 34% - NORMAL”	The LCD displayed “Lighting:	When the light level is between 25% and 45%, the LCD displays “Lighting – NORMAL,” indicating

	Lighting5 = 42 Lighting6 = 25 Lighting7 = 38 Lighting8 = 31 Lighting9 = 44 Lighting10 = 29		27% – NORMAL"	accurate light detection, and the shade remains off
6	Lighting1 = 73 Lighting2 = 91 Lighting3 = 58 Lighting4 = 84 Lighting5 = 46 Lighting6 = 100 Lighting7 = 67 Lighting8 = 79 Lighting9 = 52 Lighting10 = 95	The LCD displays “Lighting 74% - BRIGHT” Motor for Shade turns on for 10 seconds	The LCD Displayed “Lighting 80% - BRIGHT”	When the light level is 46% or higher, the LCD displays “Lighting – BRIGHT,” indicating accurate light detection, and the shade turns on
7	SoilMoisture1 = 12 SoilMoisture2 = 27 SoilMoisture3 = 5 SoilMoisture4 = 30 SoilMoisture5 = 18 SoilMoisture6 = 9 SoilMoisture7 = 22 SoilMoisture8 = 16 SoilMoisture9 = 33 SoilMoisture10 = 11	The LCD displays “Soil Moisture 18% - DRY” Relay for light bulb (a.k.a. sprinkler) turn on	The LCD displayed “Soil Moisture 26% - DRY”	When the soil moisture is below 35%, the LCD displays “Soil Moisture – DRY,” and the light bulb turns on.
8	SoilMoisture1 = 63 SoilMoisture2 = 47 SoilMoisture3 = 70 SoilMoisture4 = 55 SoilMoisture5 = 41 SoilMoisture6 = 68 SoilMoisture7 = 60 SoilMoisture8 = 39 SoilMoisture9 = 75 SoilMoisture10 = 52	The LCD displays “Soil Moisture 57% - NORMAL”	The LCD displayed “Soil Moisture 60% - NORMAL”	When the soil moisture is between 36% and 75%, the LCD displays “Soil Moisture – NORMAL,” and the light bulb turns off.
9	SoilMoisture1 = 92 SoilMoisture2 = 81 SoilMoisture3 = 99 SoilMoisture4 = 84 SoilMoisture5 = 77 SoilMoisture6 = 95 SoilMoisture7 = 88 SoilMoisture8 = 79 SoilMoisture9 = 100 SoilMoisture10 = 93	The LCD displays “Soil Moisture 88% - WET”	The LCD displays “Soil Moisture 83% - WET”	When the soil moisture is 75% or higher, the LCD displays “Soil Moisture – WET,” and the light bulb remains off.
10	Reset Button is pressed	System resets The LCD Displays “System Reset” The system returns to Active status	The LCD displays “System Reset,” and the system returns to active status.	The LCD shows Resetting status for 3 seconds, after which the system returns to active status.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 5.1 SMART Greenhouse Monitoring System TinkerCAD Test Case 1 Result

If the temperature sensor detects 17°C or lower, the LCD displays “Temperature - COLD,” confirming correct detection, with the fan remaining off.

