



People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific
Research

University Echahid Hamma Lakhdar – El Oued



End-of-Study Project Memorandum

Submitted in partial fulfillment of the requirements for the degree of

ACADEMIC MASTER

Field: Mathematics and Computer Science

Branch: Computer Science

Specialty: Internet of Things and Cybersecurity

An Intelligent Irrigation Monitoring System Based on Soil Moisture and Temperature

Prepared by

Ba Ali

Ghayout Ahmed Ben Salem

Tidjani Mohammed Lghrisi

Dr. RETIMA Farida:	University of El-Oued	- President
Dr. HIMA Abdelkader:	University of El-Oued	- Examiner
Dr. BERHOUM Adel:	University Center of Barika	- Supervisor
Dr. SACI Medileh:	University of El-Oued	- Supervisor

Academic Year 2025/2026

Acknowledgements

First and foremost, we thank Almighty Allah for granting us the strength, patience, and perseverance to complete this work.

We would like to express our sincere gratitude to our supervisors, Dr. BERHOUM Adel and Dr. SACI Medileh, for their valuable guidance, continuous support, helpful advice, and encouragement throughout the preparation of this project.

We also extend our respectful thanks to the members of the jury, Dr. RETIMA Farida and Dr. HIMA Abdelkader, for accepting to examine and evaluate this work. Their remarks and suggestions are highly appreciated.

Our sincere thanks go to all the teachers and staff of the Department of Mathematics and Computer Science at the University of Echahid Hamma Lakhdar – El Oued, who contributed to our academic training during our years of study.

We are deeply grateful to our families for their love, prayers, sacrifices, and constant encouragement. Their support has been a major source of motivation throughout this journey.

Finally, we thank all our friends, classmates, and everyone who helped us, directly or indirectly, in completing this End-of-Study Project Memorandum.

Dedication

I dedicate this work to my beloved parents, whose endless love, prayers, patience, and sacrifices have been the foundation of every success I have achieved.

To my family, for their constant encouragement, understanding, and support throughout this journey.

To my teachers and supervisors, whose guidance, knowledge, and advice helped shape this project and strengthen my academic path.

To my friends and colleagues, who shared with me the challenges, efforts, and memorable moments of this experience.

Finally, I dedicate this humble work to everyone who believed in me and supported me, directly or indirectly, until the completion of this project.

Abstract

The rapid advancement of the Internet of Things (IoT) has generated significant opportunities to enhance agricultural productivity through intelligent monitoring and automation. This study details the design and implementation of a smart greenhouse monitoring and control system that autonomously regulates environmental conditions critical for plant growth. The system continuously monitors temperature, humidity, and soil moisture to sustain optimal greenhouse conditions. The proposed system integrates DHT22 and soil moisture sensors with an ESP32 microcontroller to collect and process environmental data in real time. Communication utilizes Wi-Fi connectivity and the MQTT protocol, ensuring reliable data transmission and remote accessibility. Based on predefined threshold values, the system autonomously controls irrigation and ventilation mechanisms, including pumps, fans, and window actuators. A web-based dashboard offers real-time visualization of sensor data and facilitates remote monitoring and control. The system was validated through simulation and practical implementation. Experimental results demonstrated successful real-time monitoring and automatic control of environmental parameters, with a response time of approximately 2–3 seconds. The developed solution improves irrigation efficiency, reduces water consumption and manual intervention, and provides a scalable platform for future integration of artificial intelligence techniques to enhance agricultural productivity and decision-making.

Keywords: Smart Greenhouse, Internet of Things (IoT), ESP32, Soil Moisture Monitoring, MQTT Protocol, Intelligent Irrigation System.

المخلص

أتاح التطور السريع لإنترنت الأشياء IoT فرصاً واعدة لتحسين الإنتاجية الزراعية من خلال أنظمة المراقبة والأتمتة الذكية. يهدف هذا المشروع إلى تصميم وتنفيذ نظام ذكي لمراقبة البيوت المحمية والتحكم فيها، بما يضمن الإدارة الآلية للظروف البيئية الضرورية لنمو النباتات. ويركز النظام على مراقبة درجة الحرارة والرطوبة ورطوبة التربة بهدف الحفاظ على بيئة زراعية مثالية داخل البيت المحمي.

يعتمد الحل المقترح على دمج مستشعرات DHT22 ومستشعرات رطوبة التربة مع المتحكم الدقيق ESP32، الذي يتولى جمع البيانات البيئية ومعالجتها في الزمن الحقيقي. ويتحقق الاتصال بين مكونات النظام عبر شبكة Wi-Fi وبروتوكول MQTT، مما يضمن نقلاً موثقاً للبيانات وإمكانية الوصول إليها عن بُعد. واستناداً إلى قيم عتبية محددة مسبقاً، يتحكم النظام تلقائياً في آليات الري والتهوية، بما في ذلك المضخات والمراوح ومحركات فتح النوافذ. كما توفر لوحة تحكم ويب عرضاً لحظياً لقراءات المستشعرات، وتدعم عمليات المراقبة والتحكم عن بُعد.

تم التحقق من فعالية النظام من خلال المحاكاة والتطبيق العملي، حيث أظهرت النتائج نجاحه في المراقبة الفورية والتحكم الآلي في المعايير البيئية بزمان استجابة يقارب 2-3 ثوانٍ. ويسهم النظام المطور في تحسين كفاءة الري، وتقليل استهلاك المياه والتدخل البشري، كما يوفر منصة قابلة للتطوير لدمج تقنيات الذكاء الاصطناعي مستقبلاً بهدف تعزيز الإنتاجية الزراعية ودعم اتخاذ القرار.

الكلمات المفتاحية: البيت المحمي الذكي، إنترنت الأشياء، ESP32، مراقبة رطوبة التربة، بروتوكول MQTT، نظام الري الذكي.

Contents

Abbreviations	xi
General Introduction	1
1 Generalities	3
1.1 Agricultural greenhouses	4
1.1.1 What is the greenhouse?	4
1.1.2 Types of Greenhouses	4
1.1.2.1 Tunnel greenhouses	4
1.1.2.2 Horticultural greenhouses	5
1.1.3 Components of a Greenhouse	5
1.2 Internet of Things (IoT)	7
1.2.1 Definition of IoT:	7
1.2.2 Architecture of IoT	7
1.2.3 IoT Technologies	8
1.2.4 applications in Greenhouse Agriculture	8
1.3 Challenges of IoT in Smart Agriculture	10
1.4 Objectives and Contributions	11
1.5 Conclusion	12
2 System's Architecture	13
2.1 Overview of the Proposed System	14
2.2 System Lifecycle	15
2.3 system data flow	17
2.4 UML Modeling	19
2.4.1 Use Case Diagram	19
2.4.2 Class Diagram	19

2.4.3	Sequence Diagram	20
2.4.4	State Machine Diagram	21
3	Methodology	24
3.1	System Circuit Design	25
3.1.1	Circuit Diagram Description	25
3.1.2	Component Connections	25
3.2	System Simulation (Tinkercad)	27
3.3	Hardware Components	27
3.3.1	Microcontroller (ESP32)	27
3.3.2	Environmental Sensors	28
3.3.2.1	Temperature and Humidity Sensor	28
3.3.2.2	Soil Moisture Sensor	29
3.3.3	Actuators	30
3.3.3.1	Irrigation Pump	30
3.3.3.2	Electric valve	31
3.3.3.3	Ventilation Fan	32
3.3.3.4	Window Motor	32
3.4	Communication and Data Transmission	33
3.4.1	Wi-Fi Connectivity	33
3.4.2	MQTT Communication Protocol	33
3.5	Software Architecture	35
3.5.1	Embedded System Programming	35
3.5.2	Control Logic	35
3.5.2.1	Controlling the temperature according to the DHT22 tem- perature sensing:	35
3.5.2.2	Controlling the Humidity according to the dht11 temper- ature sensing:	35
3.5.2.3	Soil moisture measurement and Irrigation Pump control diagram	36
3.5.3	Monitoring Interface / Dashboard	36
3.6	Conclusion	39

4	Experimental Results	40
4.1	Experimental Setup	41
4.2	Sensor Data Monitoring	43
4.3	Ventilation System Results	44
4.4	Irrigation System	45
4.5	Real-Time Monitoring Results	47
4.6	System Response Time	47
	Conclusions	49
	References	52
	Appendix A: Source Code Repository	53

List of Figures

1.1	Tunnel greenhouses	4
1.2	Horticultural greenhouses	5
1.3	General Architecture of the Internet of Things	8
1.4	Applications of IoT in Greenhouse Agriculture	9
2.1	Smart greenhouse system lifecycle	15
2.2	Data flow architecture of the system	18
2.3	Use Case Diagram of the system	19
2.4	Class Diagram of the system	20
2.5	Sequence Diagram of the system	21
2.6	State machine diagram of the system	22
3.1	System Circuit Design	25
3.2	The development board ESP32-DevKitC	28
3.3	Humidity and Temperature Sensor	29
3.4	Soil Moisture Sensor	30
3.5	Irrigation Pump	31
3.6	Electric valve	31
3.7	Ventilation Fan	32
3.8	Window Motor	33
3.9	MQTT Protocol Movement	34
3.10	DHT22 Sensor and Temp Control	35
3.11	DHT22 Sensor and Humidity Control	36
3.12	Soil Moisture measurement and Irrigation Pump	36
3.13	The main page	37
3.14	Login Page	37
3.15	dashboard	37

3.16	Sensor state page	38
3.17	Analytics page	38
3.18	Control page	38
4.1	Experimental Setup of the Smart Greenhouse System	41
4.2	Main Controller	42
4.3	Serial Monitor	43
4.4	Result of temp&hum before	44
4.5	Result of temp&hum after	44
4.6	Ventilation System Using Fan and Window	45
4.7	Result of Soil Moisture	45
4.8	Automatic Watering System	46
4.9	Real-Time Monitoring Results	47

List of Tables

3.1	Serial communication between Arduino and ESP32	25
3.2	Relay module connections with Arduino	25
3.3	Pump wiring through relay	26
3.4	Fan wiring through relay	26
3.5	Window motor wiring through relay	26
3.6	DHT22 sensor connections with ESP32	26
3.7	Soil moisture sensors connections with ESP32	26
3.8	Properties Comparison of IoT Communication Protocols [1]	34

Abbreviations

CoAP	Constrained Application Protocol
DHT22	Digital Humidity and Temperature Sensor 22
ESP32	Espressif 32-bit Microcontroller
ESP8266	Espressif Wi-Fi Microcontroller Module
GPIO	General Purpose Input/Output
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit
IBM	International Business Machines
IETF	Internet Engineering Task Force
IoT	Internet of Things
IPSO	Internet Protocol for Smart Objects
LoRa	Long Range Wireless Communication
MQTT	Message Queue Telemetry Transport
NC	Not Connected
NO	Normally Open
OMA	Open Mobile Alliance
PE	Polyethylene
PWM	Pulse Width Modulation
RFID	Radio Frequency Identification
SPI	Serial Peripheral Interface
TCP	Transmission Control Protocol
TDR	Time Domain Reflectometry
UDP	User Datagram Protocol
UML	Unified Modelling Language
UART	Universal Asynchronous Receiver/Transmitter

VCC	Voltage Common Collector
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network

General Introduction

Agriculture plays a fundamental role in ensuring food security and supporting economic development worldwide. However, traditional agricultural practices often face significant challenges related to climate variability, inefficient resource utilization, water scarcity, and the increasing demand for higher crop productivity. Greenhouses have emerged as an effective solution for creating controlled environments that enhance plant growth and protect crops from adverse weather conditions. Nevertheless, conventional greenhouse management still relies heavily on manual monitoring and intervention, which may lead to delayed responses, excessive water consumption, and reduced operational efficiency. Recent advancements in the Internet of Things (IoT) have opened new opportunities for transforming traditional agriculture into smart agriculture. IoT technologies enable the continuous collection, transmission, and analysis of environmental data through interconnected sensors, communication networks, and intelligent control systems. By integrating sensing devices with automated decision-making mechanisms, greenhouse conditions can be monitored and adjusted in real time, improving productivity while reducing resource consumption. Despite the considerable progress achieved in smart greenhouse systems, several limitations remain. Many existing solutions are expensive, difficult to deploy, or focus on monitoring without providing intelligent control capabilities. Furthermore, some systems lack real-time remote access, scalability, or efficient communication mechanisms suitable for agricultural environments. These limitations highlight the need for cost-effective and reliable solutions capable of combining environmental monitoring, automated control, and remote management within a unified platform. The main problem addressed in this work can be formulated as follows: How can an intelligent and low-cost IoT-based system be designed and implemented to monitor and control greenhouse environmental conditions in real time while improving irrigation efficiency and reducing human intervention? To address this problem, this project aims to design and develop a smart greenhouse monitoring and control system based on IoT technologies. The pro-

posed system integrates environmental sensors, communication protocols, and automated control mechanisms to regulate critical parameters such as temperature, humidity, and soil moisture. The main objectives of this work are:

Design a smart greenhouse architecture based on IoT technologies.

Monitor environmental parameters in real time using temperature, humidity, and soil moisture sensors.

Implement automated irrigation and ventilation control mechanisms.

Develop a web-based dashboard for remote monitoring and management.

Ensure reliable communication using the MQTT protocol and Wi-Fi connectivity.

Improve water management efficiency and reduce manual intervention.

