Mohamed khider University of Biskra
Faculty of Science and Technology
Electrical Engineering Department

MASTER MEMORY

Science and Technology
Electronics
Electronics of Embedded Systems

Réf. : Entrez la référence du document

Presented and Doubted by:
Labbaz Abderrahmane **Benbouzid Mostefa**

At: 2025-06-03

Smart IOT Greenhouse

Jury :

Mr.	Guesbaya Tahar	**MCA**	University of Biskra	**President**
Mrs.	Tobbeche Souad	**Pr**	University of Biskra	**Supervisor**
Mrs.	Abida Toumi	**Pr**	University of Biskra	**Examiner**

Academic Year: 2024-2025

Resumé

Ce mémoire présente un projet qui vise à concevoir et à mettre en œuvre une serre intelligente basée sur la technologie de l'Internet des objets (IOT). Le système s'appuie sur un contrôleur ESP32 pour collecter des données à partir de capteurs tels que le capteur de température et d'humidité DHT22, le capteur d'humidité du sol et le capteur de gaz MQ135. Les données sont envoyées en temps réel à la plateforme cloud Thing Speak pour y être stockées, analysées et visualisées, ce qui permet une surveillance à distance. L'application Blynk est intégrée pour permettre à l'utilisateur de contrôler les appareils (par exemple, la pompe, le ventilateur, l'humidificateur) à l'aide d'un téléphone portable. Le système prend des décisions automatiques basées sur les relevés des capteurs afin de maintenir des conditions optimales pour la croissance des plantes.

Ce projet vise à accroître l'efficacité de l'agriculture, à minimiser l'intervention humaine et à soutenir l'agriculture durable grâce à des solutions intelligentes basées sur l'internet des objets, démontrant que les environnements agricoles traditionnels peuvent être transformés en systèmes intelligents et efficaces.

Keywords: IoT, smart greenhouse, ESP32, smart agriculture, environmental monitoring, ThingSpeak, Blynk, sensors, agricultural automation, remote control.

ملخص

تقدم هذه المذكرة مشروعا يهدف إلى تصميم وتنفيذ بيت محمي ذكي يعتمد على تقنية انترنت الأشياء(IOT). وذلك لأتمتة عملية مراقبة البيئة داخل البيت المحمي والتحكم فيها بشكل ذكي.يعتمد النظام على وحدة تحكم ESP32 لجمع البيانات من المستشعرات مثل مستشعر DHT22 لدرجة الحرارة والرطوبة , مستشعر رطوبة التربة , مستشعر الغازات MQ135 .يتم إرسال البيانات بشكل لحظي إلى منصة Thing Speak السحابية من اجل تخزينها وتحليلها وعرضها بطريقة مرئية , مما يتيح إمكانية المراقبة عن بعد , كما تم دمج تطبيق Blynk لتمكين المستخدم من التحكم في الأجهزة (مثل المضخة , المروحة , المرطب) من خلال الهاتف المحمول . ويتخذ النظام قرارات تلقائية بناءا على قراءات المستشعرات بهدف الحفاظ على الظروف المثالية لنمو النباتات.يسعى هذا المشروع لرفع كفاءة الزراعة وتقليل التدخل البشري , ودعم الزراعة المستدامة من خلال حلول ذكية قائمة على انترنت الأشياء , مما يبرهن على إمكانية تحويل البيئات الزراعية التقليدية إلى أنظمة ذكية وفعالة.

الكلمات المفتاحية : انترنت الأشياء , بيت محمي ذكي. ESP32 .الزراعة الذكية, مراقبة بيئية . Blynk. Thing Speak .المستشعرات, الأتمتة الزراعية, التحكم عن بعد.

Abstract

This dissertation presents a project to design and implement a smart greenhouse based on Internet of Things (IOT) technology. The system relies on an ESP32 controller to collect data from sensors such as the DHT22 temperature and humidity sensor, the soil moisture sensor and the MQ135 gas sensor. Data is sent in real time to the Thing Speak cloud platform for storage, analysis and visualization, enabling remote monitoring. The Blynk app is integrated to enable the user to control devices (e.g. pump, fan, humidifier) using a cell phone. The system makes automatic decisions based on sensor readings to maintain optimal conditions for plant growth.

This project aims to increase agricultural efficiency, minimize human intervention and support sustainable agriculture through smart solutions based on the Internet of Things, demonstrating that traditional agricultural environments can be transformed into smart, efficient systems.

Keywords: IoT, smart greenhouse, ESP32, smart agriculture, environmental monitoring, ThingSpeak, Blynk, sensors, agricultural automation, remote control.

Dedication

We dedicate this humble work to our generous parents who were a school of education and an oasis of morals.

To the professors who taught us the love of science and the passion for reading, as well as to all relatives and friends, and we do not exclude anyone from them.

To all those who contributed to this research from near and far

Acknowledgements

First and foremost, praise be to Allah, who has the credit first and last, outwardly and inwardly.

We praise and thank Him for His guidance and help in accomplishing this work despite the difficulties and challenges we faced, and I ask Him to make this effort purely for His face and to benefit us in our world and the hereafter.

We also extend our sincere thanks and gratitude to Professor Tobbeche Souad for her supervision of this work, who did not skimp on her knowledge and experience, and dedicated her valuable time and effort to us, whose guidance and valuable comments had a great impact in bringing this work to light in the best form.

I would like to extend my sincere thanks and appreciation to Professor Guesbaya Tahar and Professor Abida Toumi for agreeing to discuss our memorandum and allocating part of their valuable time to review and evaluate this work. We are honored by your presence and your scientific contribution, and I wish you success in your scientific career.

List of Contents

List of Tables

List of Figures

List of Graph

I. General Introduction

Agriculture is a cornerstone of human sustenance, providing essential food and raw materials. The advancement of this sector significantly contributes to the overall economic development of nations. One innovative approach to boosting agricultural productivity is greenhouse cultivation, which offers a controlled environment for the optimized growth of fruits and vegetables. Within a greenhouse, critical factors such as temperature, humidity, irrigation, light intensity, carbon dioxide levels, and nutrition can be precisely managed. This controlled greenhouse also shields crops from external threats like pests, diseases, and adverse weather conditions, ensuring more reliable production outcomes.

Greenhouses, therefore, represent an efficient solution for modern agriculture, facilitating improved plant growth conditions, disease prevention, and pest control. However, the potential of these systems can be further enhanced by integrating cutting-edge technologies[1]. This work focuses on leveraging intelligent technologies, particularly the Internet of Things (IoT), to innovate greenhouse cultivation. By utilizing IoT-enabled devices and a wireless network (WiFi), data from various sensors installed in the greenhouse will be collected and transmitted to cloud platforms like "Thing Speak" via HTTP protocols. These sensors and control systems will enable real-time monitoring, automation, and remote management of greenhouse environments.