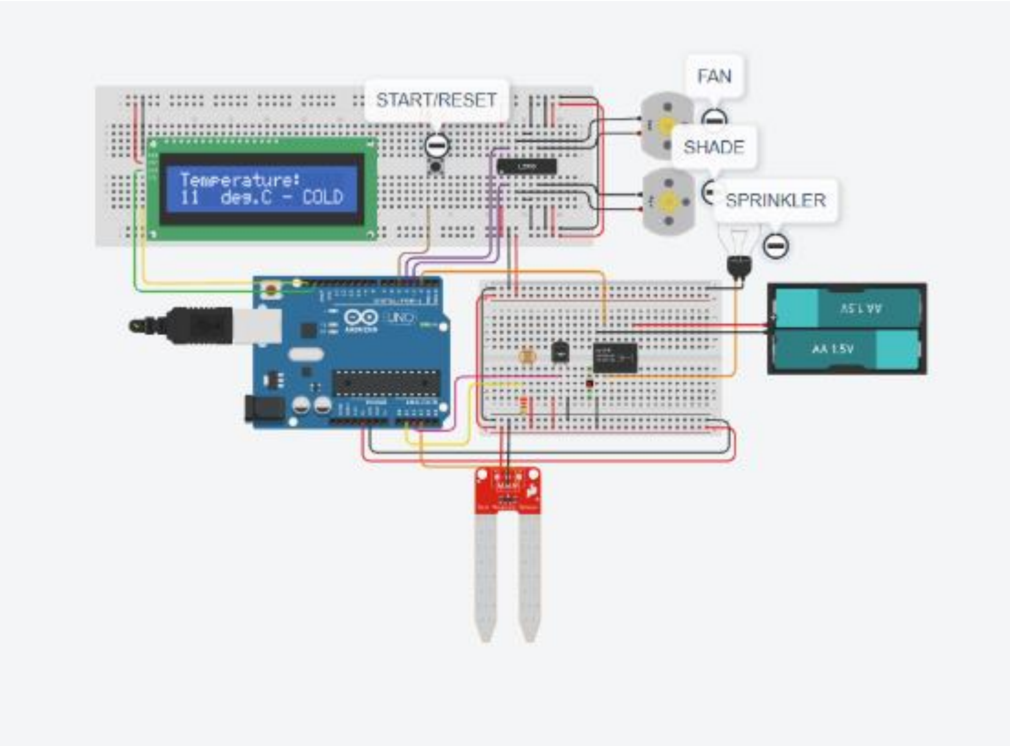
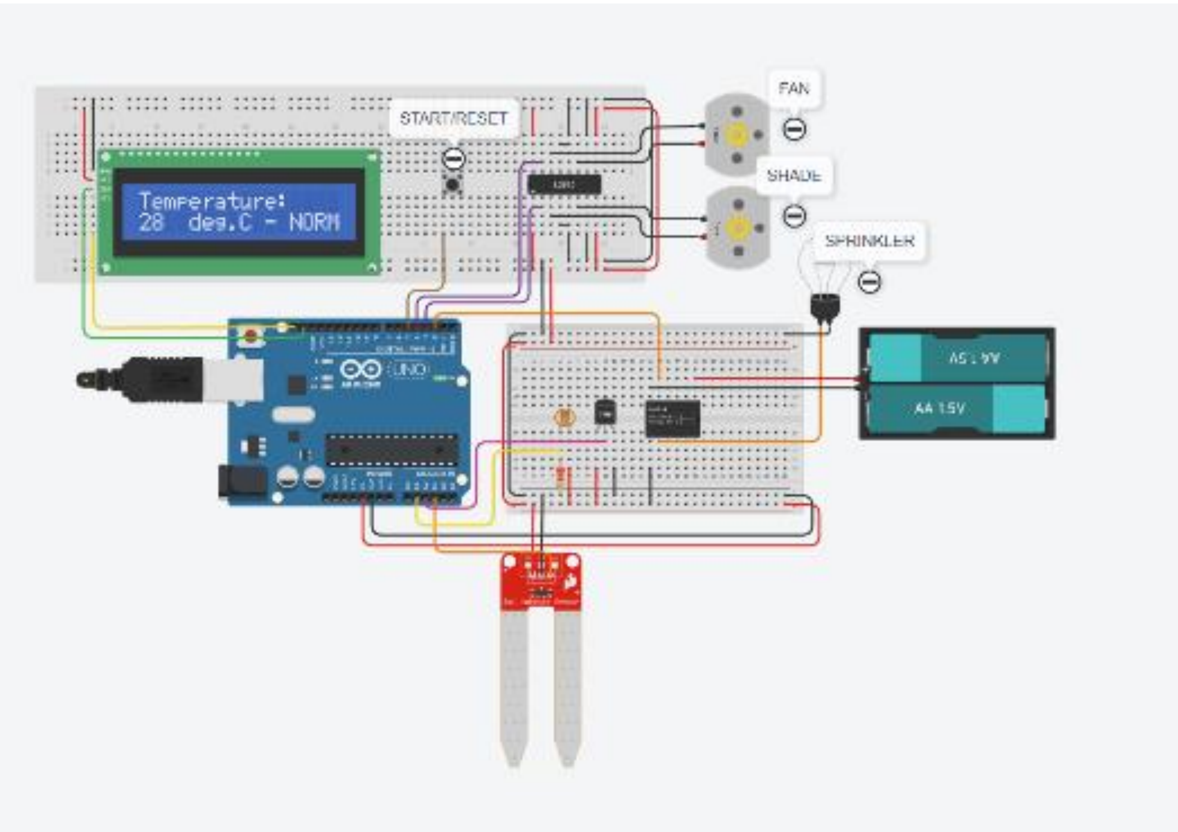


Figure 5.2 SMART Greenhouse Monitoring System TinkerCAD Test Case 2 Result

If the temperature sensor reads from 18°C to 28°C, the LCD displays “Temperature - NORMAL,” confirming correct detection, while the fan remains off.



ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 5.3 SMART Greenhouse Monitoring System TinkerCAD Test Case 3 Result

If the temperature sensor reads 29°C or above, the LCD shows "Temperature: 48°C – HOT," indicating correct detection, and the fan turns on.

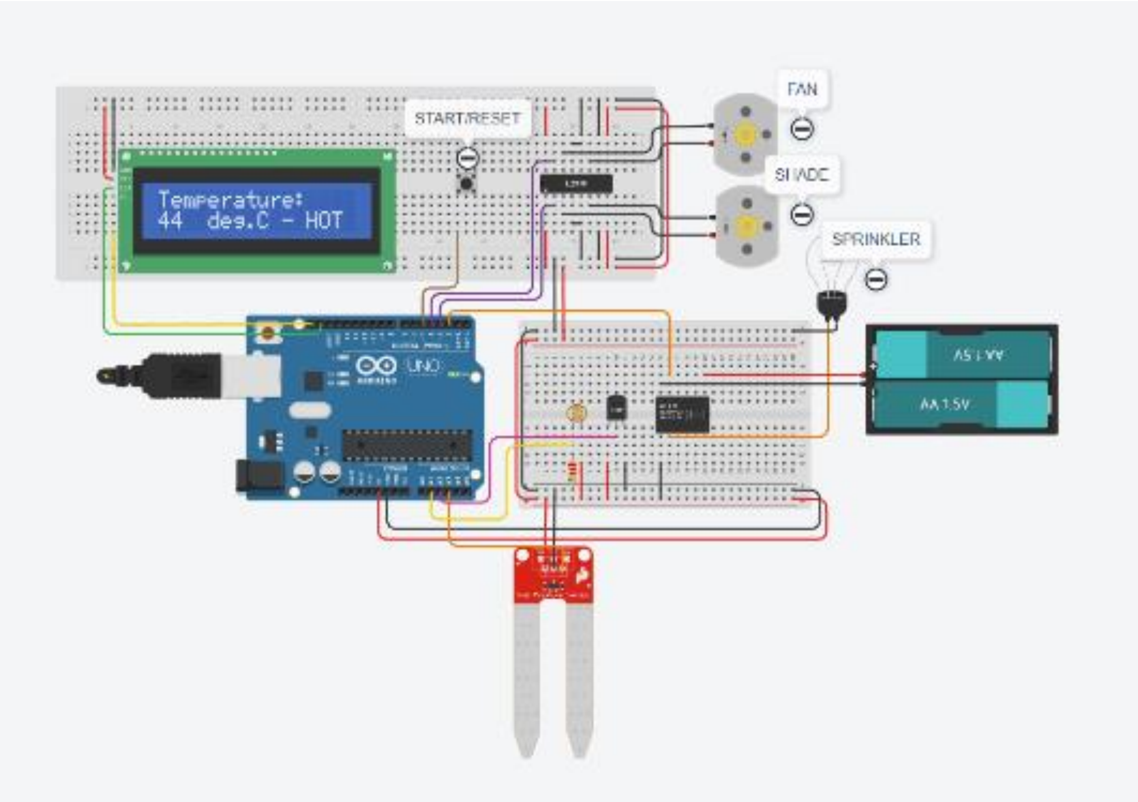
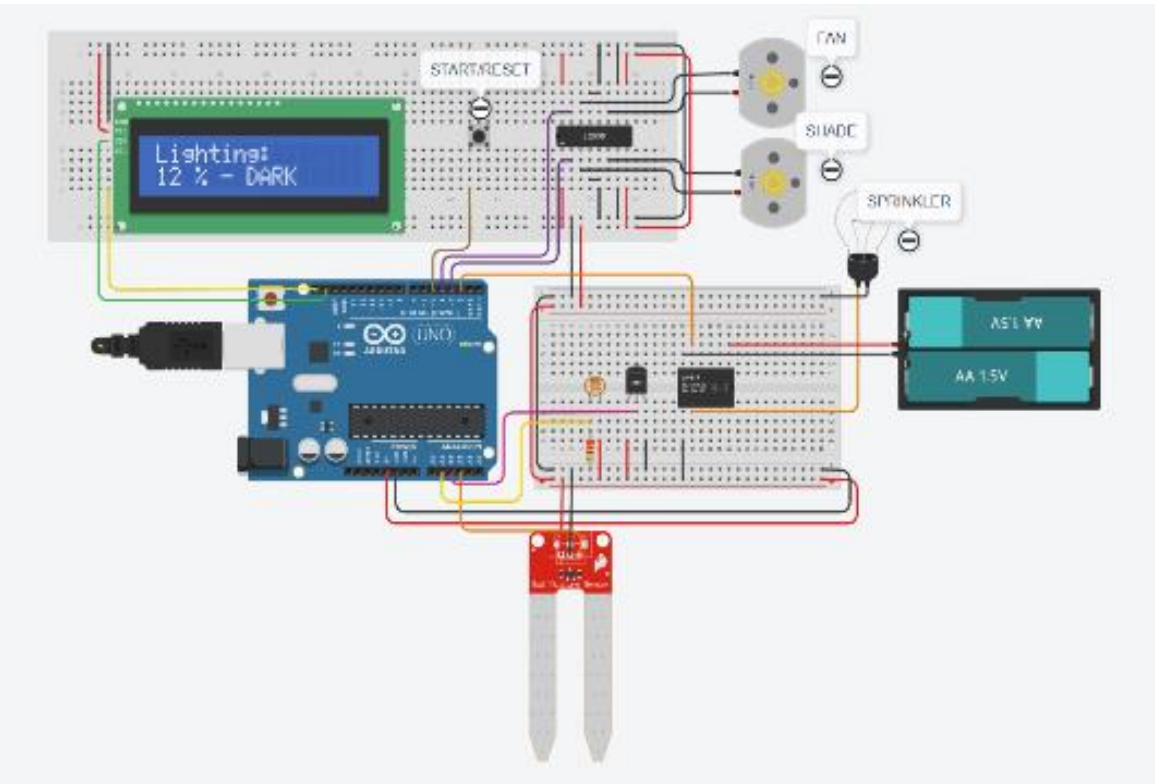