The main contributions of this project can be summarized as follows:

Development of a complete IoT-based greenhouse monitoring and control platform.

Integration of ESP32 microcontroller technology with environmental sensing devices.

Implementation of automatic irrigation and ventilation control strategies based on predefined thresholds.

Design of a real-time monitoring dashboard for remote supervision and system management.

Validation of the proposed solution through simulation and experimental testing.

This dissertation is organized into five chapters: Chapter 1: General Background and Literature Review presents the fundamental concepts related to smart agriculture, greenhouses, Internet of Things technologies, communication protocols, and existing research works. Chapter 2: System Analysis and Design describes the proposed solution, system requirements, overall architecture, data flow, and UML modeling used to represent the system structure and behavior. Chapter 3: Methodology and Implementation details the hardware and software components, communication technologies, circuit design, control logic, and dashboard development. Chapter 4: Experimental Results and Evaluation presents the implementation process, experimental setup, obtained results, system performance evaluation, and discussion of findings. Chapter 5: Conclusions and Future Work summarizes the achievements of the project, highlights its limitations, and proposes future improvements and research directions for enhancing smart greenhouse systems.

Chapter 1

Generalities

Agriculture helps ensure food security and economic development, and greenhouses offer an ideal solution. However, the main challenge lies in precisely managing the greenhouse environment. Controlling temperature, humidity, and light is critical, as fluctuations can significantly affect plant growth and crop quality.

With the growing population, it has become essential to integrate Internet of Things (IoT) technology into agriculture to boost production and adopt effective strategies to maintain optimal, stable conditions for plant growth throughout their life cycle, especially in harsh environments.

This chapter explains in detail the concept of greenhouse systems and the Internet of Things, along with its applications. This study will help understand how the Internet of Things can improve the performance and productivity of greenhouse systems in desert environments.

1.1 Agricultural greenhouses

Agricultural greenhouses are structures that facilitate controlled, year-round crop cultivation and can achieve yields approximately twice those of open-field agriculture.

1.1.1 What is the greenhouse?

The greenhouse is a small space which can be between 1 m² up to 100 m², it is in generally closed or partially open, and generally intended for agricultural production, or an enclosure intended for the and protection of plants by exploiting radiation solar it has become an industrial premises for the production of plant matter where we attempts to adapt the immediate environment of the plant, so as to improve its productivity and its quality, freeing it from the external climate, the local soil and even seasons [2].

1.1.2 Types of Greenhouses

1.1.2.1 Tunnel greenhouses

In general, the tunnel greenhouse consists of a series of juxtaposed elements each made up of a steel tube frame and profiles assembled by bolts. The width is (3 to 9) m. The plastic film is fixed by various systems of clips which trap the film against the profile or between two rods throughout the greenhouse. The classic tunnel greenhouse is rounded, but there are also models with straight legs as for glass greenhouses, Some of them are also convertible for their covers [3].



Figure 1.1: Tunnel greenhouses

1.1.2.2 Horticultural greenhouses

The chapel is the basic construction unit of a greenhouse. It is formed by two vertical side walls (or slightly inclined) and a roof with two slopes, generally symmetrical. The chapel is characterized by its width, with common dimensions approximately 3, 6, 9, 12, and 16 meters. When two consecutive chapels are not separated by an internal vertical wall, the structure is called a multi-chapel greenhouse or twin chapels.

The truss is the main load-bearing structural element of the chapel and is repeated at regular intervals. The distance between the trusses is characteristic of the greenhouse type. The module is a characteristic surface of the greenhouse obtained by multiplying the width of the chapel by the distance between two trusses. The gables are the vertical walls forming the two ends of a chapel. The ridge is the line formed by the top of the chapel.

The portal frame is the load-bearing structure that, in older greenhouses, consisted of the truss legs and a beam connecting the tops of the vertical posts. However, this type of structure is becoming less common. The uprights are the vertical side walls of a chapel [3].



Figure 1.2: Horticultural greenhouses

1.1.3 Components of a Greenhouse

A Framing Materials:

greenhouses may be constructed from several different materials. Among the most popular are aluminum and wood.

- **Aluminum:** It is the most widely used because it is more economical and safer due to its durability and resistance to corrosion. They allow the construction of precise, stable and easy-to-form greenhouse structures
- **Wood:** Unlike aluminum, wood is less commonly used and relatively inexpensive because it cannot extend for a long time and is susceptible to rotting and corrosion. It is usually used in small or traditional greenhouses.

B Covering Materials:

greenhouse coverings must be clear enough to provide optimum light transmission and at the same time be durable as well as economical.

- **Glass:** Glass is considered one of the best materials used in building greenhouses because it provides excellent light transmission and acceptable durability, which is useful for plants to photosynthesise, but glass structures require high cost and sturdy supports.
- **Double sheets of polyethylene (PE) film inflated with air:** This cover is widely used in greenhouse structures because it is economical due to its low cost and flexibility. It consists of two air-insulated layers for thermal insulation that helps maintain the greenhouse climate.

1.2 Internet of Things (IoT)

1.2.1 Definition of IoT:

There is no universally accepted definition of IoT. In fact, many groups, including academicians, researchers, practitioners, innovators, developers, and corporate professionals, have defined the term, although its initial use has been attributed to Kevin Ashton, an expert in digital innovation. What all of the definitions have in common is the idea that the first version of the Internet was about data created by people, while the next version is about data created by things. The best definition for IoT would be the following:

“An open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting in face of situations and changes in the environment”[4].

1.2.2 Architecture of IoT

A critical requirement of an IoT is that the things in the network must be inter-connected. IoT system architecture must guarantee the operations of IoT, which bridges the gap between the physical and the virtual worlds. Design of IoT architecture involves many factors such as networking, communication, business models and processes, and security. In designing the architecture of IoT, the extensibility, scalability, and interoperability among heterogeneous devices and their models should be taken into consideration. Due to the fact that things may move physically and need to interact with each other in real-time mode, IoT architecture should be adaptive to make devices interact with other things dynamically and support unambiguous communication of events [5].

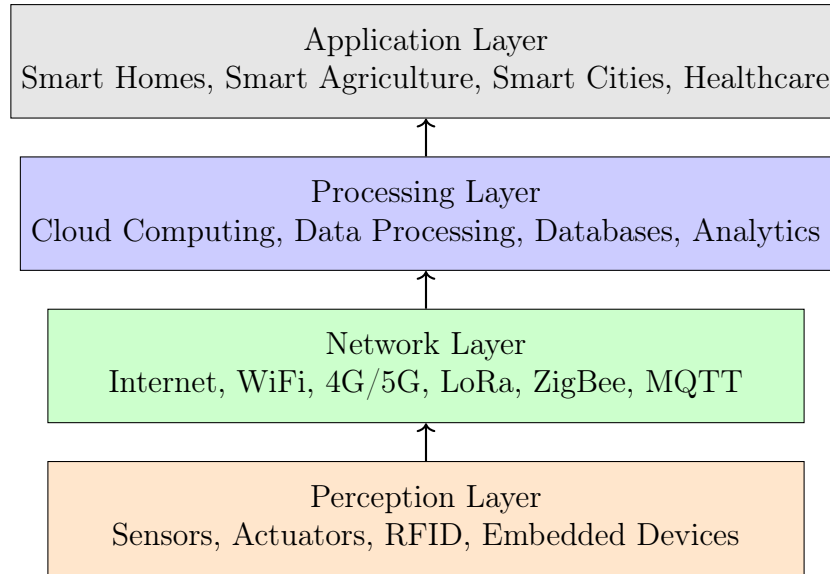


Figure 1.3: General Architecture of the Internet of Things

1.2.3 IoT Technologies

The smart greenhouse system has the Internet of Things that connects devices. These devices collect data from the environment. We have three elements in the system:

- **Sensors:** it is an iot technology that collects data from the surrounding environment, they convert the physical data to digital for example "DHT22", "MQ-6".
- **Actuators:** are like the systems output. They get commands from the Microcontroller then execute it. for example "servo motor", "buzzer".
- **Microcontroller:** processes the data from the sensors. It analyzes the data. Compares it with old values. Then it makes a decision. it is like the brain of the system. It makes all the all decisions. for example "Arduino", "Esp32".

1.2.4 applications in Greenhouse Agriculture

The advancement of IoT technology empowers things to be regulated remotely. Instead of controlling and monitoring a greenhouse farm manually, users can observe the day-by-day progress of cultivated crops by using IoT sensors/devices and mobile applications remotely. Several applications have been developed for making the farmers capable of efficiently controlling their greenhouse farms. [6].

- **Monitoring:** In greenhouses, the crop is grown under a controlled environment. Smart greenhouse cultivation is very productive and therefore needs high precision

in terms of monitoring. Humidity, temperature and ventilation are the most important factors that must be monitored to obtain high crop quality. Continuous monitoring of these variables provides valuable and informative data to greenhouse farm managers for better understanding. Internet of Things technology creates new patterns by reducing labor costs and providing direct communication between farmers and greenhouses.

- **Controlling:** The main goal of using IoT in the greenhouse area is to increase and improve the product and develop effective and simple solutions to control climatic factors to achieve optimal plant growth. A greenhouse is an uncontrolled agricultural method of growing plants and crops. To obtain high productivity of plants and crops, it is necessary to constantly control environmental variables such as air and soil humidity, soil pH, and temperature. Through sensors and actuators,
- **Tracking:** Through an application or online control panel linked to the greenhouse and the emergence of the Internet of Things with big data, artificial intelligence, satellite images, drones, and the Global Positioning System (GPS), farmers can track and control the direction of production movement and the possibility of alarms and unwanted movement on the farm.

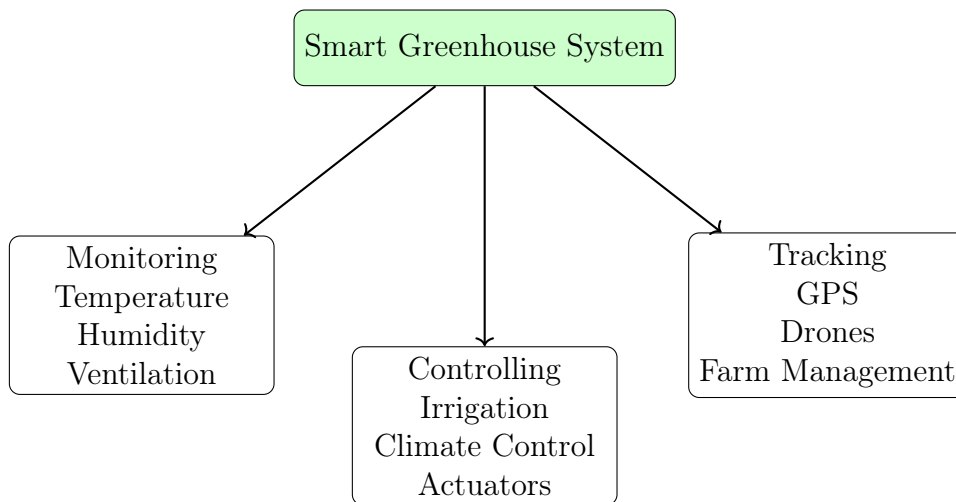


Figure 1.4: Applications of IoT in Greenhouse Agriculture

1.3 Challenges of IoT in Smart Agriculture

Many farms rely on manual monitoring and control of irrigation, despite the advantages of growing in greenhouses, which can lead to several problems:

- **Inefficient resource usage:** Using too much water and energy because of excess irrigation and no climate control.
- **Lack of real-time monitoring:** Farmers can't quickly respond to changes in temperature, humidity and soil moisture.
- **Labor-intensive management:** Manual monitoring and administration take a lot of time and effort.
- **Limited decision support:** Without real-time data farmers can't make informed decisions to boost production.

In light of these challenges, an automated system must be created capable of monitoring environmental variables and controlling irrigation and climatic conditions in real time. By incorporating IoT technologies, greenhouses can become more efficient, productive and sustainable.

1.4 Objectives and Contributions

Our project focuses on the development of a smart irrigation and environmental monitoring system. The main objectives of this work are:

- Design an interconnected system capable of collecting and monitoring environmental parameters such as soil moisture, air temperature, and humidity.
- Develop an automated control mechanism that regulates irrigation and ventilation processes according to predefined environmental conditions.
- Enable remote monitoring and supervision through an Internet-based platform, allowing users to access real-time system information.

The main contribution of this project is the development of a cost-effective and accessible smart agriculture solution that improves water and energy management while enhancing the efficiency of conventional irrigation systems.

1.5 Conclusion

In this chapter, we talked about some ideas that are important for greenhouses that grow plants and the Internet of Things (IoT). Greenhouses provide a controlled environment that allows crops to grow under ideal conditions, and combined with IoT technology enables real-time monitoring and control, improving productivity and crop quality and reducing labor costs.

The Internet of Things and greenhouses are a deal because they help us make smart farming systems.

In the next chapter, we will focus our discussion on designing and implementing a monitoring and control system based on the Internet of Things.

Chapter 2

System's Architecture

Introduction

This chapter describes the design and basic structure of the proposed intelligent greenhouse monitoring and control system. The system's main goal is monitor the climate inside the greenhouse and maintain ideal conditions for plant growth. It does this by constantly checking and adjusting key factors, like humidity, temperature, irrigation, and ventilation.