The proposed intelligent system offers several benefits. It ensures optimal conditions for plant growth regardless of external climate fluctuations, allowing for year-round production. The system also enhances efficiency by automating key processes and enabling remote control, providing greater flexibility and convenience for managing greenhouse operations. By analyzing data from sensors, precise adjustments can be made to maximize productivity and optimize resource use.

The dissertation detailing this project is structured into three chapters, each elaborating on the key aspects of the study and its implementation.

- Chapter One: The first chapter presents a state of the art on the Internet of Things, smart agriculture, and smart greenhouses by detailing the environmental conditions such as temperature, humidity, etc. It also presents different irrigation systems and explains good water management practices in agriculture.
- Chapter Two: We review the most important factors affecting plant growth, sensors and devices used to create a system to control and manage the greenhouse.
- Chapter Three: We present the method used to implement this system.

Chapter I

Overview of Smart Agriculture and Irrigation Systems

I.1. Introduction

This chapter begins with a review of the state of the art in the Internet of Things (IOT). Next, it provides an outline of smart agriculture, highlighting its application and benefits. Then, the chapter presents the intelligent greenhouses that are a key component of modern agriculture, with particular attention to environmental factors such as temperature, humidity, and more. This chapter includes several irrigation methods and effective water management practices in agriculture.

I.2. Internet of Things

I.2.1. Definition of Internet of Things:

The Internet of Things (IoT), often referred to as the Internet of Everything (IoE), is a communication network that connects internet-enabled devices capable of gathering, transmitting, and processing data from their surroundings. These devices are equipped with embedded sensors, processors, and communication modules, enabling seamless data exchange commonly known as Machine-to-Machine (M2M) communication.

Expanding on this concept, the Internet of Things can be viewed as an advanced extension of the Internet, designed to enable interaction not only between users but also between physical systems, such as digital devices, machinery, and objects in the environment. Through specialized sensors, these devices collect and transmit data across the network, where it is processed to inform decisions or trigger specific actions carried out by the system's stakeholders[2,3].

Figure I.1: Internet of Things.[4]

I.2.2. History of the Internet of Things

The term "Internet of Things" was born in 1999 at the MIT center (Massachusetts Institute of Technology), thanks to Kevin Ashton, a British researcher, a pioneer in his field (IDO). His team launched the promotion of open connectivity of all objects using RFID tags (Radio Frequency Identification). Thanks to the appearance of the new IPv6 protocol, sectors such as aeronautics quickly seized the concept of the Internet of Things, and participated in research. This concept of the Internet of Things began to become popularin 2007. It was then considered to set up a Global, Ubiquitous Internet of Things. [4]

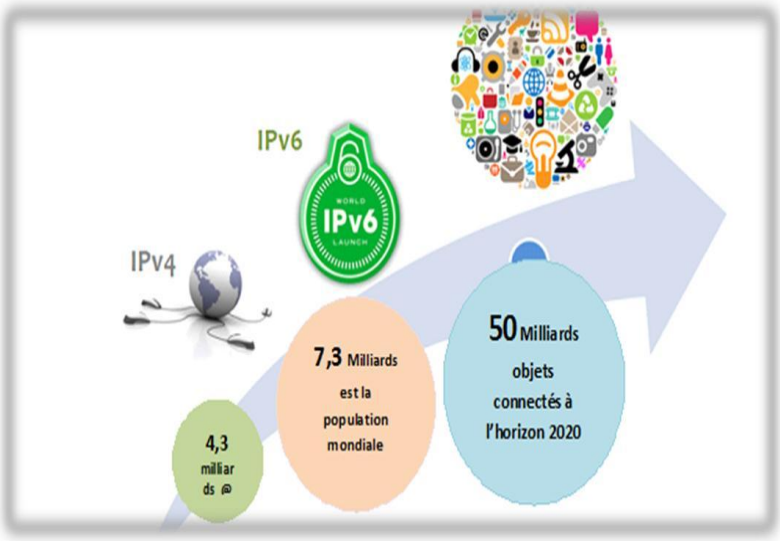


Figure I.2 : **Using the Internet of Things.[4]**

I.2.3. Application areas

The Internet of Things (IoT) offers immense potential due to its pervasive nature, enabling the development of numerous innovative applications. However, only a limited number of these applications are currently in use. In the future, IoT is expected to drive the creation of smart solutions across various sectors, including home automation, urban development, transportation, healthcare, and industrial processes. Key areas of impact include:

- Modern homes and smart buildings
- Energy management
- Transportation systems
- Healthcare advancements
- Industrial automation
- Agriculture

Sensors play a crucial role in IoT by detecting or measuring environmental variables and generating output signals. When these signals are received by controllers, such as West devices, they enable the display, recording, or regulation of processes, adapting the media's properties to meet specific application requirements. [3]

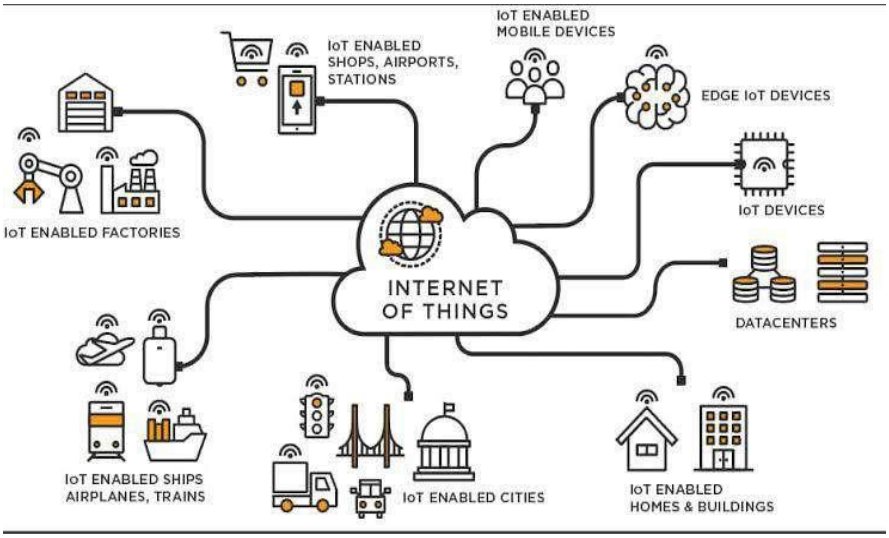


Figure I.3 :The Application areas of IOT.[3]

I.2.4. Characteristics

1) **Dynamic & Self Adapting:** According to their operational conditions, the context of the user, or their perceived surroundings, IoT devices and systems may be able to dynamically adjust to changing contexts and take appropriate action. For example, the surveillance system is adjusting itself in response to shifting circumstances and context.
2) **Self-Configuring** : permitting numerous devices to cooperate in order to provide a certain function
3) **Inter Operable Communication Protocols**: have the ability to communicate with infrastructure and other devices, and they support a variety of compatible communication protocols.
4) **Unique Identity**: Every Internet of Things device has an own identity and IP address.
5) **Integrated into Information Network** : that enable them to interact and share information with other systems and devices.[5]

I.2.5. IoT Architectures

IoT architectures are mainly classified into two categories such as IoT physical architecture (IPA) and IoT layered architecture (ILA).