Figure 5.4 SMART Greenhouse Monitoring System TinkerCAD Test Case 4 Result

If the light level is below 25%, the LCD shows "Lighting – DARK," indicating accurate light detection, and the shade remains off.



ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 5.5 SMART Greenhouse Monitoring System TinkerCAD Test Case 5 Result

When the light level falls below 25%, the LCD displays “Lighting – DARK,” indicating correct light detection, and the shade remains off.

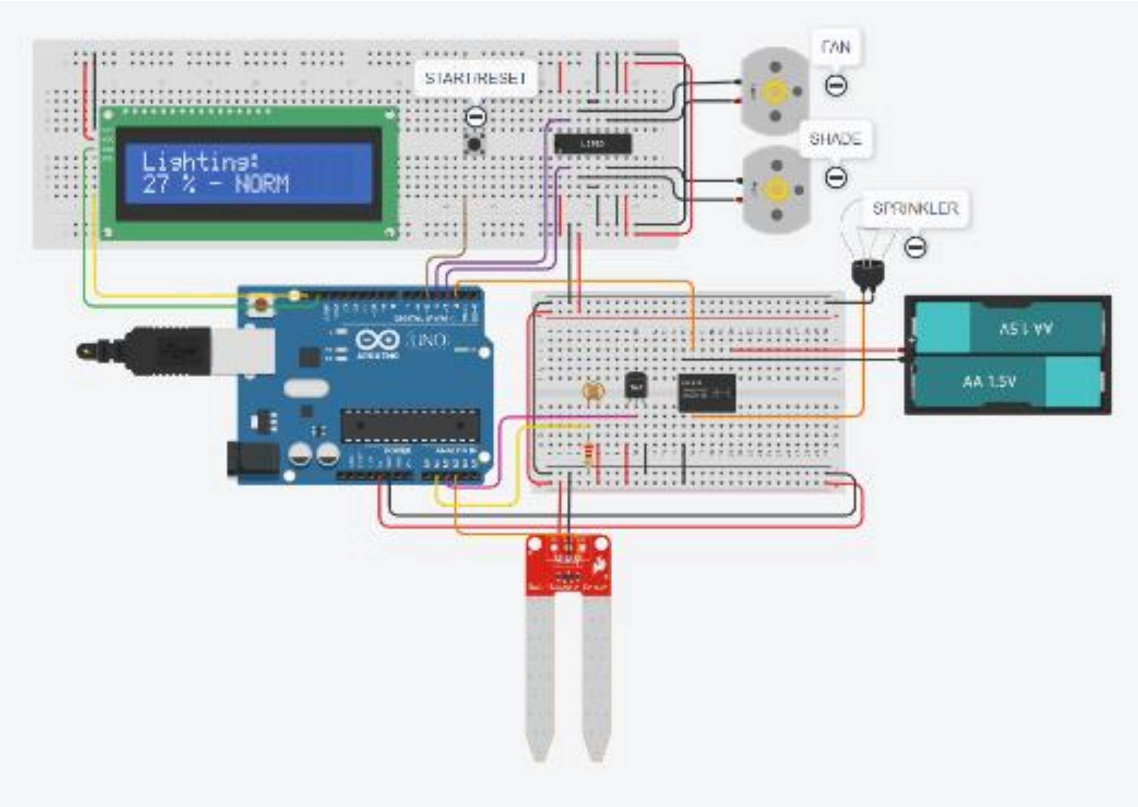