At this section, we will focus our discussion on the structure of this system and how it works, using various designs and diagrams, such as activity diagrams, state machine diagrams, and class diagrams. It provides a clear representation of how this system collects and analyzes environmental data. And he makes decisions.

2.1 Overview of the Proposed System

The proposed system consists of a smart greenhouse equipped with an Internet of Things (IoT)-based monitoring and control system designed to continuously supervise environmental parameters in real time while minimizing human intervention.

The system aims to maintain optimal growing conditions for agricultural crops by monitoring key environmental variables such as temperature, humidity, and soil moisture. Sensor data is continuously acquired and processed to evaluate the current state of the greenhouse environment.

Based on the collected data, the system automatically performs appropriate control actions, such as activating irrigation or ventilation mechanisms, to maintain environmental parameters within predefined thresholds and ensure suitable conditions for plant growth.

In addition, a web-based monitoring interface enables users to remotely access real-time environmental data and supervise system operations. The platform also provides remote control capabilities when required.

The overall objective of the project is to improve crop productivity, optimize resource utilization, and reduce the need for continuous manual monitoring and intervention.

2.2 System Lifecycle

Figure 2.1 explains the lifecycle of the proposed greenhouse monitoring and control system. The system works in a cycle, it helps to monitor the environment and control irrigation and ventilation. Each stage in the life cycle is important because it helps keep the conditions for plant growth, inside the greenhouse.

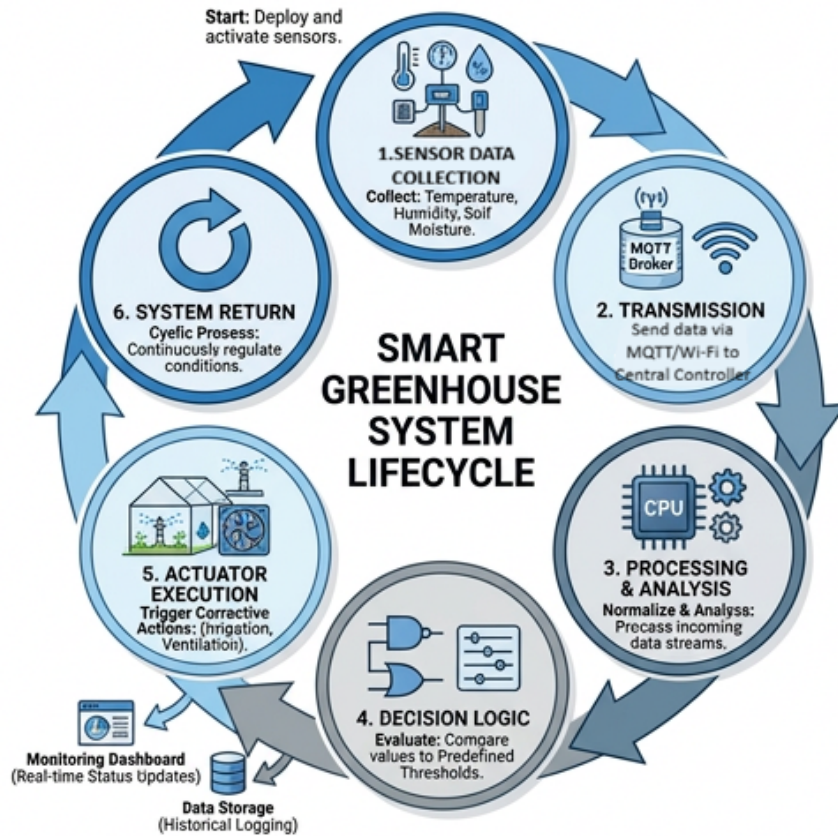


Figure 2.1: Smart greenhouse system lifecycle

The lifecycle of the system consists of several main stages:

- **Sensor Data Collection:** In the stage sensors inside the greenhouse measure things like temperature, humidity and soil moisture. These sensors keep collecting real-time data on what's happening that affects plant growth.
- **Transmission:** The data from sensors is sent to the controller using technologies like Wi-Fi or Internet of Things (IoT) protocols such as MQTT. This helps get the data to where it needs to go for further processing.
- **Processing & Analysis:** When the controller receives the sensor data, it processes and analyzes it. The measurements are then compared with predefined envi-

ronmental thresholds to determine whether the greenhouse conditions are suitable for optimal plant growth.

- **Decision Logic:** Based on the analysis the system decides what to do, like If the soil moisture is too low it decides to water the plants.
- **Actuator Execution:** The system then takes action by turning on actuators such as water pumps or fans. These actions help fix the environment in the greenhouse and make it right for plant growth.
- **System Return:** After that the system goes back to monitoring. The sensors keep collecting data and the whole process starts over. This keeps the greenhouse environment steady, over time.

2.3 system data flow

The image-2.2 shows the systems data flow. This is where the process starts. The system collects data from the environment. It does this through sensors placed inside the greenhouse. These sensors measure the condition of the greenhouse continually. They check things like temperature and humidity.

The data is sent through the communication layer. This is done using technologies like MQTT or WiFi. This allows the system to transfer information reliably between the components of the system.

The data goes to the microprocessor for analysis. Then we have information. This information is compared to the pre-determined limits. These limits represent the conditions for the greenhouse.

The system determines the actions to be taken based on this. For example the system activates the irrigation system when the soil moisture is insufficient. It operates the ventilation system when the temperature exceeds limits.

It also stores the processed data in the sensor database. This is, for analysis and monitoring purposes.

The system sends status updates and environmental information to the monitoring dashboard. This allows the user to view the status of the greenhouse in time. The user can also receive notifications. This way the user can always know what is happening with the greenhouse

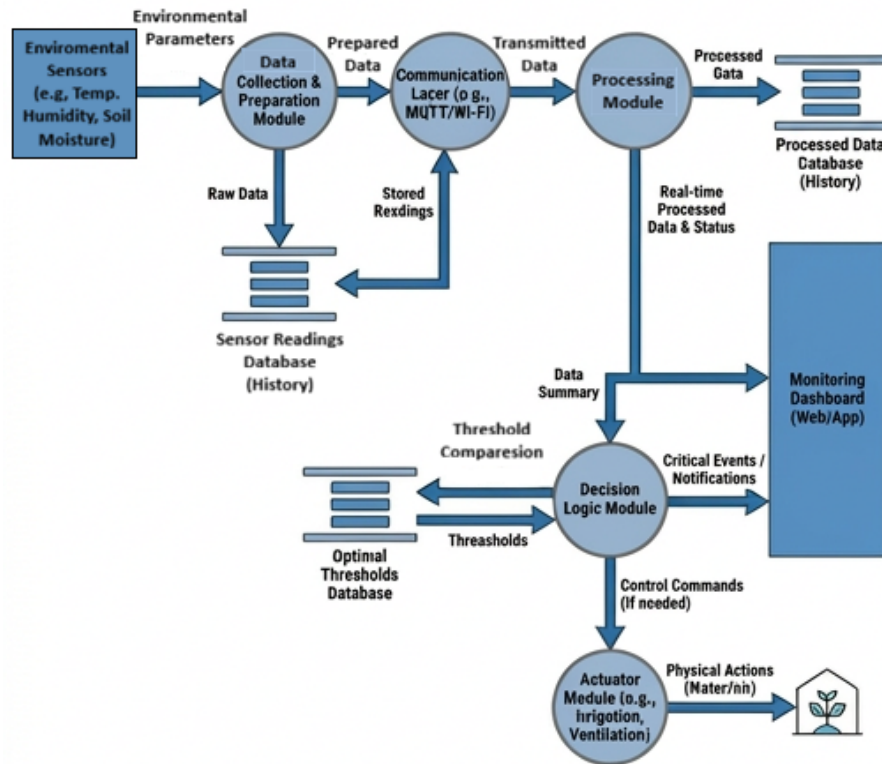


Figure 2.2: Data flow architecture of the system

2.4 UML Modeling

The Unified Modelling Language (UML) is defined as a collection of graphical models which express different properties of an object-oriented design. The two most important types of model are the structural and behavioural models [7]. In greenhouses these diagrams are used to express the behavior, interactions and structural organization of the system.

These diagrams help in understanding the operation of the system and how it interacts with its various elements. In this work, several UML diagrams are used to model the proposed system, including use case diagram, activity diagram, state machine diagram, and class diagram.

2.4.1 Use Case Diagram

This picture 2.3 shows how the farmer works with the greenhouse system. The farmer can check what the sensors are saying turn the water and air on and off and get warnings when something is wrong.

The sensors tell the system what the temperature is, how humid it is and how wet the soil is. Then the system shows this information. Does something with it.

The picture also shows that the system can do things on its own like turn on the water or air when the farmer tells it to or when it sees that something needs to be done in the greenhouse. The smart greenhouse. The farmer work together to make sure everything is okay, in the greenhouse.

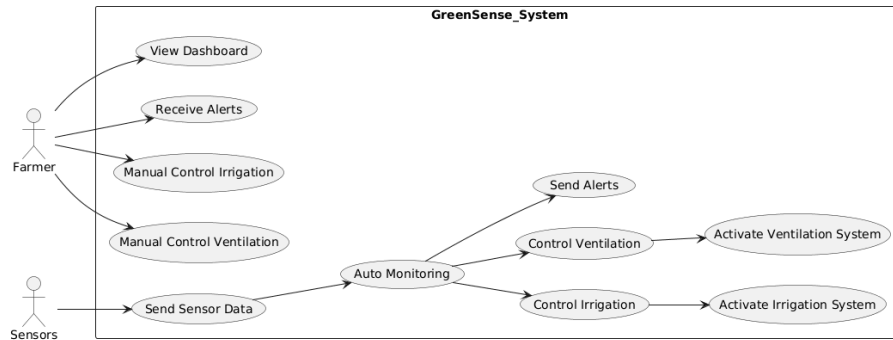


Figure 2.3: Use Case Diagram of the system

2.4.2 Class Diagram

This diagram shows the structure of the system across Classes and the relationships between them. The system collects data and sends it to the controller and the dashboard

via sensors. The role of the controller is processes and analyzes the data, then sends commands to various systems such as the irrigation system and ventilation system. Which in turn controls the motors to execute commands. The farmer can also monitor data from dashboard and receive alerts via the alert system.

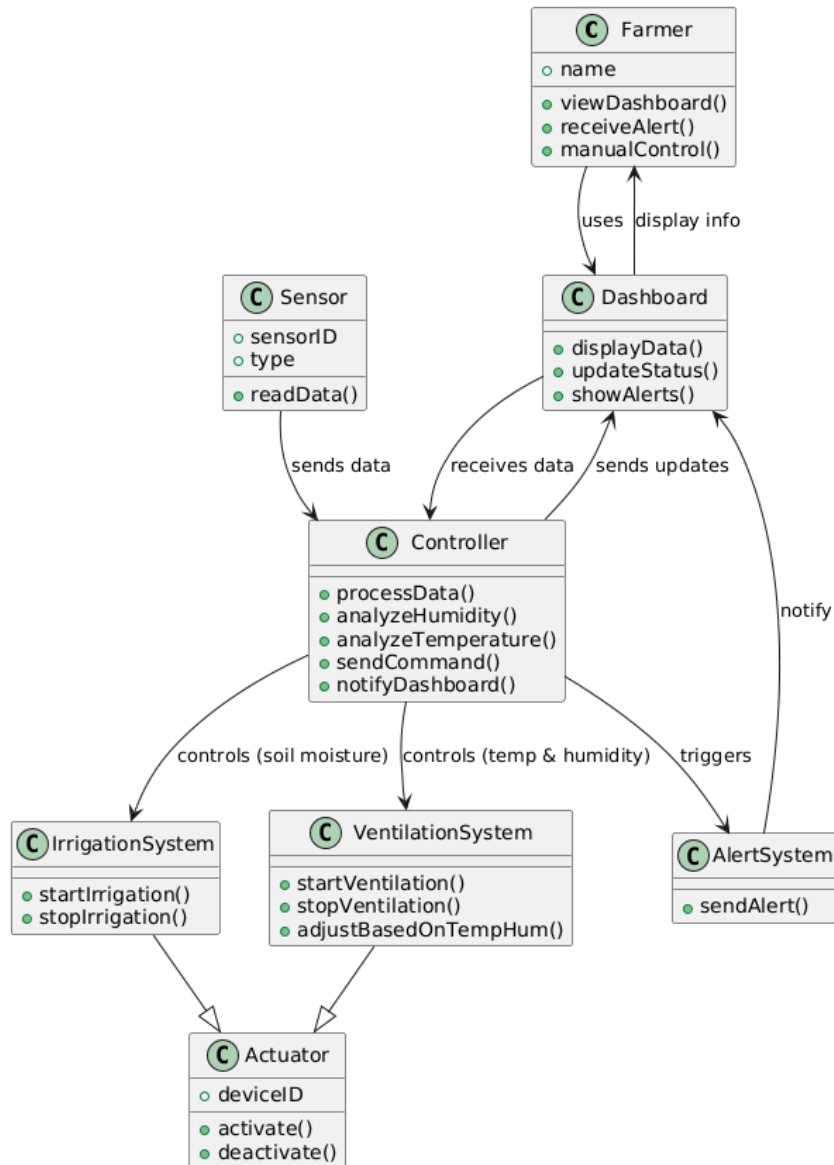


Figure 2.4: Class Diagram of the system

2.4.3 Sequence Diagram

This graph shows what happens in the system over time. Where first the sensors send information such as temperature and humidity to the controller and then the controller looks at the results of this information. If the soil is very dry, the irrigation system is turned on. If it is hot, the ventilation system will turn on, This is when we are in the automatic state, there is manual state that enable the farmer to control the ventilation

and irrigation. The system also updates the information every time and this is shown on the monitoring panel "dashboard". If a problem occurs in the system, the system sends a message to the farmer informing him of the problem.