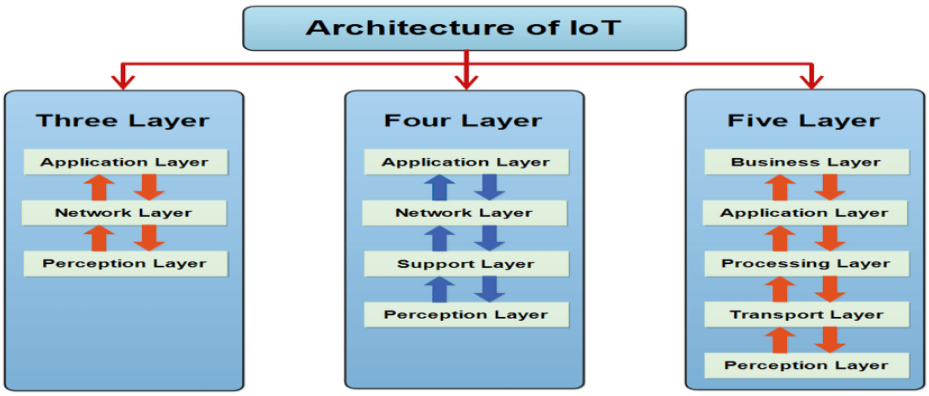
I.2.5.1. IoT Physical Architecture (IPA)

The architecture of IoT consists of four building blocks, they are :

- **Sensors**: There are sensors everywhere, and they all take in information from their surroundings.
- **IoT Gateways and systems:** As the name implies, it serves as a gateway to the internet for all of the devices and items that we must connect to it. It collects information from the sensor hubs and sends it online.
- **Cloud server**: Datais handled safely within the cloud server, such as in a server farm utilising data observation, and is transmitted through an entrance. With the aid of prepared information, astute operations are carried out to create gadgets and smart devices.[5]

I.2.5.2. IoT Layered Architectures

There isn't a universally accepted consensus among experts and the global community regarding the architecture of the Internet of Things. As seen in figure 1_4, researchers have presented a wide variety of architectures. Some academics claim that IoT architecture has three levels, however the four-layer layout is favoured by certain studies. They believe that the three-layer architecture is no longer able to meet the needs of applications because of the advancements in IoT. A five-layer architecture has also been suggested as a solution to the security and privacy issues in the Internet of Things. It is believed that a recently suggested design can satisfy the security and privacy needs of the Internet of Things.



FigureI.4 :IoT Architectures.[5]

I.2.5.2.1. IoT 3 Layered Architecture

Three-layer architectures are the most fundamental. It was first presented when this field of study was just getting started. The perception, network, and application layers are its three tiers.

- **Perception Layer:** Also referred to as the physical layer, this layer collects information and perceives the outside world. Every actuator in this layer operates in accordance with the data that is gathered by various items' sensors so that the relevant objects can carry out particular tasks.
- **Network Layer**: The middle layer, known as the network layer, creates an interface between the perceptual and application layers. It is in charge of the preliminary processing of data, data streaming, and device connectivity.

I.2.5.2.2. IoT 5 Layered Architecture

The addition of transportation, processing, and business distinguishes the Five Layer IoT Architecture from the Three Layer IoT Architecture.

The perception and application layers play the same function as the three-layer architecture layers. We describe how the other three layers work.

- **Transport layer:** The transport layer transfers information from the perception layer to the processing layer, which comes behind it, and vice versa. This will be accomplished with networks such as LAN, wireless technologies, RFID, LTE, 3G, and 4G, among others.
- **Processing layer:** The third layer, the processing layer, must complete the main duty since it will process all of the data that the perception layer has collected. The quantity of data is enormous.

  It will be kept using methods like cloud computing or any database management system. After that, it will determine how to retrieve data as needed to finish the intended activity.[5]

I.3. Importance of the Internet of Things in agriculture

Smart agriculture solutions are emerging as a result of the Internet of Things' (IoT) integration, which is transforming the agricultural industry. To address the many issues that farmers confront, almost 100 new and creative applications have been created with the goal of improving the quantity, quality, sustainability, and profitability of agricultural output.

IoT has many benefits, but its potential to transform conventional farming methods is particularly noteworthy. IoT sensors are essential because they give farmers up-to-date

information on soil conditions, pest outbreaks, rainfall patterns, and crop yields. This information helps farmers make well-informed decisions to enhance agricultural practices over time and is crucial for streamlining production operations.[2]

I.4. Smart Farming

I.4.1. Definition of smart agriculture

Smart farming is the application of intelligent information and communication technology systems such as sensors, IoT, cloud-based processes, machine learning, artificial intelligence, networking to the farming system such as crop cultivation, livestock farming, aquatic, snail farming just to mention a few with the sole purpose of boosting the farm produce. It can be inferred that smart farming involves the implementation of both technological software and hardware solutions to improve the farm's outcome. According to farmers in the past years have tilled the soil using holes, animals to power the plow, used bush burning practices to clear farmlands for planting. Some have used animal waste for manure, but today fertilizers are used, which are rich in nitrogen, potassium, and many more minerals to make the soil suitable for effective farming practices. It can be inferred that new farming practices have changed over the years, from using holes and cutlasses to machine tilling the fields and machine harvesting the crops. In this respect, smart farming has introduced a more efficient technique where farmers use IoT to improve all farming practices and methodologies. Today farmers can monitor remotely their farmers many kilometers away from their farms and remotely activate actuators using IoTs installed on the farms.[6]

Figure I.5:Traditional Agriculture in Vietnam.[7]

I.4.2. The impact of climate to smart farming

There are some key factors influencing surface fluxes and soil heat storage, such as energy consumed during photosynthesis and advection conditions, showing that heat storage enhances energy balance closure. Findings indicate that higher surface heat fluxes occur in thinner, well-watered canopies with regular advection. However, limitations include short data collection periods and the need for early-season data capture for more accurate results. Research also reveals that temperature and photoperiod impact leaf senescence, influencing vegetation coloration and the carbon cycle, but lacks detailed analysis on the rate of color change. Additionally, the photochemical reflectance index (PRI) is effective in detecting late-stage heat stress in wheat, though it has not been tested on other crops. The implementation of a smart surface sensing system (4S) shows promise for remote vegetation monitoring, but its inability to monitor multiple remote sites is a limitation. Studies on winter wheat suggest that crop yield is higher in certain latitudinal regions, impacting agricultural production and market dynamics. However, models used in these studies have not been applied across different crops and regions, limiting their forecasting ability. Moreover, research on vegetation indices (VI) and Gross Primary Productivity (GPP) indicates that monthly data capture provides better accuracy than daily data, as short-term VI fluctuations reduce reliability, especially in dry land ecosystems. Overall, farming and climate share a dynamic relationship, where climate conditions influence crop growth and productivity. While studies provide valuable insights, limitations such as short data collection periods and restricted crop applicability hinder broader generalization and predictive accuracy.[6]

I.4.3. The benefits of smart agriculture

We may state that smart agriculture has a number of advantages and objectives. The preservation and

FigureI.6:Smart Agriculture.[7]

protection of the environment through improved natural resource management is crucial, above all.