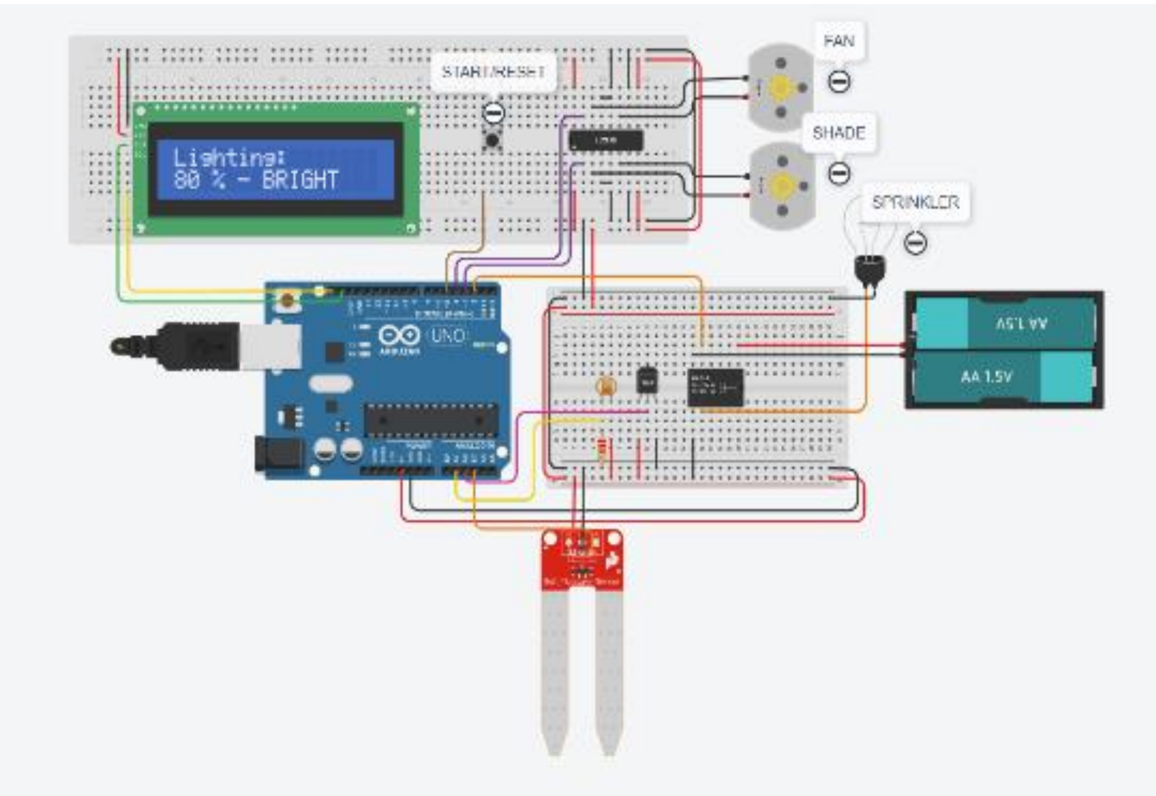


Figure 5.6 SMART Greenhouse Monitoring System TinkerCAD Test Case 6 Result

When the light level reaches 46% or higher, the LCD shows “Lighting – BRIGHT,” indicating correct light detection, and the shade turns on.



ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 5.7 SMART Greenhouse Monitoring System TinkerCAD Test Case 7 Result

If the soil moisture drops below 35%, the LCD shows "Soil Moisture – DRY," and the light bulb turns on.