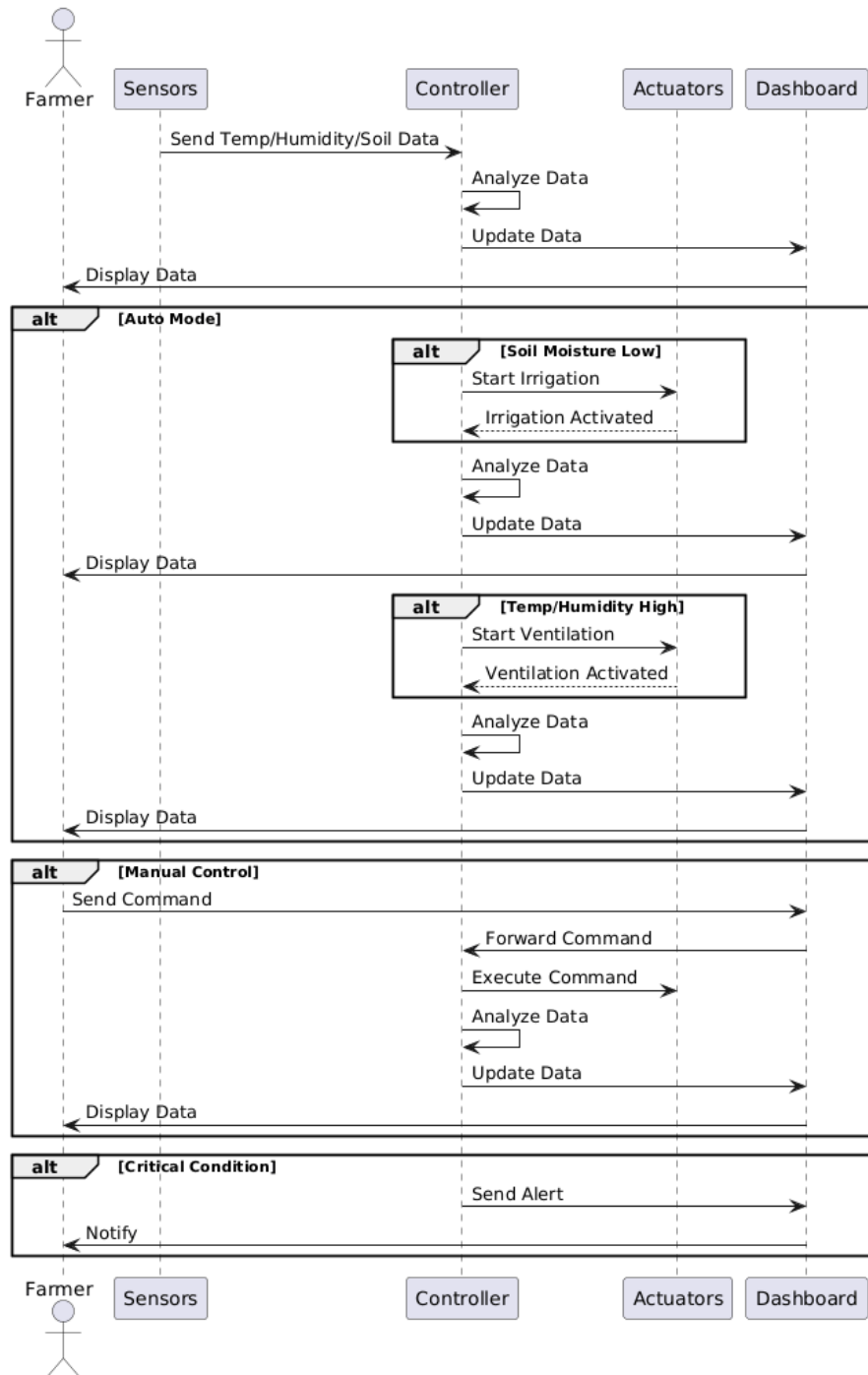


Figure 2.5: Sequence Diagram of the system

2.4.4 State Machine Diagram

The diagram shows the state of the system, as the system operates automatically.

The system starts in idle mode. Then it goes into monitoring mode. In monitoring

mode, it reads data from sensors (temperature, humidity, soil moisture). Based on the data, decisions are made. If the soil is dry, the system goes into irrigation mode. If the humidity or temperature is high, the system switches to ventilation mode. If there is a problem with the system, a message is sent to the farmer. Then, when the problem is fixed and things return to normal, the system returns to monitoring mode and continues to operate periodically.

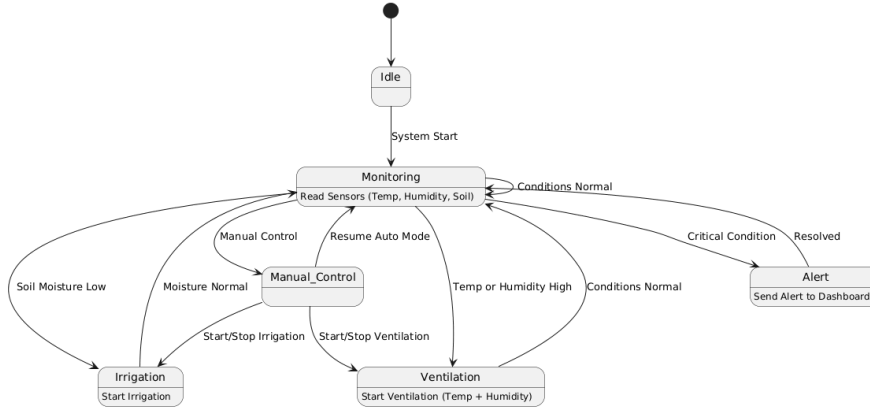


Figure 2.6: State machine diagram of the system

Conclusion

In this chapter we have provided all the details of designing our own smart greenhouse system. We first introduced the propose of our system and how it work, the general architecture of the system and then explained the details of each part.

In the next chapter, we will learn how this system and all its components are structured, how to communicate between them, and the logic applied to deal with it

Chapter 3

Methodology

Introduction

In most traditional greenhouses, the wired connection is complex and tangled, so the use of wireless communication technology has become more urgent, and in this context it is better to enable this technology, and obtain the greatest amount of profitability within agricultural greenhouses, we must have good control over climatic factors such as humidity, temperature and ventilation, and this control will ensure year-round production.

In this chapter, we will learn about the microcontrollers, sensors, and cloud platforms, whether hard or soft, that we chose in our project to help us manage the climate system inside the greenhouse efficiently.

3.1 System Circuit Design

3.1.1 Circuit Diagram Description

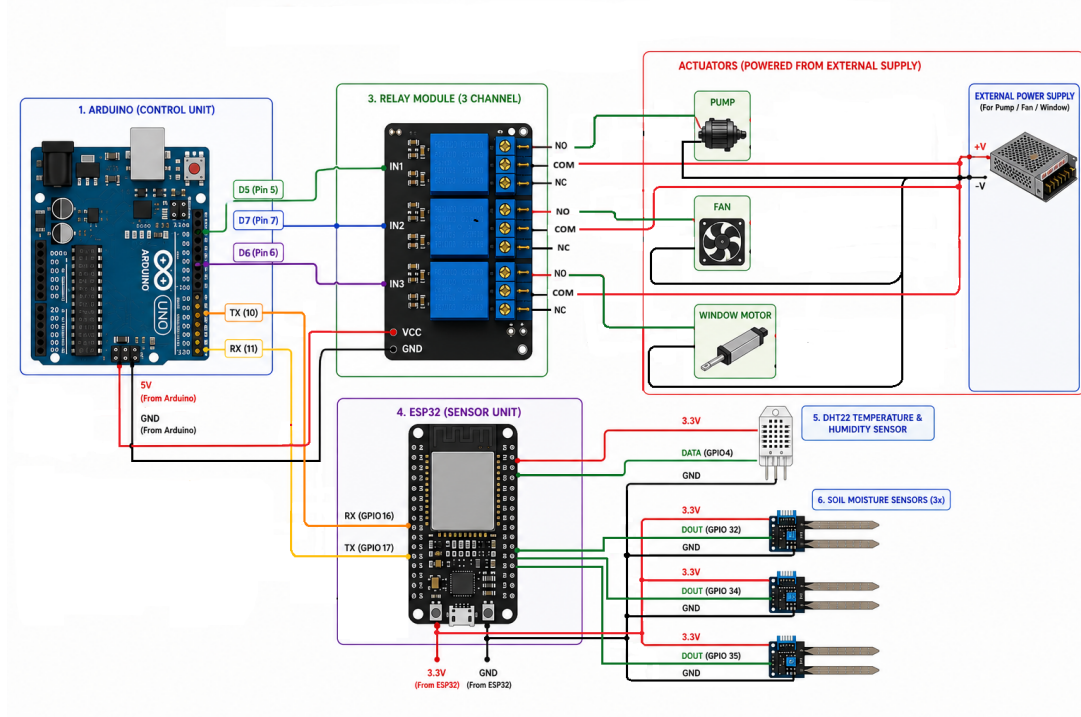


Figure 3.1: System Circuit Design

3.1.2 Component Connections

- Microcontrollers Communication (Arduino To ESP32)

Arduino	ESP32
TX (10)	RX (GPIO16)
RX (11)	TX (GPIO17)
GND	GND (Common Ground)

Table 3.1: Serial communication between Arduino and ESP32

- Relay Module Connections (Control Side)

Relay Pin	Arduino Pin	Function
IN1	Pin 5	Pump Control
IN2	Pin 7	Fan Control
IN3	Pin 6	Window Control
VCC	5V	Power Supply
GND	GND	Ground

Table 3.2: Relay module connections with Arduino

- Pump

Relay Terminal	Connection
COM	External Power (+)
NO	Pump (+)
Pump (-)	External Power (-)

Table 3.3: Pump wiring through relay

- Fan

Relay Terminal	Connection
COM	External Power (+)
NO	Fan (+)
Fan (-)	External Power (-)

Table 3.4: Fan wiring through relay

- Window Motor

Relay Terminal	Connection
COM	External Power (+)
NO	Window Motor (+)
Window Motor (-)	External Power (-)

Table 3.5: Window motor wiring through relay

- DHT22 Sensor Connections

DHT22 Pin	ESP32 Connection
VCC	3.3V
GND	GND
DATA	GPIO 4

Table 3.6: DHT22 sensor connections with ESP32

- Soil Moisture Sensors

Sensor	Pin	ESP32 Connection
Sensor 1	OUT	GPIO 32
Sensor 2	OUT	GPIO 34
Sensor 3	OUT	GPIO 35
All Sensors	VCC	3.3V
All Sensors	GND	GND

Table 3.7: Soil moisture sensors connections with ESP32

3.2 System Simulation (Tinkercad)

Tinkercad is an online platform that contains a group of electronics and development boards that allow the user to create many different circuits and experiment or realistic simulations. The most important advantage of using this platform is that it contains almost all the desired devices. The most important thing is simulation before experimenting, an important part of this work to avoid damaging the devices in reality.

3.3 Hardware Components

3.3.1 Microcontroller (ESP32)

ESP32 is a powerful and cost-effective platform for developing IoT applications. The ESP32, developed by the Espressif Systems Company (Shanghai, China), offers a powerful combination of features and capabilities for IoT applications. ESP32 has the following features: (1) a dual-core processor, (2) integrated Wi-Fi and Bluetooth connectivity, (3) a large number of general purpose input/output (GPIO) pins, and (4) low power consumption. ESP32 is equipped with a dual-core Tensilica (Santa Clara, CA, USA) Xtensa LX6 microprocessor, which provides higher processing power and facilitates multitasking and efficient execution of complex tasks. ESP32 has built-in Wi-Fi and Bluetooth interfaces that simplify connection and communication with other devices or networks. It supports various Wi-Fi protocols, such as 802.11 b/g/n, and it provides Bluetooth Classic and Bluetooth Low Energy (BLE) connectivity options. The ESP32 provides many GPIO pins that facilitate connection with and control of external devices and sensors. These pins support a variety of interfaces, including SPI, I2C, UART, and PWM. The ESP32 is designed to be power efficient, thus enabling the development of energy-efficient IoT applications. It offers sleep modes and power management features that help reduce power consumption, making it suitable for battery-powered or energy-constrained projects. ESP32 can be connected to displays, touchscreens, or LED indicators to provide operators or staff with a user-friendly interface [8].

The ESP32 can be programmed using various development frameworks and languages. The most commonly used programming language is C++, and it can be programmed using the Arduino IDE or PlatformIO. In addition, the ESP-IDF (Espressif IoT Development Framework) provides a comprehensive set of libraries and tools specifically for

ESP32 development [8].