• Adaptation plans that tackle the effects of climate change.

• Efforts to reduce greenhouse gas emissions through mitigation.

• Reducing poverty and hunger.

• Improving the quality and quantity of agricultural products.

• The application of sustainable methods for managing natural resources.

• Improvements to management practices and soil fertility.

• The production of biogas, a green energy source, from animal waste.

• The creation of robust aquaculture and fisheries systems, such as storm-resistant enclosures and flexible management techniques, to fight climate change.[2]

I.4.4. A Systematic Review of IoT Solutions for Smart Farming

several related works are being developed in recent years. This rich literature has already been analyzed by the academia from multiple perspectives with objective of determining the state of the smart farming development. Thus presented a systematic review of precision livestock farming in the poultry sector and made a review of state of the art of technologies used in precision agriculture, focusing in the innovations, measured parameters, technologies and application areas. On the other hand has focused on the use of big data as a tool to support agriculture, pointing out the main opportunities and challenges of using this technology.[8]

I.5. Smart greenhouse

A greenhouse, sometimes referred to as a glasshouse or hothouse, is a structure used for growing plants. Built of glass or plastic, a greenhouse warms up as a result of the sun's entering UV rays warming the soil, plants, and other interior components. The structure, the roof and walls of the building keep the air heated by the heat from the heated interior surfaces. The glass used in greenhouses acts as a selective transmission medium for certain spectrum frequencies, trapping energy inside the structure and heating the ground and plants within.[9]

Figure I.7 : smart greenhouses.[2]

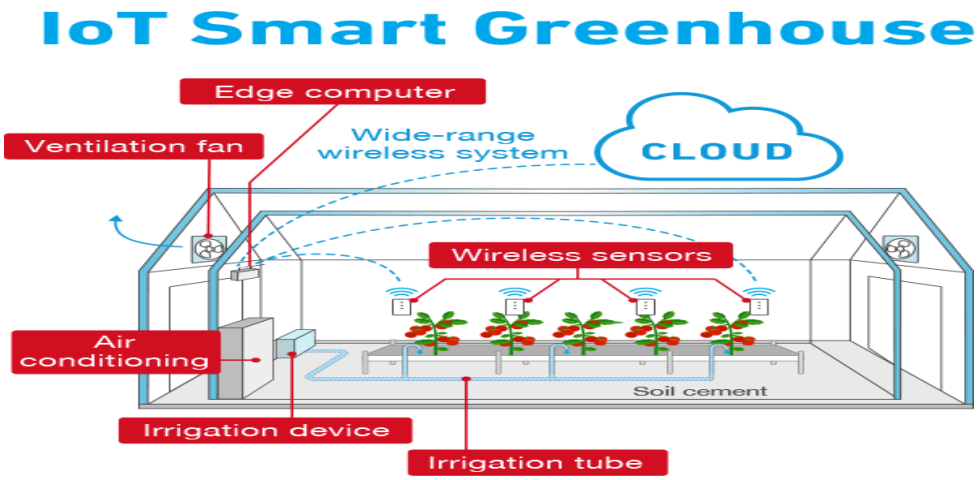


Figure I.8: IoT Smart Greenhouse.[7]

I.6. Automation Control system for Greenhouse Environment

In its contemporary sense, automation is a technology that use preprogrammed orders to carry out a certain procedure along with information feedback to verify that the directions have been carried out correctly. The procedure can function without human intervention once it is automated. In actuality, the majority of automated systems can complete their tasks faster and with more accuracy and precision than humans. Automation is meant to give engineering systems human-like perception, reasoning/learning, communication, and task planning/execution abilities. Fixed automation and flexible automation are the two main types of automation.[9]

I.6.1. Systems

The process of system analysis and integration begins with the definition of a system and its objectives and ends with a conclusion about the system's dependability, productivity, workability, and many performance metrics. knowledge on the many parts of the system and how they interact, as well as techniques for collecting and processing data to provide knowledge with added value, are two essential resources in systems analysis.[2]

I.6.2. Environment

The environment surrounding plants includes climatic, nutritional, and structural conditions, all of which play a crucial role in plant growth. Controlling these environmental factors has become a significant engineering challenge in modern plant production research. The complexity of climate control depends on the physical barrier separating the controlled environment from external conditions, influencing the extent of hardware, software, and actuator systems required for regulation. Greenhouse automation faces several challenges, including ensuring a profitable return on investment, optimizing automation through proper system integration (ACESYS concept), balancing fixed and flexible automation, maximizing machine utility, addressing market limitations, and continuously advancing research and development efforts.[2]

I.6.3. Greenhouse Environmental Control considerations

Greenhouse production, a key form of controlled environment plant production, requires engineers to design efficient structures, climate control systems, plant growth setups, and automation technologies to support growers and ensure economic viability. Effective environmental control involves regulating temperature, humidity, and CO_2 levels to optimize plant growth and improve crop yield and quality. The ACESYS approach integrates automation, plant culture, and environmental management for seamless operation. Since greenhouses are highly influenced by solar radiation and external temperatures, maintaining stable internal conditions is essential for consistent, high-quality crop production.[2]

I.7. Environmental conditions in greenhouses

I.7.1. Humidity Control

Effective humidity control in greenhouses is essential for reducing plant diseases, enhancing water and nutrient absorption, and promoting healthy growth. Despite being a standard feature in greenhouse control systems, accurate humidity sensing remains a challenge due to the limitations of

commercial sensors. Plants release significant moisture through transpiration, with a single tomato vine emitting nearly 2L per day, leading to substantial humidity buildup in large greenhouses. Proper ventilation is crucial to expel excess moisture and prevent condensation, while irrigation is necessary on hot days to support plant cooling. Humidity levels can be adjusted through ventilation, sometimes requiring additional heating, or increased using fogging systems, with wet and dry bulb thermometers used for monitoring.[9]

I.7.2. Carbon Dioxide Management

CO_2 generators are commonly employed in commercial greenhouses to optimize plant productivity. However, the effectiveness of CO_2 supplementation depends on it being the "limiting factor." This implies that unless all other growth variables such as light, fertilizer, temperature/humidity, and pH are adequately balanced, the benefits of increased CO_2 levels may not be fully realized .[2]