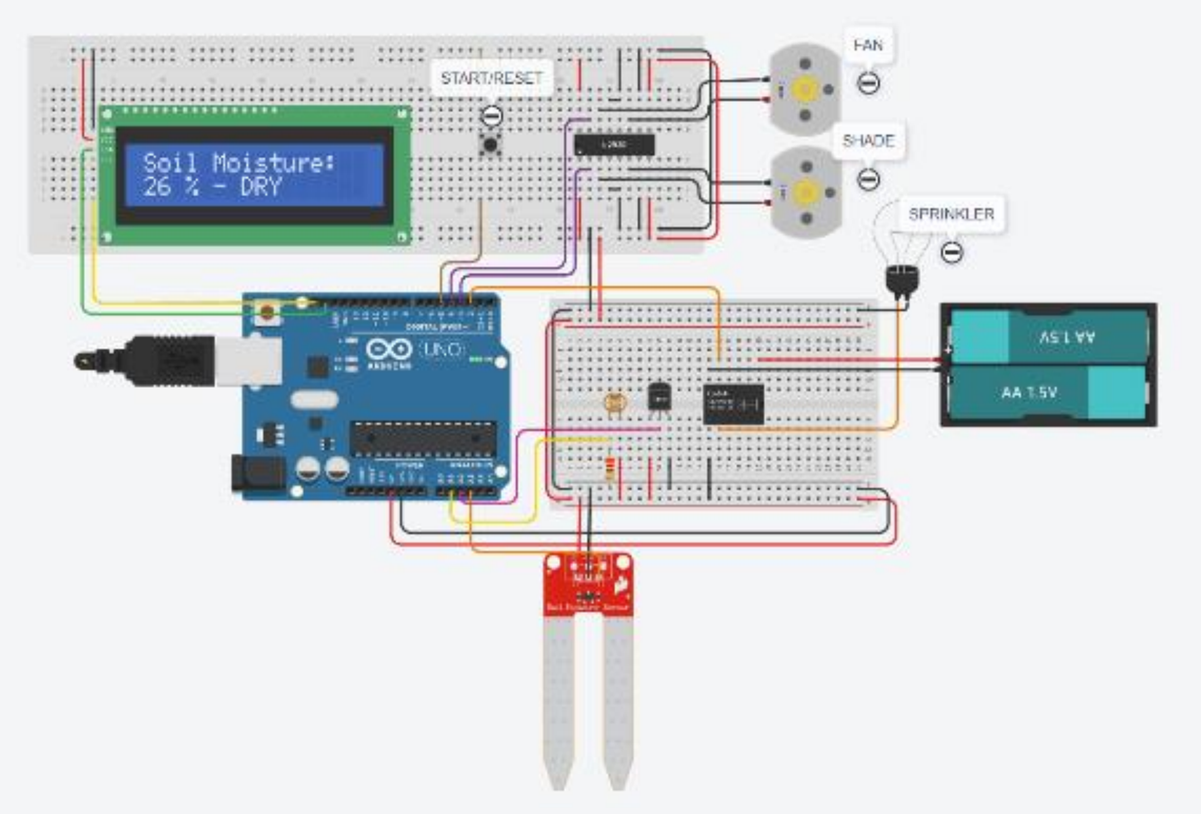
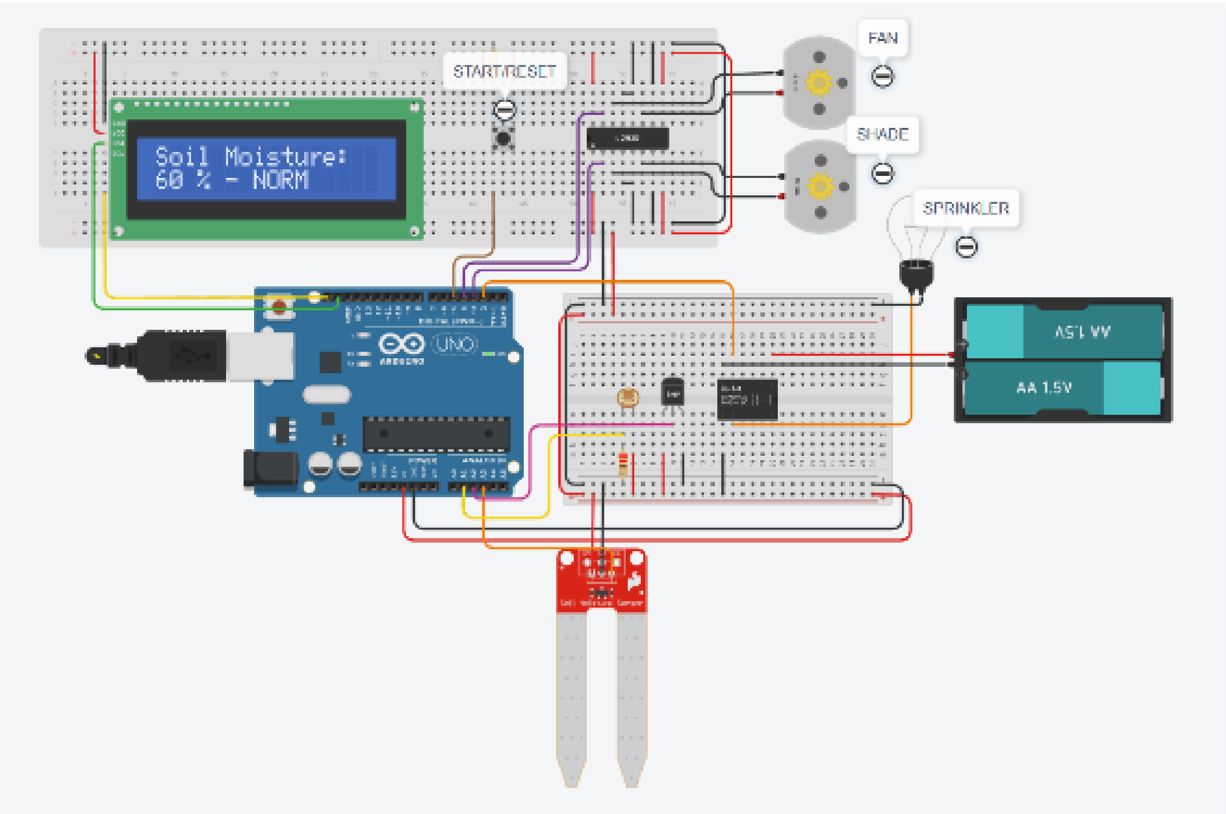