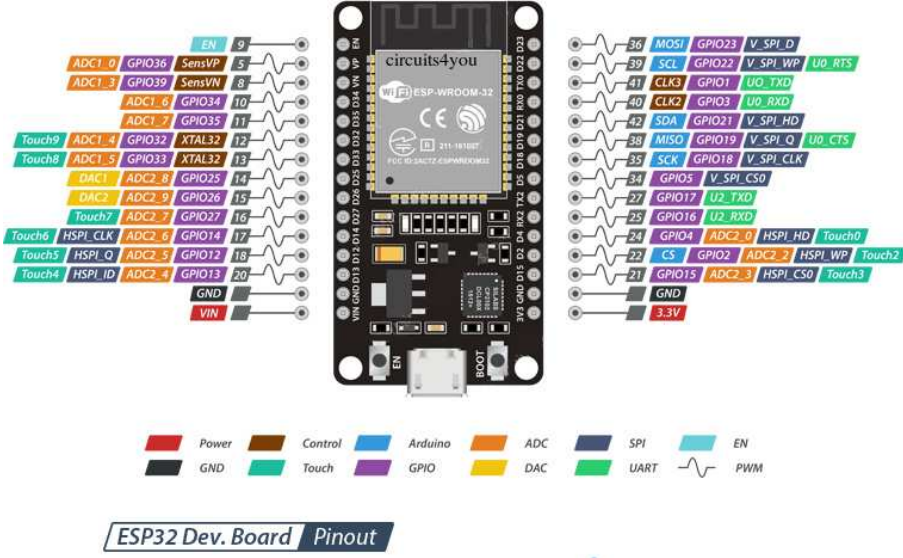
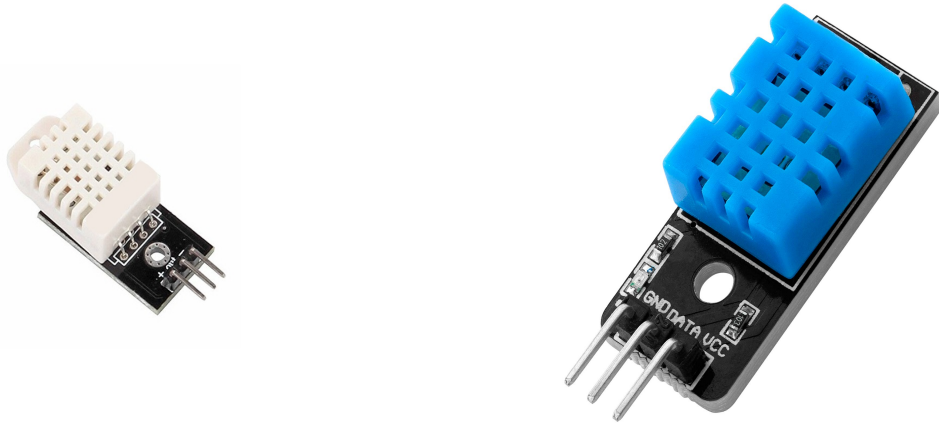


Figure 3.2: The development board ESP32-DevKitC

3.3.2 Environmental Sensors

3.3.2.1 Temperature and Humidity Sensor

The DHT22 Sensor is a digital-signal-acquisition temperature&humidity sensor. This sensor is a sensor module that functions to read temperature and humidity values that have an analog voltage output that can be processed further using a microcontroller. The sensor has anti-interference ability, low power consumption, high integration, high precision, small size and low price. The DHT22 sensor will be used in conjunction with the Arduino Uno. Calibration coefficients are stored in the OTP program memory, so that when the internal sensor detects temperature and humidity, this module includes these coefficients in the calculation [9].



(a) DHT22 module Humidity&Temperature Sensor

(b) DHT11 module Humidity&Temperature Sensor

Figure 3.3: Humidity and Temperature Sensor

1. Pin 1 is connected to VCC
2. Pin 2 is connected to DATA
3. Pin 3 is connected to NC
4. Pin 4 is connected to GND

3.3.2.2 Soil Moisture Sensor

Presents a moisture sensor that uses Time Domain Reflectometry (TDR) technology. TDR technology as known as volumetric assesses genuine soil water content rather than soil water potential. The steel poles embedded in the soil accumulates electrical signs from the TDR device while the sensors gage the signal's rate of return which approximates the amount of water in the soil. In comparison, water-deprived soil returns signals faster than wet soil. Conversely, TDR sensors insist on data interpretation and may require different calibrations with dependency on the different variety of soil makeup. The function of the soil moisture sensor is to measure the water content in the soil and with it, the user can distinguish with ease when soil requires more water or when water is over-compensated. Moisture levels of the soil are detected using this instrument and provides the outcome to Arduino (Analog Input). Subsequently, when the moisture of the soil has been reduced to a certain level of below, Arduino is programmed to trigger the water system to either dispel or with hold water [10].

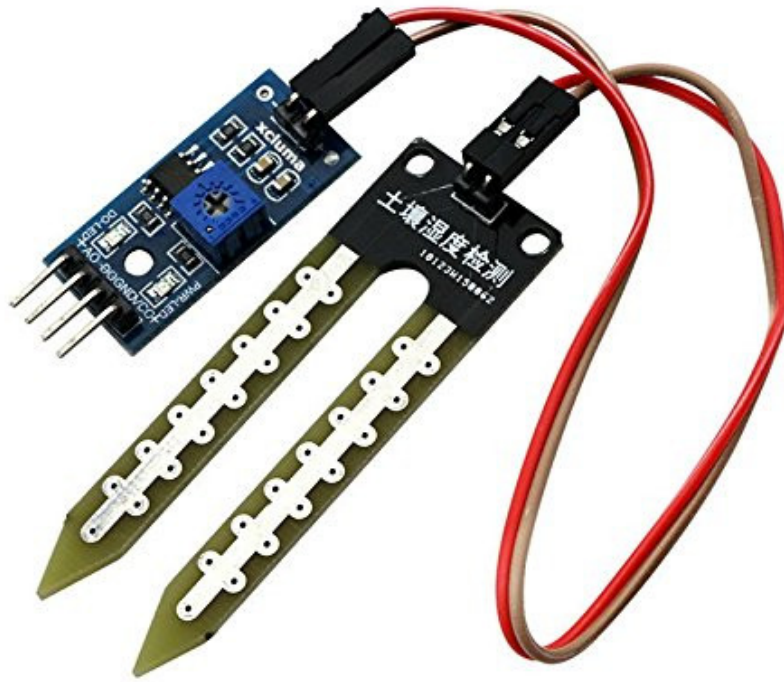


Figure 3.4: Soil Moisture Sensor

3.3.3 Actuators

3.3.3.1 Irrigation Pump

An irrigation pump (Figure 3.5) is a device used in agricultural irrigation to pump water from a water source and distribute it through irrigation pipes to the agricultural areas. The purpose of irrigation pumps is to provide an adequate amount of water and suitable pressure for watering plants and meeting their water requirements.



Figure 3.5: Irrigation Pump

3.3.3.2 Electric valve

Electric valves are considered one of the important components in the irrigation system. We used an electric valve connected to the water pump to control the irrigation process automatically. This component works via an electrical signal coming from the microcontroller. When an operating signal is sent, a valve is opened and water is allowed to pass, and when the signal is stopped, the valve is closed and water is prevented from flowing.

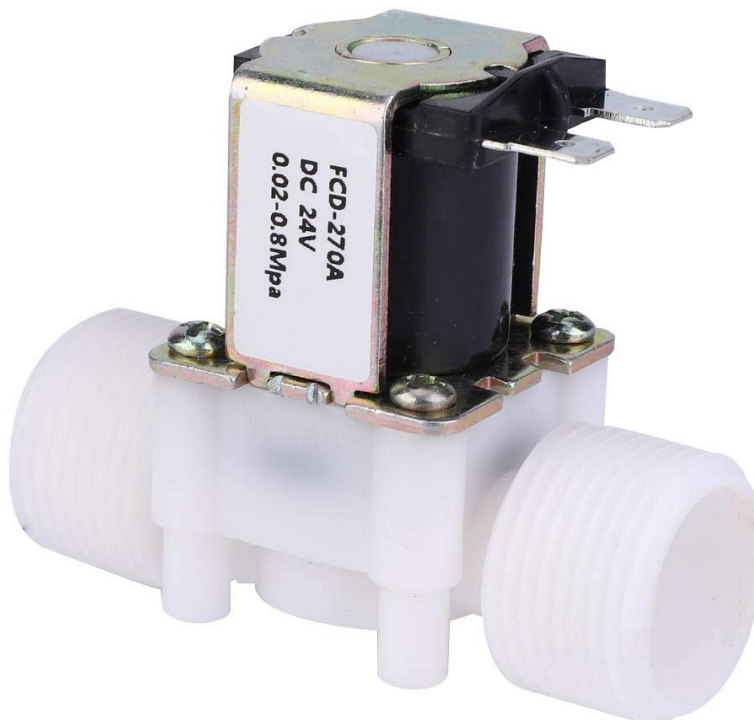


Figure 3.6: Electric valve

3.3.3.3 Ventilation Fan

Air removal equipment in greenhouses includes a set of tools used to ventilate and remove excess air from inside the greenhouse. Among these equipments, the ventilation fan (Figure 3.7) comes to the fore, as it moves air and removes excess humidity and heat from inside the greenhouse. A ventilation fan helps provide fresh airflow and reduce humidity build-up. The use of a proper ventilation fan contributes to achieving suitable conditions for plant growth and improving air quality inside the agricultural house.



Figure 3.7: Ventilation Fan

3.3.3.4 Window Motor

There is an essential part in the greenhouse related to ventilation, which are the windows. We used this motor (Figure 3.8) to open and close the window, as it helps in introducing ventilation and removing the high humidity.



Figure 3.8: Window Motor

3.4 Communication and Data Transmission

3.4.1 Wi-Fi Connectivity

Many electronic devices like smart phones, laptops, and more utilize a low power wireless communication also known as WiFi. In this setting, router capacities as a correspondence hub point wirelessly with the restriction of allowing clients to connect within close to the router within the network. WiFi is exceptionally normal in systems administration applications which manages transportability wirelessly. The maximum range of the WiFi is 100 meters and network type is a WLAN for lengthy distances. The ESP8266 WiFi module is trendy for its IoT applications. The AI-thinker Team developed ESP-12E WiFi module as known as WiFi Shield. The core processor ESP8266 in little sizes of the module incorporates Tensilica L106 integrates industry-leading ultra-low power 32-bit MCU micro, Clock speed bolster 80 MHz, 160 MHz, integrated WiFi, on-board antenna. Created for space and power limited mobile platform designers, the ESP8266 is a high integration wireless SoCs that delivers unparalleled potential to implant WiFi capabilities within our system or to operate as a standalone application with minor expenses and negligible space requirement [10].

3.4.2 MQTT Communication Protocol

Message Queue Telemetry Transport (MQTT) is a message-based communication protocol introduced by IBM in 1999. It is the most common protocol used to exchange

data between machines within the Internet of Things platform. This protocol is preferred because the size of the communication packets is small, easy to use and implement, and the network bandwidth is large [1].

Protocols	CoAP	HTTP	MQTT	WebSocket
Transport	UDP	TCP	TCP	TCP
Messaging	Request/Response	Request/Response	Publish/Subscribe	Asynchronous
Community Support	IPSO, OMA, IETF	IETF, oneM2M	IBM	IETF
Delay	Fair	Fair	Good	Good
Cloud Support	Good	Good	Good	Fair
Power Usage	Good	Good	Fair	Good

Table 3.8: Properties Comparison of IoT Communication Protocols [1]

MQTT is a protocol based on the publish / subscribe system. This system consists of Client, Broker and Topic. The client represents the devices which are connected to the network and sending and receiving data such as sensors, cell phones, and computers. An MQTT client can be both a publisher or a subscriber. The publisher is the source of the data on the system to be sent. A subscriber is a client who wants to receive messages from the publisher. A MQTT broker is a device that connects clients in the system. The broker is responsible for collecting and maintaining that data when the client connected to it publishes data and processing it properly. It then transmits this data to the subscribers connected to it. The broadcaster and subscriber do not need to work and know each other at the same time. The broker retrieves the data into the buffer in sequence when the client is offline and forwards it when the client is online. It has a queuing system in this way [1].

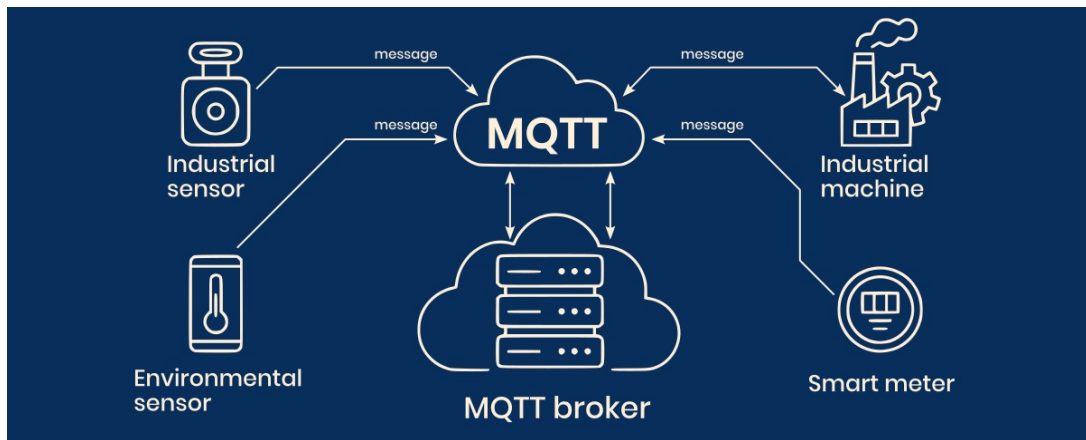


Figure 3.9: MQTT Protocol Movement

3.5 Software Architecture

3.5.1 Embedded System Programming

Embedded system programming is the process of developing and writing programs for embedded systems, which are electronic systems designed to implement a specific function within a specific device. This programming relies on direct control of hardware such as sensors, motors, and microcontrollers, with the aim of performing specific tasks with high efficiency and in real time. We rely on the controller to read the data from the sensors, process it, and then make a decision.