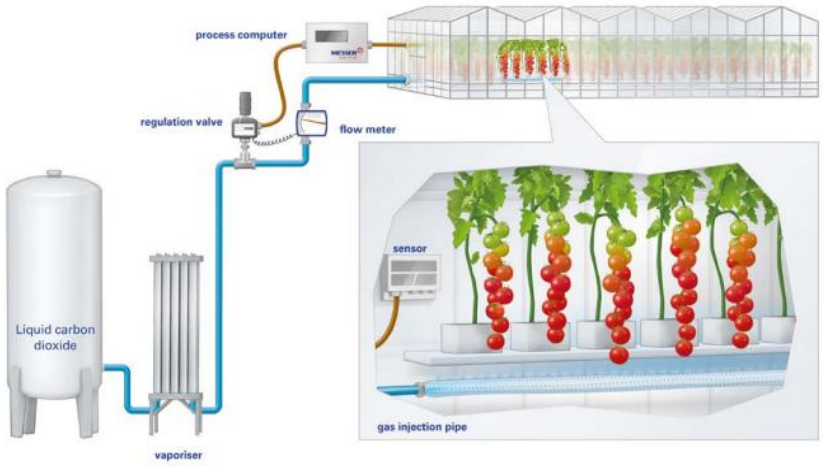


Figure I.9:carbon dioxide injection.[2]

I.7.3. Temperature Control

Techniques of cooling can be organized into several categories; each utilizes the evaporative cooling process to reduce air temperature, as well as fan ventilation for exchanging the moist air with dry outside air. Greenhouse cooling reduces plant stress caused by high leaf and air temperatures. The root and stem system may not be able to supply adequate water to the leaves, thereby limiting transpiration, the plant cooling mechanism. Also, hot and humid air around the leaves will reduce the effectiveness of transpiration at the leaf surface. [2]

Greenhouse temperature control can be accomplished a number of ways:

1. Ventilation (Natural & Mechanical)
2. Evaporative cooling
3. Shading

I.7.4.Window adjustment

Temperature, humidity, and light levels. Adapting window openings based on environmental conditions ensures that plants receive the necessary resources for healthy growth.

These factors interact with each other synergistically, and their careful management is vital for creating an ideal growth environment and maximizing productivity in greenhouse cultivation.[2]

I.8. Principles of Climate Control in Smart Greenhouses

DG-12 Heating & Cooling Technology presents an innovative approach tailored for greenhouse environments;this unit efficiently manages greenhouse conditions by regulating humidity and temperature, offering a dual-purpose functionality for both heating and nighttime cooling. Its design facilitates gradual temperature adjustments, optimizing energy usage. Integrating dehumidification with heating and cooling capabilities ensures consistent climate control, minimizing the need for extensive infrastructure compared to alternative systems. The unit relies on an external source of hot or cold water for its heating and cooling operations. The DryGair team, ensuring a bespoke solution for each application, individually computes detailed specifications. Versatile placement options include central or peripheral positioning within the greenhouse, either floor-mounted or suspended. The unit features the Smart DG system for remote monitoring and control, with the subscription available separately. DryGair dehumidifiers are the result of collaborative efforts between agronomists and climate engineers, dedicated to mitigating humidity challenges in cultivation environments. Our team provides personalized assistance to growers, guiding them in selecting the optimal solution tailored to their specific facility and conditions. Manufactured to the highest standards, DryGair products adhere to Good Manufacturing Practice (GMP) regulations. The DG-12 model is available in two variations, operating at 50 Hz and 60 Hz frequencies, ensuring compatibility with growers worldwide .[2]

FigureI.10: DG-12 Heating &Cooling.[2]

I.9. Definition of irrigation

Irrigation is the artificial application of fresh water to land for agricultural purposes, serving as a controlled form of precipitation. It can be automated, such as sprinkler irrigation, or manually applied. Irrigation plays a vital role in promoting crop growth, maintaining landscapes, and restoring vegetation in arid regions or during dry periods. By providing essential moisture, irrigation enhances soil fertility and supports sustainable agriculture.[10]

Figure I.11: Irrigation system.[7]

I.9.1. Types of irrigation

- **Basin irrigation:** Basins are made up of earth cuvettes that are relatively flat and surrounded by diguettes that are either low or high. These levées are intended to stop water

from flowing into nearby fields. Generally speaking, this technique is used to irrigate terraces with a coteau flanc or rizières on flat ground. A small cuvette, known as a bassin, is placed around each tree in the bassin method, which is also used to irrigate fruit trees. This irrigation strategy is generally applicable to any cultures that can withstand prolonged exposure to water (e.g. 12-24 hours).[11]

Figure I.12:Irrigation by basin.[11]

- **Sillon irrigation**, also known as raie irrigation, uses little rigoles in the ground that are arranged according to the slope of the land to move water between cultural ranges. Over the course of its journey, water seeps into the soil, primarily from the sillon's sides, in the sense of the terrain.

Plants are typically grown on the billons that divide the sillons.

This method can be used for online watering of any culture as well as for any culture that does not tolerate prolonged exposure to water from their foliage or collection (e.g. 12-24 hours). Water ponds situated on the berges of the amenée canal are used to feed the sillons. Such taking works may consist of siphons, simple openings positioned on the amenée canal's cliffs, or even tuyaux of food that pass through the amenée canal's cliffs.[11]

FigureI.13: Stingray Irrigation.[11]

- **Irrigation by planches:** planches are land bands that are divided by diguettes and surrounded with soft pente. They are also known as planches d'arrosage or calant. Planches can be nourished with water in a number of ways, including by using water intakes placed on the amenée canal and equipped with a vannette, siphons, or even tuyaux that travel through the amenée canal's berges. Ruisselle's water was introduced by descending the planche's pente, guided by the diguettes on both sides.[11]

FigureI.14 : Irrigation by beds.[11]

I.10. Water use efficiency

Water logging near distribution boxes and divisions is caused by excessive water discharge and improper drainage at the canal's tail end. Enhancing irrigation management is essential since excess water can be just as harmful as a lack. Accurate discharge measurements are required in order to match crop needs with water supply. The wasteful use of water in connection with agricultural inputs like fertilizers, insecticides, and herbicides is another significant problem. Agrochemical leaching is accelerated by high discharge rates, which leads to production losses, higher expenses, and contaminated drainage system water. To overcome these obstacles, irrigation and input management must be harmonized in order to maximize resource utilization and reduce environmental effect.[12]