Figure 5.8 SMART Greenhouse Monitoring System TinkerCAD Test Case 8 Result

If the soil moisture is between 36% and 75%, the LCD shows "Soil Moisture – NORMAL," and the light bulb turns off.



ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume IX Issue XI November 2025

Figure 5.9 SMART Greenhouse Monitoring System TinkerCAD Test Case 9 Result

If the soil moisture is 75% or above, the LCD shows "Soil Moisture – WET," and the light bulb remains off.

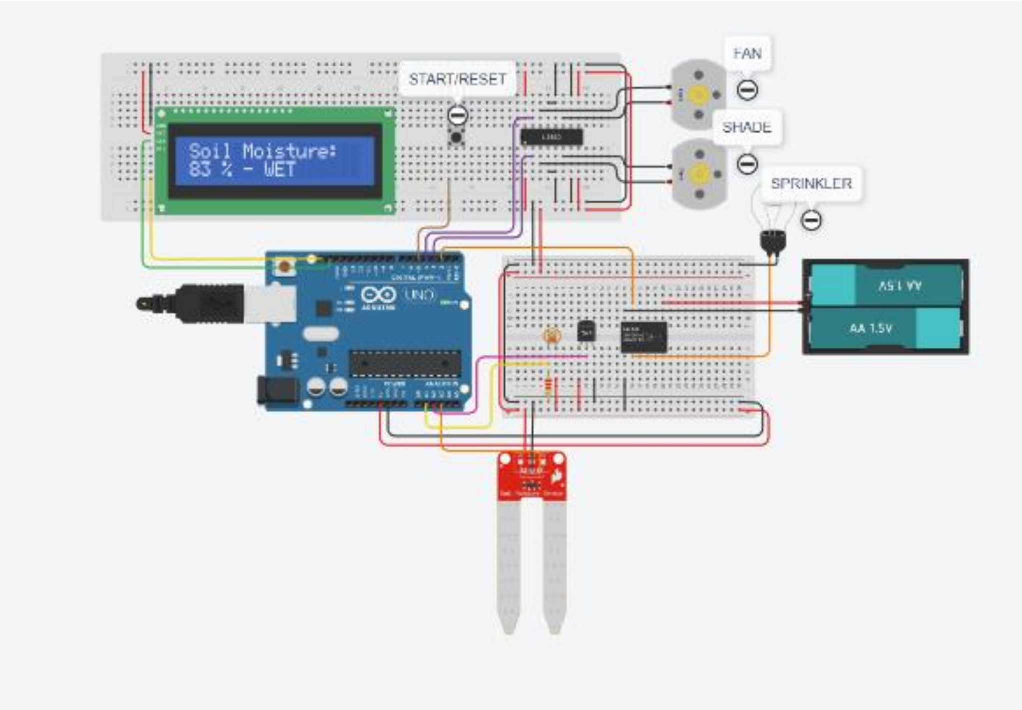
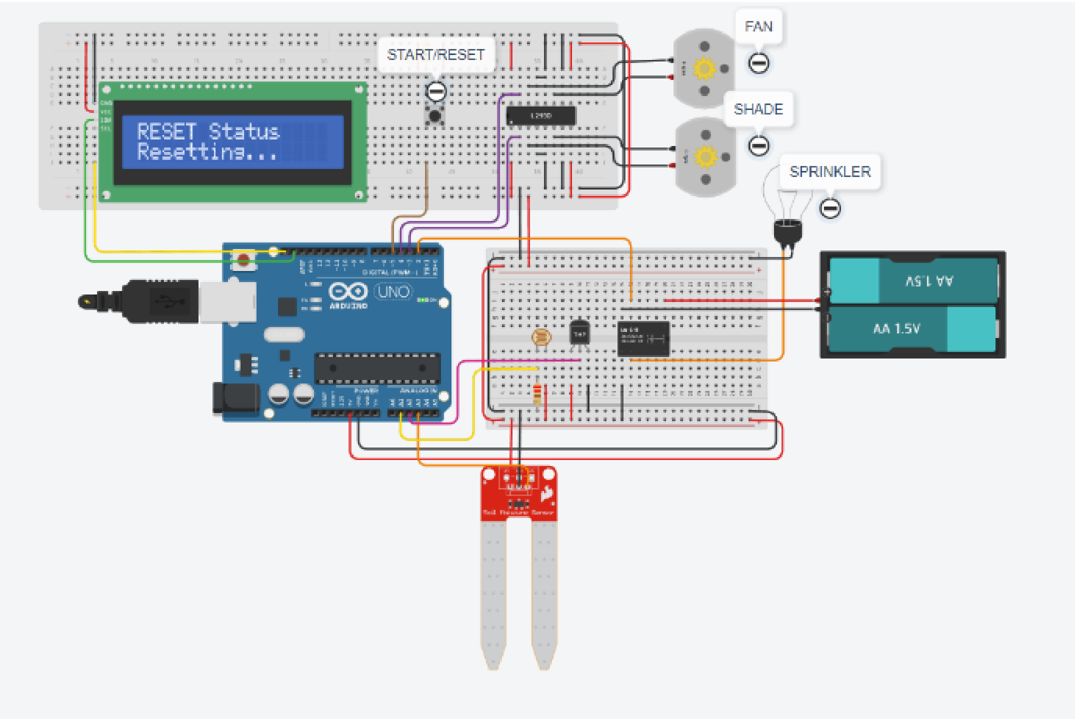