3.5.2 Control Logic

3.5.2.1 Controlling the temperature according to the DHT22 temperature sensing:

DHT22 reads temperature from the air and sends the data to the controller, which in turn sends it to the controller, then compares the measured temperature with the maximum limit, If the temperature is higher than the maximum, open the fan and window, else If the temperature is less than the maximum, close the fan and window.

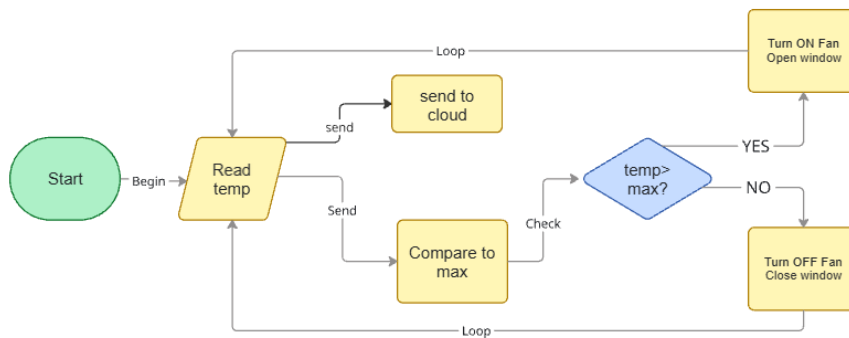


Figure 3.10: DHT22 Sensor and Temp Control

3.5.2.2 Controlling the Humidity according to the dht11 temperature sensing:

DHT22 reads humidity from the air and sends the data to the controller, which in turn sends it to the control panel. The control unit compares the measured temperature with the maximum limit. If the humidity is higher than the maximum, he opens the fan and window, else If the humidity is less than the maximum, he closes the fan and window.

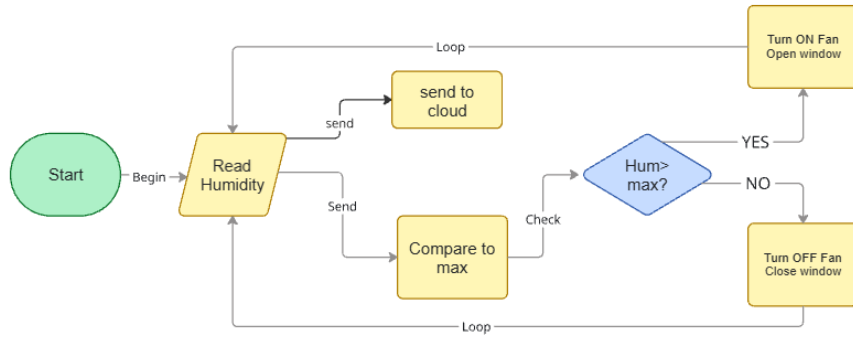


Figure 3.11: DHT22 Sensor and Humidity Control

3.5.2.3 Soil moisture measurement and Irrigation Pump control diagram

The soil moisture sensor reads moisture from the air and sends the data to the controller, which in turn sends it to the controller then compares the measured humidity with the maximum, If the humidity is higher than the maximum, the pump shuts down , else If the humidity is less than the maximum, he opens the pump.

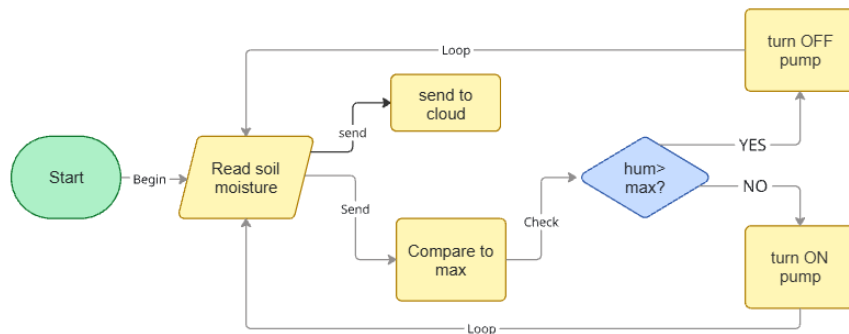


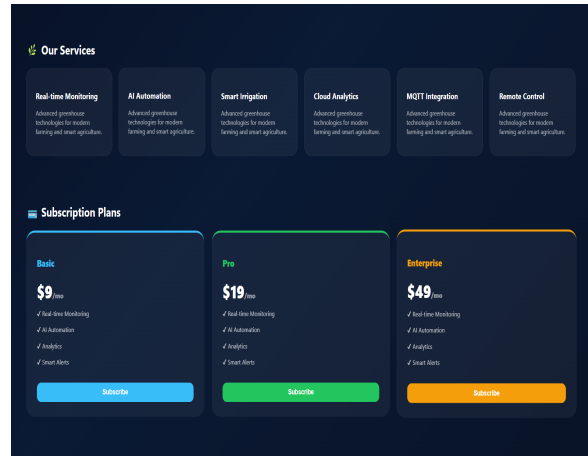
Figure 3.12: Soil Moisture measurement and Irrigation Pump

3.5.3 Monitoring Interface / Dashboard

Figures (3.13, 3.16, 3.17, 3.18) show our system dashboard that display all the things it happen in real time, it display sensors readings, the current states of the actuators, and allows switching change between auto and manual mode.



(a) side 1



(b) side 2

Figure 3.13: The main page

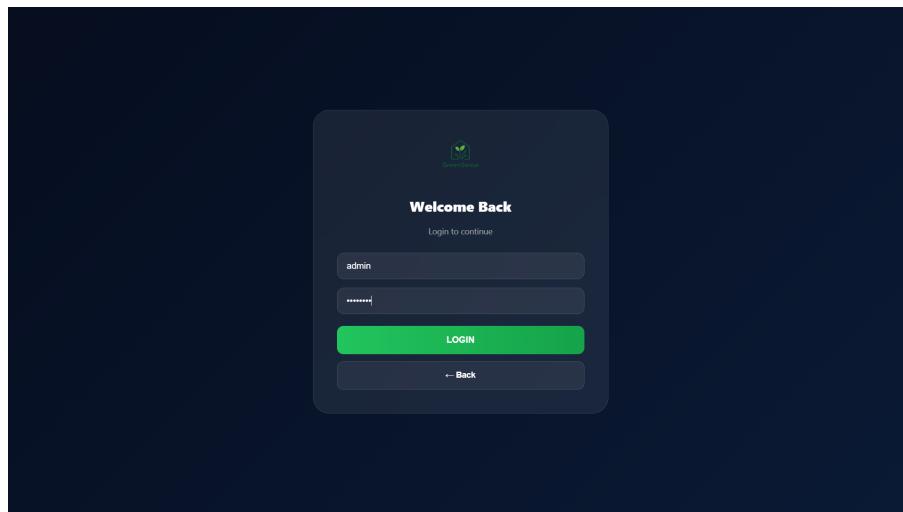


Figure 3.14: Login Page

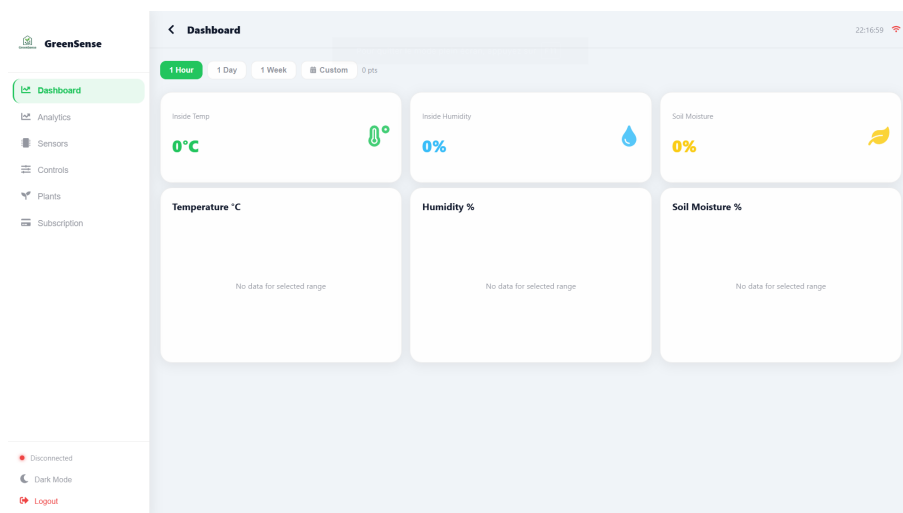


Figure 3.15: dashboard

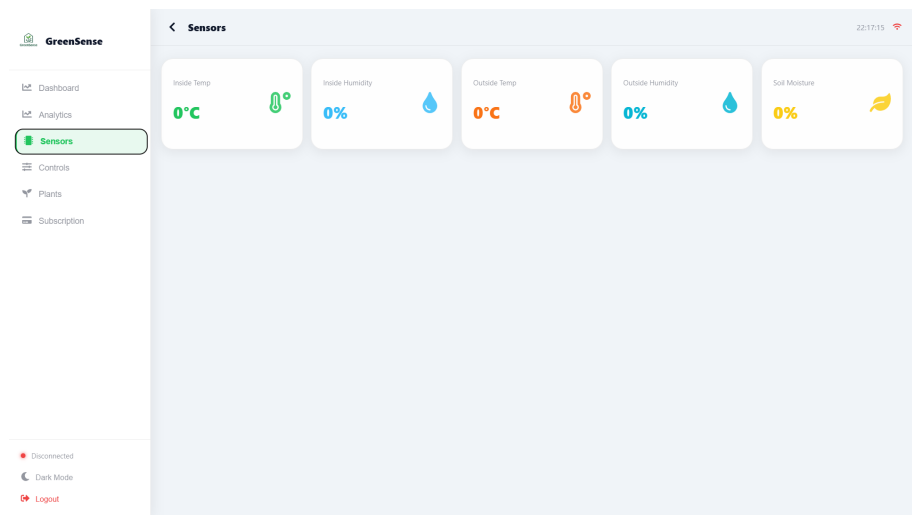


Figure 3.16: Sensor state page

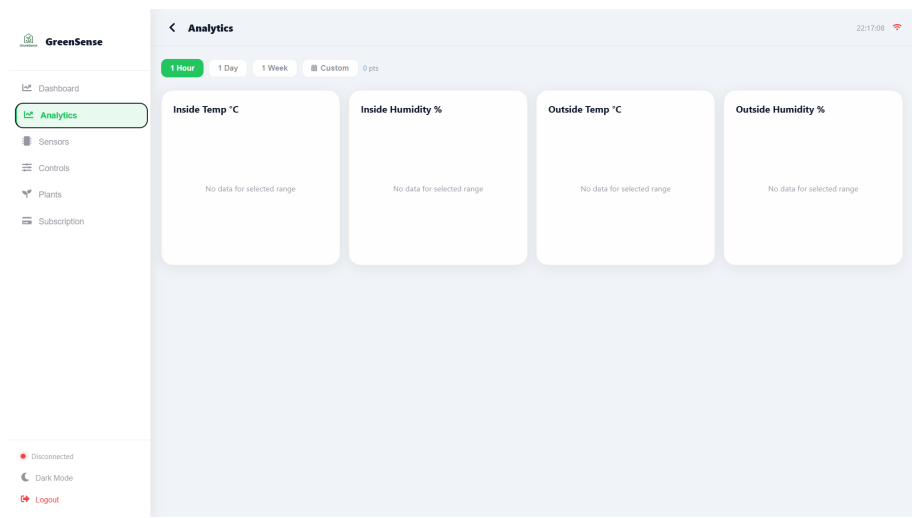


Figure 3.17: Analytics page

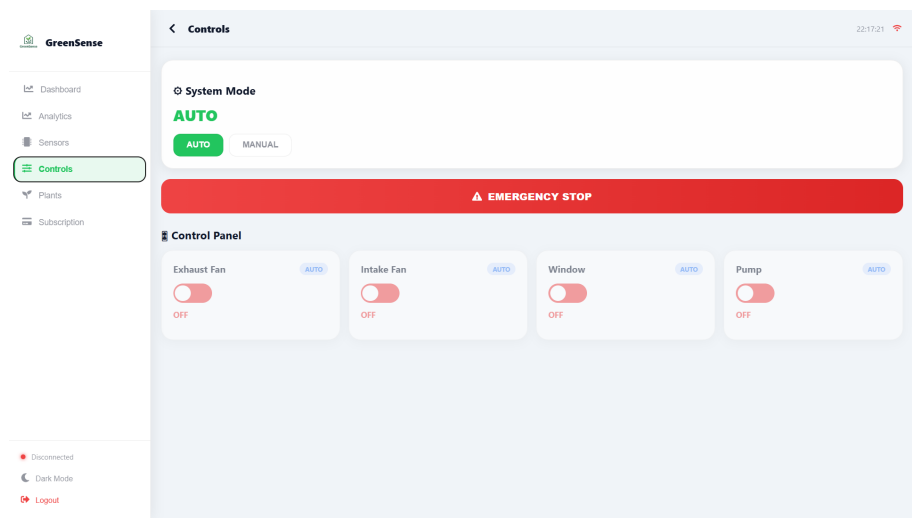


Figure 3.18: Control page

3.6 Conclusion

In this chapter, It has been clarified the greenhouse control circuit, which serves as the fundamental basis for our project. We have also outlined and elucidated the necessary accessories required for the implementation of our project. Furthermore, we have expounded upon the benefits and significance of these accessories in relation to our project.