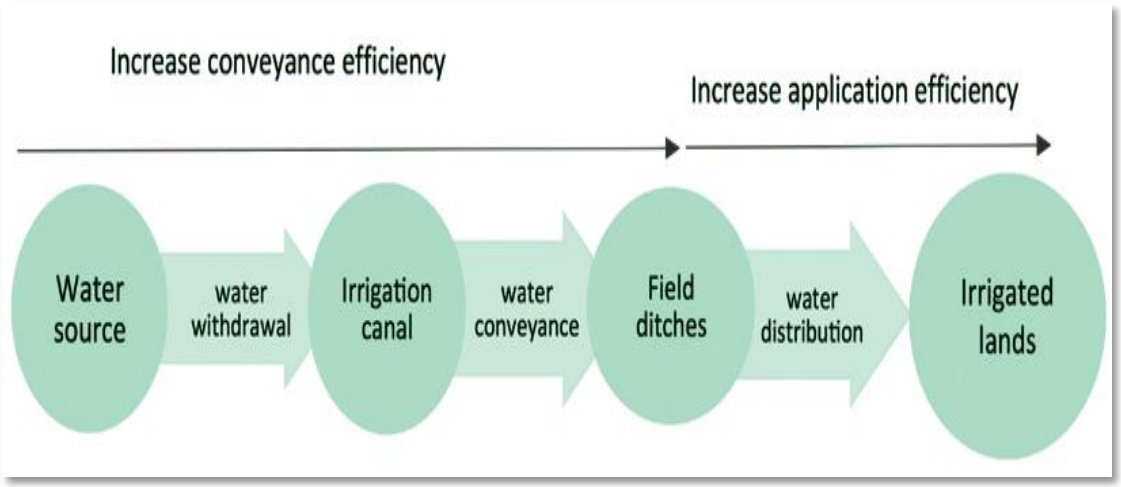


Figure I.15: Flow chart of water use efficiency.[12]

I.11. Water use efficiency measures for improved irrigation practices

The Guide provides a step-by-step approach to improving small-scale irrigation practices, drawing insights from pilot cases in Burkina Faso, Morocco, and Uganda. However, since farms have unique conditions and cropping patterns that change over time, water use efficiency measures are not universally applicable. Instead, these measures should be implemented flexibly to address site-specific challenges.

To support irrigation practitioners, the Guide presents both operational and management options, encouraging innovative thinking. It is structured around three key areas:

- An overview of various water use efficiency measures applicable to irrigation schemes.

- Demonstrations of different combinations of these measures through country case studies.
- Emphasis on the critical role of water monitoring in enhancing agricultural yields.

Beyond serving as an instructional tool, the Guide fosters knowledge exchange among irrigation practitioners. It promotes a continuous learning process where scheme managers can develop and modernize their irrigation systems based on shared experiences and best practices.[12]

Conclusion

We discussed the Internet of Things (IOT) in this chapter, including its uses, domains, and significance for the advancement of smart greenhouses and agriculture. Along with discussing how to conserve and maximize water, we also covered the idea of irrigation, its varieties, and its significance in enhancing crops. We stress the necessity of stepping up efforts to accelerate the transition to smarter and more sustainable agriculture as well as the significance of investing in the development and use of smart technology in this sector.

Chapter II

Devices used to create a greenhouse control and management system

II.1. Introduction

Traditional greenhouse monitoring systems typically rely on wired connections and complex wiring, which can be inefficient and cumbersome. Wireless transmission technology has therefore become a critical necessity, particularly for enabling greenhouse sensors. Wireless sensors not only simplify monitoring and save time but also significantly reduce the need for human labor.

In this chapter, we will explore the microcontrollers, sensors, and cloud platforms encompassing both hardware and software elected in our project to efficiently manage and optimize the greenhouse climate system.

II.2. Sensor

Three primary components of our study are soil moisture, greenhouse temperature, and greenhouse humidity, all of which we regularly monitor.[2]

II.3. The Soil Moisture Sensor

II.3.1. Capacitive sensor

It consists of an electrically sensitive material placed between two electrodes, forming a capacitor whose capacity varies with relative humidity changes. These sensors can provide extremely accurate atmospheric measurements. These sensors are commonly used in both industry and meteorology.[13]

Figure II.1 : capacitive soil moisture sensor.[13]

II.3.2. Features:

Measurement range: from 0% to 100%

Speed: they have a fast response time

Temperature range: -70°C to +200°C)

Size: they are small and easy to integrate

They are stable, accurate and reliable.

They are resistant, for example, to chemicals.[13]

II.4. Humidifier

The miniature ultrasonic humidifier is essential to our project's ability to maintain ideal humidity levels in the greenhouse-like setting. It helps produce an environment that is conducive to plant growth by adding moisture to the air, particularly during dry seasons or in areas with low natural humidity. The humidifier's small size makes it simple to incorporate into our prototype greenhouse setup, guaranteeing that plants get the moisture they require for wholesome growth.[2]

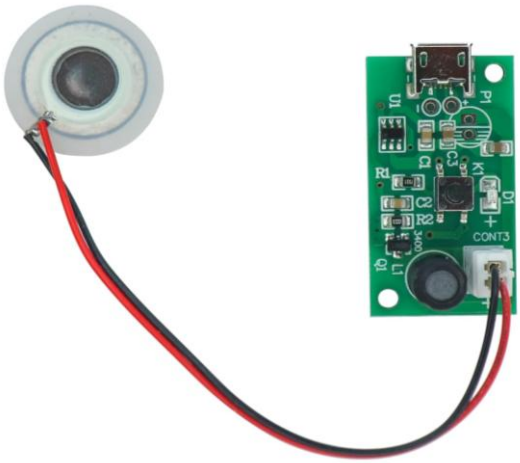


Figure II.2:Ultrasonic Humidifier.[2]

II.5. Air Humidity Sensor-DHT22

We used a DHT22 humidity and temperature sensor, as shown in figure (II.3).It is low-cost and uses a capacitive sensor to measure the humidity in the air. It also uses a thermistor to measure temperature. Data can be obtained from the data pin of the DHT22 .The DHT22 is suitable for humidity readings from 0-99.9 % with an accuracy of ±2% .[14]

Figure II_3 : DHT22 air humidity sensor.[14]

II.6. Light Sensor LDR

A light sensor is a widely used electronic component, with the photo resistor or Light Dependent Resistor (LDR) being one of the simplest types. This light-sensitive resistor changes its resistance based on the light it receives. As the light intensity increases, the resistance of the LDR decreases, often varying across a wide range. This makes LDRs a fundamental and versatile tool in detecting and measuring light levels.

II.6.1. Advantages

• Sensitivity is High.

• Simple & Small devices.

• Easily used.

• Inexpensive.

• There is no union potential.

• The light-dark resistance ratio is high.

• Its connection is simple**.**

II.6.2. Disadvantages

• The spectral response is limited.

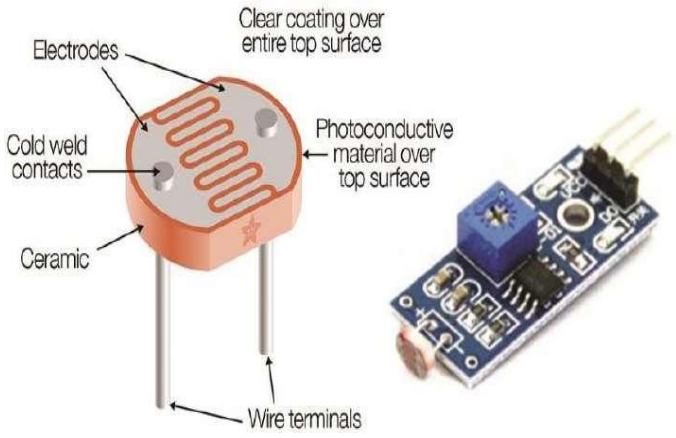
• The best materials have limited temperature stability due to the hysteresis effect.