Figure 5.10 SMART Greenhouse Monitoring System TinkerCAD Test Case 10 Result

The system reset for 3 seconds and the system returned to active status.



CONCLUSION AND RECOMMENDATIONS

The simulated Greenhouse Monitoring System successfully demonstrated the fundamental concept of monitoring and managing environmental factors such as temperature, humidity, and soil moisture using an Arduino-based setup in TinkerCAD. Although the system does not utilize real-time data, the simulation effectively illustrated how various components interact and respond to different conditions. On the other hand, due to TinkerCAD Code Block's limited features and capabilities, the developers were constrained to developing the fundamental features only. Features such as threshold control, and independent timers for each actuator, were not feasible since the developers reached the maximum allowable number of blocks and pin numbers.

It is recommended that future enhancements focus on developing a more integrated control logic that allows multiple sensors to work together for coordinated system responses. Incorporating simultaneous sensor data processing would improve accuracy, efficiency, and automation, resulting in a more realistic and adaptive greenhouse monitoring system suitable for practical implementation.

REFERENCES

1. Dolom, P., Calimag, C., Nicmic1, J., Eleazar1, P., Casin, Ma., Villanueva, Ma., Garcia, D., & Buot, M. (2023). *Assessment of climate change vulnerabilitiesof upland vegetable farmers in selectedareas in Benguet, Philippines.* https://ovcre.uplb.edu.ph/journals-uplb/ index.php/EDJ/article/view/954/807
2. Akpulonu,V., Agbese, A., Obizue C., Naterm A., Abdulsalam, N., Ogochukwu, I., Aminu-Baba, M., & Ene, A. (2024, July 9). *Design and construction of Arduino based greenhouse monitoring system using IoT.* https://wjaets.com/sites/default/files/WJAETS-2024-0280.pdf
3. Patil, M., Patil, T., Paradhi, V., Patil, J., & Naz, F. (2024, May). *Arduino UNO-based Internet of Things greenhouse monitoring and control system.* https://tijer.org/tijer/papers/TIJER2405229.pdf
4. Hoque, M., Ahmed, M. & Hannan, S. (2020, April). *An automated greenhouse monitoring and controlling system using sensors and solar power.* https://www.researchgate.net/publication/341031255_An_Automated_Greenhouse_Monitoring_and_Controlling_System_using_Sensors_and_Solar_Power