In the next chapter, we will focus to clarified clarifying the results of our system and the dashboard.

Chapter 4

Experimental Results

Introduction

In this chapter, we will discuss the experimental tests that were conducted on the greenhouse and the implementation and presentation of the results. These experiments were conducted in order to verify the smooth operation of the system and verify its performance of monitoring environmental conditions inside the greenhouse. The operation of the sensors has been verified accurately through conducting and implementing tests. The extent of the control system's response to climate fluctuations has also been verified. The effectiveness of the automatic monitoring process has also been verified. Through these tests and experiments conducted on the system, we show that it is capable of controlling the climatic conditions inside the greenhouse.

4.1 Experimental Setup

The experimental environment for the system has been created and built in order to test it in real working conditions inside the greenhouse. This environment contains the installation of sensors, control systems and actuators in a way that allows for excellent and continuous monitoring of the climate conditions inside the greenhouse. In installing the sensors, care was taken to place each sensor in the appropriate place inside the greenhouse in order to obtain accurate and actual results for the climatic conditions used in our system. Also, all elements of the system were connected to the controller that analyzes and studies the results and makes the appropriate decision in all cases. We also make sure to keep the system running at all times in order to test it in all conditions and monitor how it responds to climatic fluctuations such as high humidity and others.

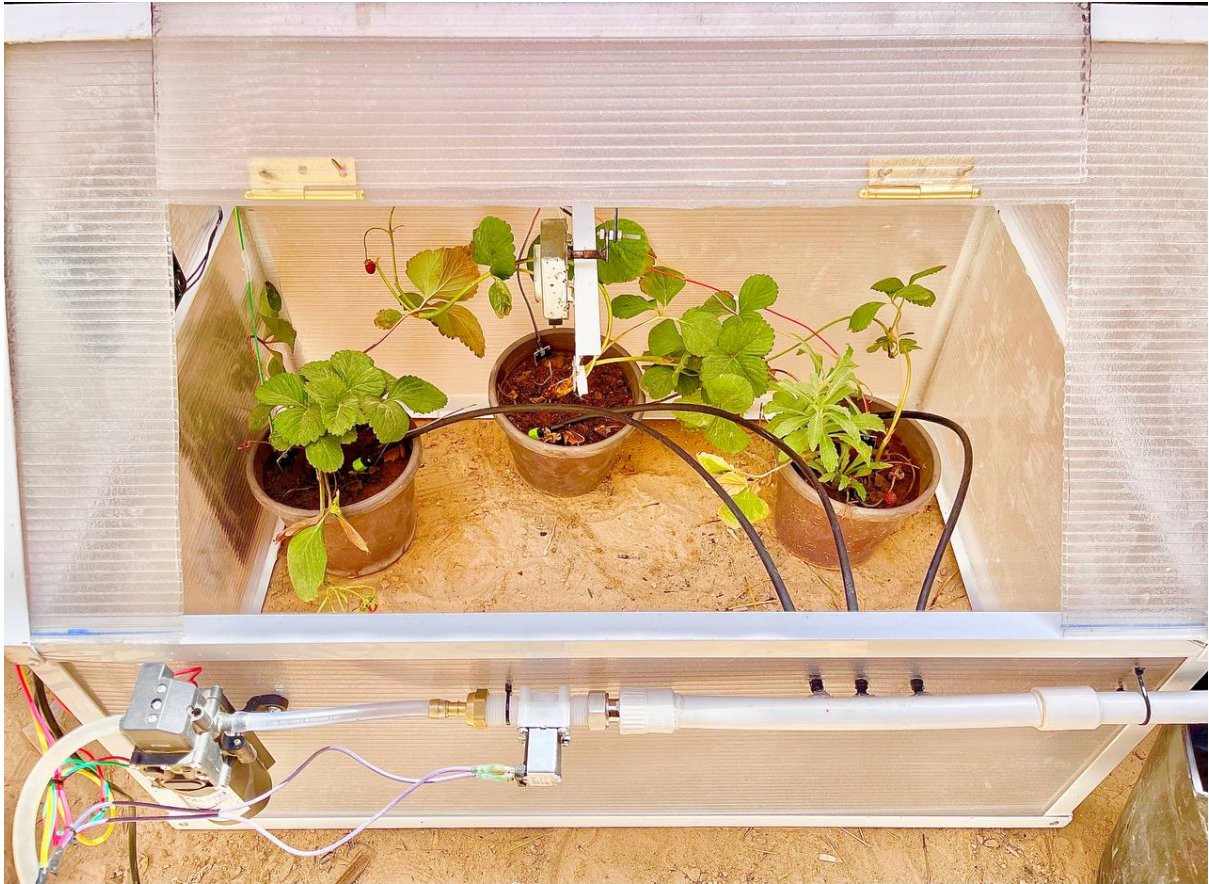


Figure 4.1: Experimental Setup of the Smart Greenhouse System

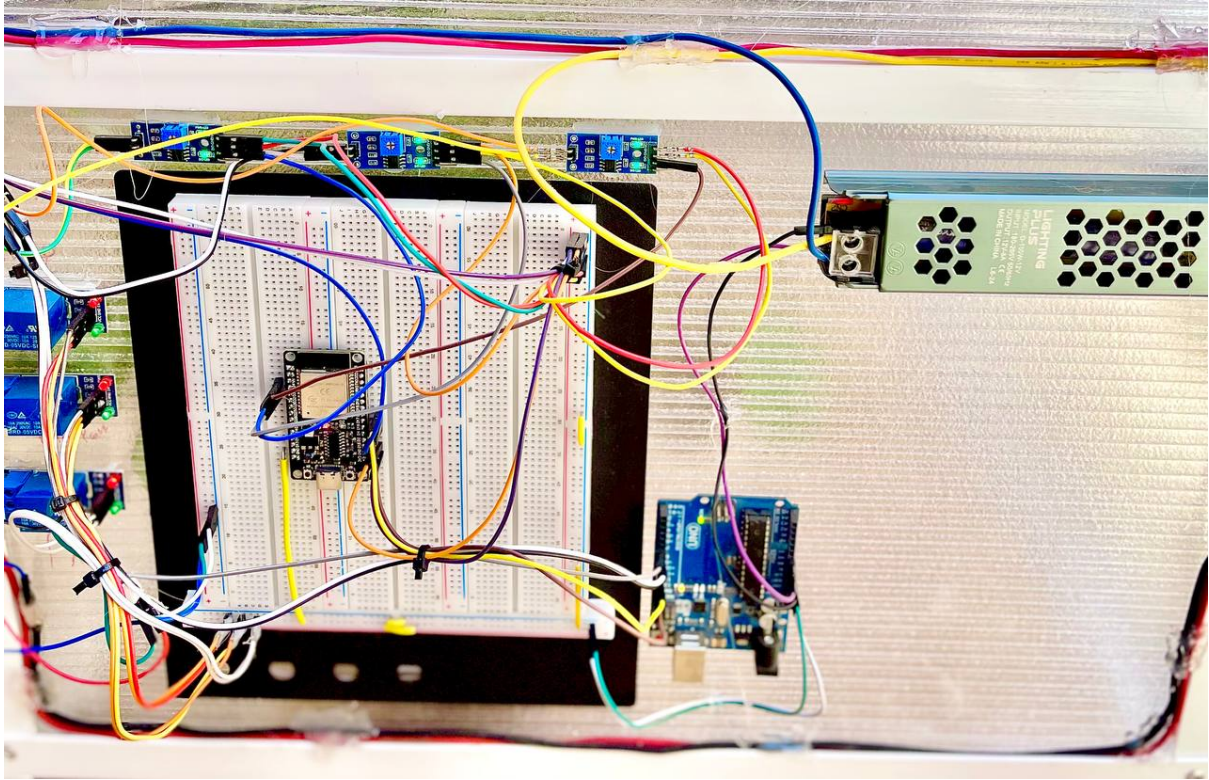
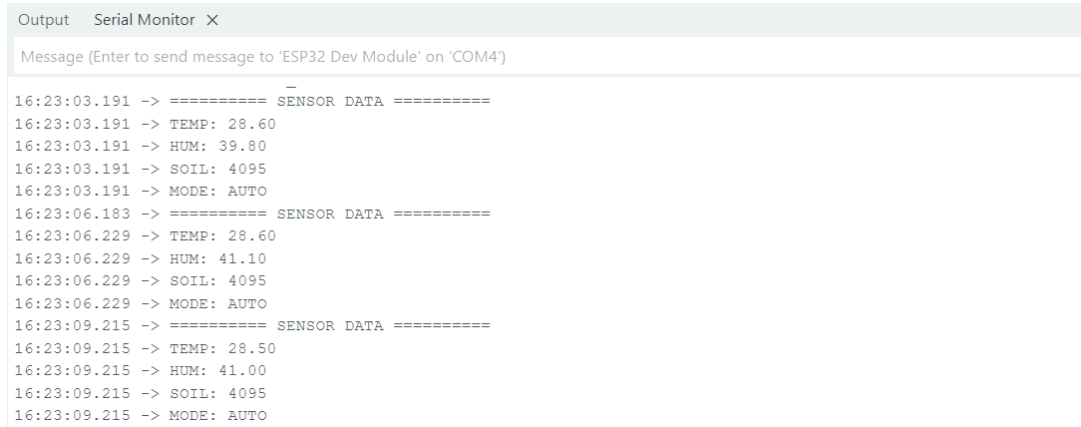


Figure 4.2: Main Controller

4.2 Sensor Data Monitoring

Here we have data sent from the sensors that we constantly monitored while the system was running. Temperature, humidity, and soil moisture are the most important factors that affect the growth of agricultural crops. The sensors send information to the controller continuously and periodically, which in turn processes the data, analyzes it, and compares it to the specified threshold. This process helped us monitor environmental conditions continuously and smoothly. The results showed us that the sensors are capable of monitoring the climatic conditions inside the greenhouse, which helped us obtain appropriate decisions from the system.



The screenshot shows a Serial Monitor window titled "Output Serial Monitor X". Below the title bar is a text input field with the placeholder "Message (Enter to send message to 'ESP32 Dev Module' on 'COM4')". The main area displays a log of sensor data. The data is organized into three groups, each starting with a separator line "===== SENSOR DATA =====". Each group contains four lines of data: "TEMP: 28.60", "HUM: 39.80", "SOIL: 4095", and "MODE: AUTO". The first group is timestamped "16:23:03.191", the second "16:23:06.183", and the third "16:23:09.215".

```

16:23:03.191 -> ===== SENSOR DATA =====
16:23:03.191 -> TEMP: 28.60
16:23:03.191 -> HUM: 39.80
16:23:03.191 -> SOIL: 4095
16:23:03.191 -> MODE: AUTO
16:23:06.183 -> ===== SENSOR DATA =====
16:23:06.229 -> TEMP: 28.60
16:23:06.229 -> HUM: 41.10
16:23:06.229 -> SOIL: 4095
16:23:06.229 -> MODE: AUTO
16:23:09.215 -> ===== SENSOR DATA =====
16:23:09.215 -> TEMP: 28.50
16:23:09.215 -> HUM: 41.00
16:23:09.215 -> SOIL: 4095
16:23:09.215 -> MODE: AUTO

```

Figure 4.3: Serial Monitor

4.3 Ventilation System Results

When the temperature and humidity exceeds the specified limit, a fan is turned on or the window is opened for ventilation automatically. The system succeeded in maintaining a stable climate.