• Its chemical reaction in stable materials.

• LDR Sensor is only used in situations when the light signal fluctuates dramatically.

• It is not a particularly responsive tool.

• As soon as the operating temperature changes, it gives the wrong results.



FigureII.4: Light Sensor LDR.[3]

In the IoT greenhouse monitoring system project, the LDR (Light Dependent Resistor) module is employed to detect changes in light intensity. It serves a significant role in managing the illumination system within the greenhouse. The microcontroller receives input from the LDR sensor measuring light intensity and uses the threshold to determine whether to switch on or off the lights. This guarantees that the plants get the ideal quantity of light for development and growth. The LDR module in the IoT greenhouse system measures light intensity and communicates data to the microcontroller, which turns grow lights on or off based on predefined thresholds, ensuring optimal lighting conditions for plant growth.[15]

II.7. Relay

Using a relay in a greenhouse automation system offers precise control over various devices, such as pumps, ventilation systems, and lighting. The relay figure(II.5) can be programmed to perform specific tasks at designated times, such as irrigating plants, ensuring proper airflow, or providing adequate lighting for plant growth. This integration enhances system efficiency, conserves energy, and helps maintain optimal conditions for plant cultivation.

The relay switches on/off devices like incandescent lamps or air blowers to maintain the desired temperature and humidity levels in the greenhouse. The relay activates a water pump to irrigate the plants when the soil moisture sensor detects low moisture levels. The

relay turns on/off grow lights based on the ambient light intensity measured by a light sensor to provide the plants with the required photoperiod. The relay operates exhaust fans or vents to regulate the CO_2 levels and prevent heat buildup in the greenhouse. The relay sounds a buzzer or sends alerts when sensor readings go beyond the safe thresholds, allowing the user to take corrective actions.[3]

Working Principle: Relays work by using a low voltage signal to control a higher-voltage circuit, allowing devices like lights, fans, or pumps to be turned on or off based on sensor inputs. In the IoT greenhouse system, relays help automate environmental control for optimal plant growth by interfacing with a microcontroller to manage conditions like temperature, humidity, soil moisture, and lighting.[15]

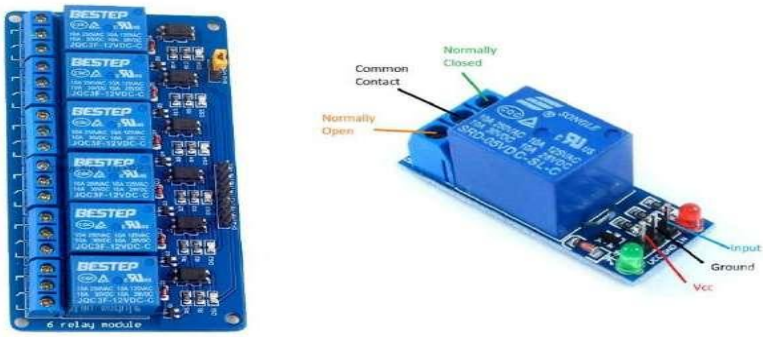


Figure II.5:Relay 6 channel.[3]

II.8. Irrigation Pump

An irrigation pump Figure(II.6) is a key device in agricultural irrigation, designed to draw water from a source like a well or reservoir and distribute it to crops. Its primary function is to supply sufficient water at the necessary pressure to meet plant hydration needs. Powered by a motor, the pump moves water through irrigation pipes to the designated agricultural areas. Irrigation pump scome in various sizes and capacities, with the selection depending on the farm's size and specific irrigation requirements.[3]

FigureII.6: Irrigation Pump.[7]

II.9. Ventilation fan

Air removal equipment in greenhouses includes tools designed to ventilate and expel excess air, maintaining optimal growing conditions. Among these, the ventilation fan (Figure (II.7)) is particularly important. It facilitates air circulation, removing excess moisture and heat, which helps prevent humidity-related issues and diseases. By improving airflow, the fan ensures better air quality and creates an environment conducive to healthy plant growth.[3]

Figure II.7: Ventilation fan.[3]

II.10. MQ135 Air Quality Sensor

The MQ135 air quality sensor (Figure (II.8)) is a versatile MQ gas sensor designed to detect and monitor various gases, including ammonia, alcohol, benzene, smoke, and carbon dioxide. It operates on a 5V supply with a current of 150mA and requires a 20-second preheating period to ensure accurate readings.[3]

Figure II.8: MQ135 Air Quality Sensor.

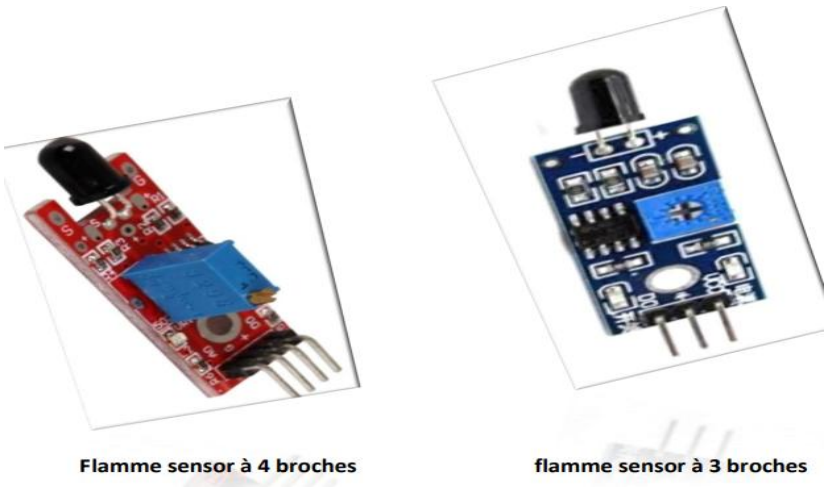
II.11. Features and Specifications of the MQ135 Sensor

The MQ135 sensor is designed with the following characteristics:

- Wide detection range with high sensitivity, fast response, long lifespan, and stability.
- Operates on a +5V power supply.
- Detects gases such as NH3, alcohol, NOx, benzene, CO2, and smoke.
- Functions as both an analog and digital sensor, with an output voltage range of 0V to 5V (TTL logic).
- Heating voltage: 5V ± 0.1, with a heater resistance of 33 ohms ± 5%.
- Adjustable load resistance and heating power consumption below 800mW.
- Operates within a temperature range of -10°C to 45°C and can be stored between -20°C and 70°C.
- Supports humidity levels below 95% RH and requires 21% oxygen concentration for optimal sensitivity.
- Sensing resistance ranges from 30 kΩ to 200 kΩ, with a concentration slope rate ≤ 0.65.[3]

II.12. The flame sensor

The infrared flame detector is designed to detect open flames by responding to the light emitted during combustion. It can distinguish flames from other light sources based on the sensitivity to specific optical wavelengths and flame pulse frequencies. There are two types of flame detectors: 3-pin and 4-pin. These detectors provide either analog or digital output.[16]



FigureII.9: the different types of sensor flame. [16]

Operating Principle

The module consists of three main components: a sensor, an amplifier, and a potentiometer. The sensor performs the measurement, producing an analog signal, which is then sent to the amplifier. The amplifier boosts the signal based on the gain set by the potentiometer and sends it to the module's output. Notably, the signal is inverted: a higher sensor measurement results in a lower output voltage [15]. This sensor only sends out a weak signal once the sensor detects information (such heat or light). It sends this signal to the amplifier to boost its strength because it is insufficient for immediate usage. But before the signal goes out, it passes through a potentiometer that we may use to modify the strength of the signal as needed. This system's special feature is that the output voltage and sensor reading have an inverse relationship; the higher the reading, the lower the output voltage. This is helpful in many applications where an inverse reaction is required.