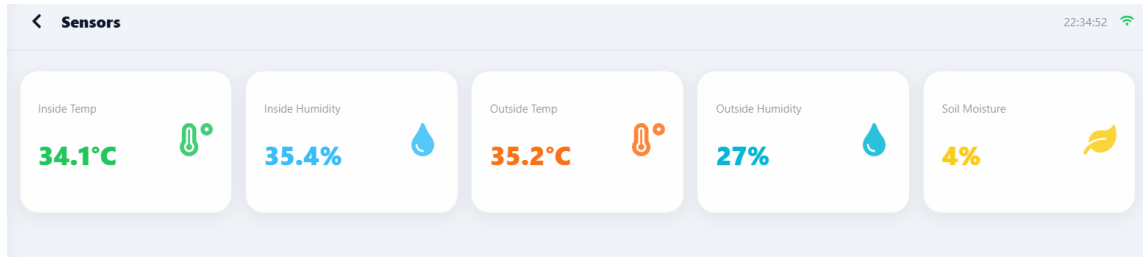


Figure 4.4: Result of temp&hum before

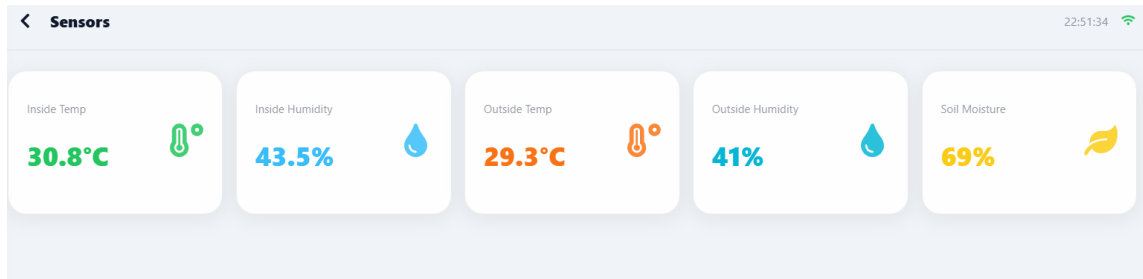


Figure 4.5: Result of temp&hum after

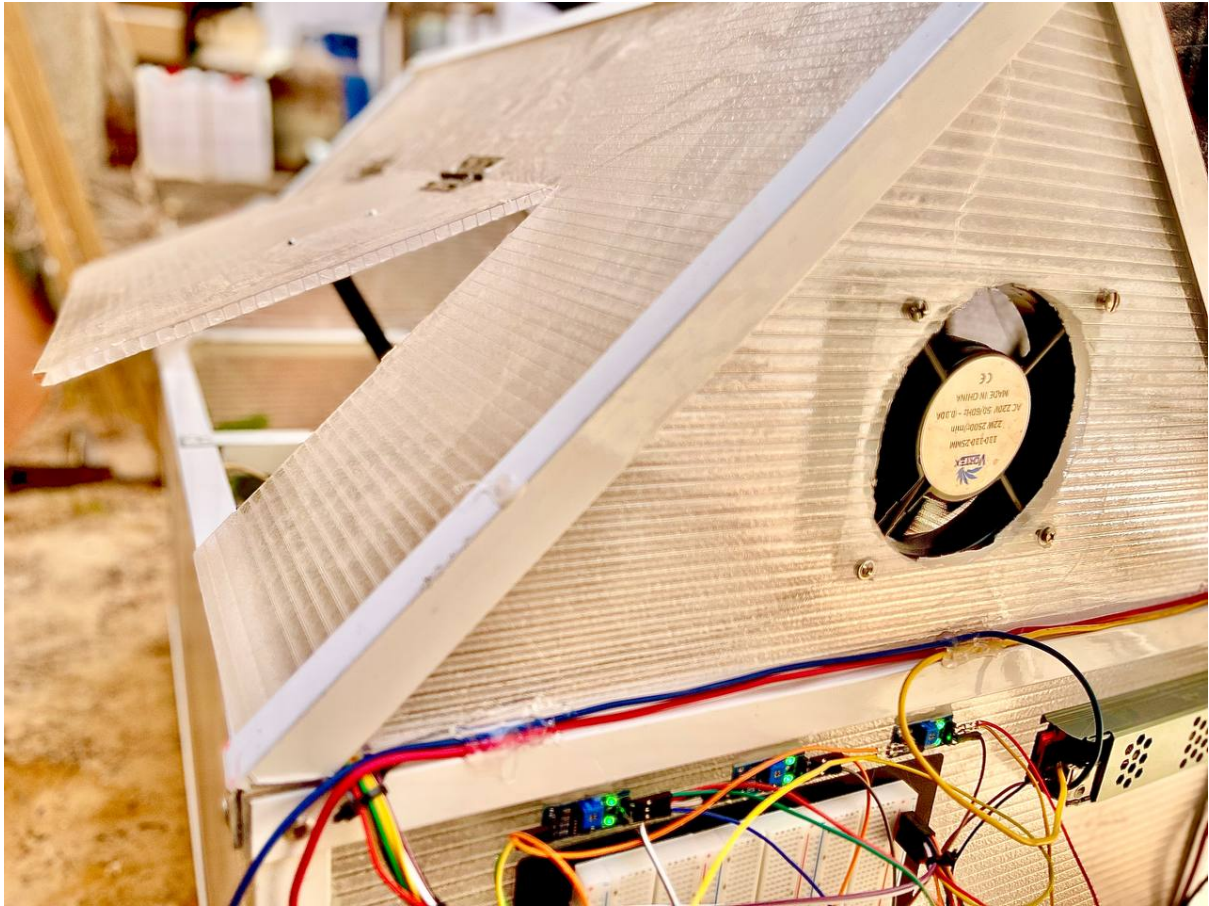
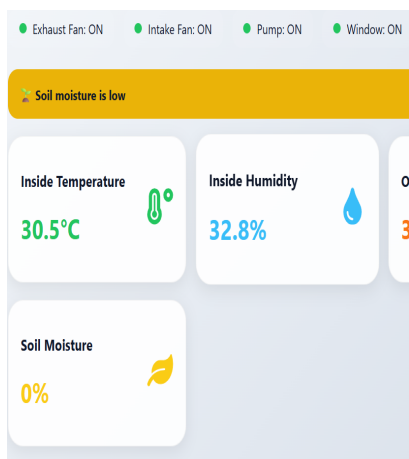


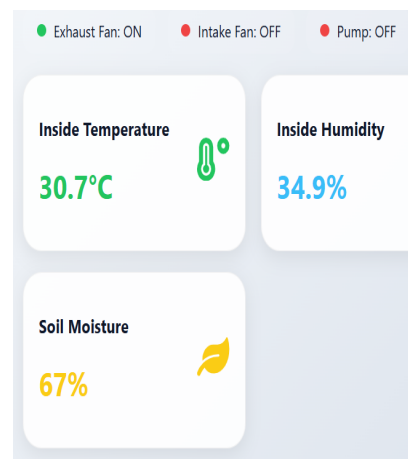
Figure 4.6: Ventilation System Using Fan and Window

4.4 Irrigation System

The irrigation system turns on automatically when the soil moisture level drops below the specified limit. The pump kept running until the humidity level returned to normal. That is, the system responded correctly.



(a) Soil before



(b) soil after

Figure 4.7: Result of Soil Moisture



Figure 4.8: Automatic Watering System

4.5 Real-Time Monitoring Results

Sensor data was transmitted successfully in real time using the MQTT protocol and Wi-Fi connectivity. The monitoring dashboard displayed temperature, humidity, and soil moisture values continuously without significant delay, and The obtained results (figure 4.9) displays sensor changes at every moment represented by charts.



Figure 4.9: Real-Time Monitoring Results

4.6 System Response Time

The system response time was tested by changing environmental conditions manually. The average response time between sensor detection and actuator activation was approximately 2–3 seconds.

Conclusion

In this chapter, we have detailed the fabrication process of the prototype greenhouse management and control system. and show all the results from the dashboard and real world for Irrigation system and ventilation system.

Conclusions

The climate inside a greenhouse is a very complex and dynamic system. Since temperature and humidity are linked, this leads to the use of highly accurate monitoring methods. For this reason, we developed a system to control several factors inside the greenhouse, such as soil moisture, air, and temperature. We used the ESP32 controller in our project, which led to very satisfactory results in real time.

Integrating the web application into our project has proven to be very useful in remote controlling the greenhouse from anywhere around the world.

Working on this project helped us to comprehensively test and enhance our theoretical and practical understanding of electronics. Additionally, it served as a catalyst for exploration and learning in diverse fields of study including agriculture (climate management, greenhouse types, equipment), and embedded electronics.

The goal of the project is to successfully implement Internet of Things technology to automate the greenhouse and facilitate remote work for the farmer, especially since the greenhouse will be controlled automatically or through a web application that investors can use to make agriculture very precise. It is worth noting that our primary focus was on climatic factors rather than plant growth.

Perspectives

In terms of future perspectives, we have outlined several areas for improvement and expansion in our project:

- Due to the large energy consumption, we suggest using solar energy to generate electricity to reduce consumption, and by harnessing clean energy, we can make the system more sustainable and environmentally friendly.
- Artificial intelligence and machine learning technologies can be integrated that help in analysis and decision-making processes that lead to improved efficiency and productivity.
- Design a suitable greenhouse structure to collect rainwater instead of wasting it and using groundwater.
- We plan to integrate smart farm systems used in developed countries, such as fertilizer irrigation systems, carbon dioxide injectors, water heating, and other advanced technologies. This integration will further improve the farming process.

References

- [1] F. Yalçinkaya, H. Aydılek, M. Y. Erten, and N. İnanç, “IoT based smart home testbed using MQTT communication protocol,” *International Journal of Engineering Research and Development*, vol. 12, no. 1, pp. 317–324, 2020. x, 34
- [2] W. Djafri, “Composition des services dans l’internet des objets,” Master’s thesis, Universite Ferhat Abbas Setif 1, 2016. 4
- [3] K. Mesmoudi, *Etude Expérimentale et Numérique de la Température et de l’Humidité de l’Air d’un Abri Serre Installé dans les Hauts Plateaux d’Algérie, Région des Aurès*. PhD thesis, Université de Batna 2, 2017. 4, 5
- [4] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of things (IoT): A literature,” *Journal of Computer and Communications*, vol. 3, pp. 164–173, 2015. 7
- [5] P. Gokhale, O. Bhat, and S. Bhat, “Introduction to IoT,” *International Advanced Research Journal in Science, Engineering and Technology*, vol. 5, no. 1, pp. 41–44, 2018. 7
- [6] M. S. Farooq, S. Riaz, M. A. Helou, F. S. Khan, A. Abid, and A. Alvi, “Internet of things in greenhouse agriculture: A survey on enabling technologies, applications, and protocols,” *IEEE Access*, vol. 10, pp. 53374–53397, 2022. 8
- [7] T. Clark and A. Evans, “Foundations of the unified modeling language,” in *Proceedings of the 2nd BCS-FACS Northern Formal Methods*, BCS Learning & Development, 1997. 19
- [8] D. Hercog, T. Lerher, M. Truntič, and O. Težak, “Design and implementation of ESP32-based IoT devices,” *Sensors*, vol. 23, no. 15, p. 6739, 2023. 27, 28
- [9] T. I., S. H. Sinaga, A. Sudiati, and Nasution, “Design of monitoring system temperature and humidity using DHT22 sensor and NRF24L01 based on arduino,” in

- Journal of Physics: Conference Series*, vol. 2421, p. 012018, IOP Publishing, 2023.
- 28
- [10] K.-M. Chew, S. C.-W. Tan, G. C.-W. Loh, N. Bundan, and S.-P. Yiiiong, “IoT soil moisture monitoring and irrigation system development,” in *Proceedings of the 2020 9th International Conference on Software and Computer Applications*, pp. 247–252, 2020. 29, 33
 - [11] T. Ahmad, D. Zhang, and C. Huang, “Internet of things and artificial intelligence in smart agriculture: Recent advances and future challenges,” *Computers and Electronics in Agriculture*, vol. 221, p. 108905, 2024.
 - [12] V. P. Kour and S. Arora, “Recent developments of the internet of things in agriculture: A survey,” *IEEE Access*, vol. 11, pp. 14875–14905, 2023.
 - [13] R. R. Shamshiri, F. Kalantari, and K. C. Ting, “Advances in greenhouse automation and controlled environment agriculture,” *Biosystems Engineering*, vol. 229, pp. 152–170, 2023.
 - [14] M. Javaid, A. Haleem, and R. P. Singh, “IoT applications for sustainable smart farming and agriculture,” *Sustainable Operations and Computers*, vol. 4, pp. 275–284, 2023.
 - [15] V. K. Sarker, J. P. Queralta, and T. Westerlund, “Smart greenhouse monitoring and control system using IoT technologies,” *Sensors*, vol. 22, no. 18, p. 6981, 2022.
 - [16] M. Ayaz, M. Ammad-Uddin, and Z. Sharif, “Internet-of-things based smart agriculture: Toward sustainable farming,” *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 12937–12956, 2022.
 - [17] M. S. Farooq, S. Riaz, and A. Abid, “A survey on the role of IoT in agriculture for the implementation of smart farming,” *IEEE Access*, vol. 9, pp. 156237–156271, 2021.
 - [18] L. Benos, A. Bechar, and D. Bochtis, “Artificial intelligence and IoT technologies for smart greenhouse management,” *Agronomy*, vol. 14, no. 2, p. 356, 2024.

Appendix A: Source Code Repository

A.1 Repository Overview

The complete source code of the proposed smart greenhouse monitoring and control system is available in a public GitHub repository. The repository contains all software components developed during this project, including:

- ESP32 firmware for sensor monitoring and MQTT communication.
- Arduino firmware for actuator control.
- React-based monitoring dashboard.
- MQTT broker configuration files.
- System documentation and installation instructions.

A.2 GitHub Repository

The complete project source code can be accessed at:

<https://github.com/aliba39/greenhouse-dashboard>

A.3 Repository Structure

```
smart-greenhouse/  
|  
+-- dashboard/  
|   +-- src/  
|   +-- public/  
|   +-- package.json  
|  
+-- esp32/  
|   +-- greenhouse_esp32.ino  
|  
+-- arduino/  
|   +-- greenhouse_arduino.ino
```

```
|  
+-- prototype/  
|   +-- images/  
|  
+-- README.md
```

A.4 Notes

- The ESP32 acts as the main controller of the system.
- The Arduino is responsible for actuator control.
- MQTT is used for real-time communication.
- The dashboard provides monitoring and manual control features.