II.13.Water Level sensor

The water sensor operates using a series of parallel wires exposed to the amount of water. By measuring the changes in the sensor’s analog output values, it can detect the presence and size of water droplets across the wires. The sensor features low power

consumption and high sensitivity, allowing it to be connected to a microprocessor or other logic circuits. Because of its extra-flat design, this kind of sensor is typically used for rain detection. Nevertheless, because of its small size, it is employed in our situation to measure the amount of water in the water basin [18].

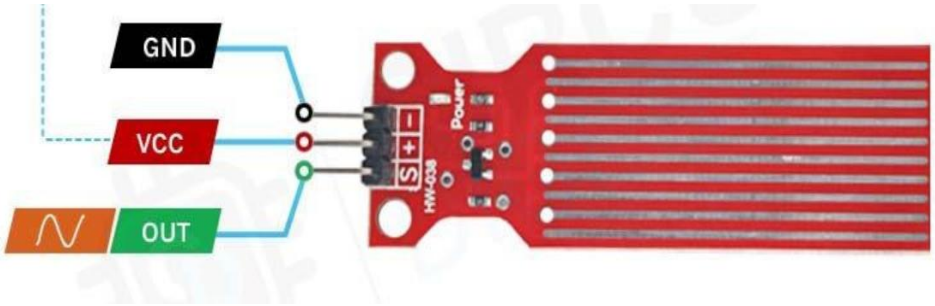


Figure II.10: water level sensor.[18]

Features :

- VCC: 3 V-5V
- GND: GND
- Output (analog output)

II.14.ESP32

II.14.1. Presentation of ESP32

The ESP32 is a family of inexpensive, low-power microcontrollers with built-in Bluetooth and Wi-Fi capabilities. The ESP32 series incorporates integrated antenna switches, a power amplifier, a low-noise receive amplifier, filters, and power management modules. It uses either a dual-core or single-core TensilicaXtensaLX6 microprocessor, a dual-core Xtensa LX7 microprocessor, or a single-core RISC-V microprocessor. The ESP32 was designed and developed by Espressif Systems, a Shanghai-based company, and is manufactured by TSMC, a Taiwanese corporation [18].

II.14.2. Features

Number of hearts	2 (dual core)
Wi-Fi	**2.4 GHz up to 150 Mbits/s**

Bluetooth	BLE (Bluetooth Low Energy) and legacy Bluetooth
Architecture	**32 bits**
Frequency	**240 MHz**
RAM	**512 KB**
Pines	30, 36, or 38 (depending on the model)
Peripherals	Capacitive touch, ADC (analog todigital converter), DAC (digital toanalog converter), I2C (Inter-Integrated Circuit), UART (universalasynchronous receiver/transmitter), CAN 2.0 (Controller Area Network),SPI Serial Peripheral Interface), I2S(Integrated Inter-IC Sound), RMII(Reduced Media-Independent interface), PWM (pulse width modulation),and more.

Table II.1: ESP32 features[18].

Figure II.11: ESP WROOM 32-38 Pin Parts.

II.15. Software Presentation

II.15.1. Arduino Software (IDE)

The Arduino Software IDE (Integrated Development Environment) is a platform-application used to program microcontrollers, including ESP32 that allows you to write, compile, and upload code to ESP32 development boards using the familiar Arduino programming language (C/C^{++}). Arduino IDE is an open source and official Arduino software, which make programming code easier, even for people who have no prior knowledge. It is available in operating systems such as MAC, Windows, and Linux. The IDE generally consists of two basic parts: Editor and Compiler. The former is used for writing the code and the latter is used for compiling and uploading the code into the ESP32 boards [14]. When we start the IDE, a window shown in figure II.10 will appear.

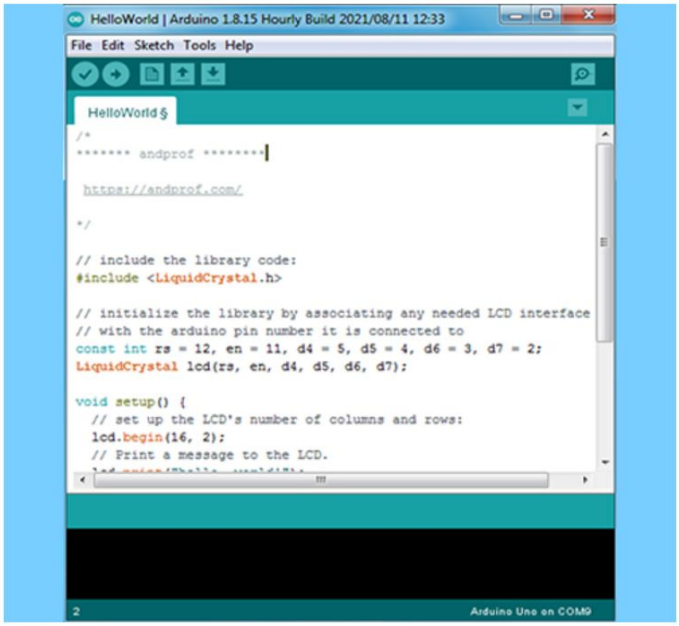


Figure II.12: New sketch in Ardunio IDE.

Menus section:

Figure II.13: Menus section.

Menus are the main menus of the program, and they are 5 menus (File, Edit, Sketch, Tools, Help), and they are being used to add or modify the code that you are writing.

Toolbar section:

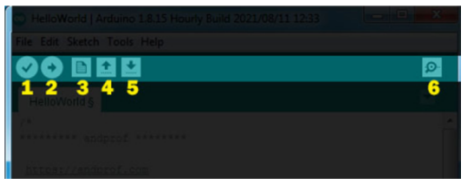


Figure II.14: Toolbar section

The toolbar is the most important section in the Arduino software, because it contains the tools that you will use continuously while programming the Arduino board. These tools are:

- Verify: this button use to review the code, or make sure that is free from mistakes.
- Upload: this button is use to upload the code on the Arduino board.
- New: this button use to create new project, or sketch (sketch is the file of the code).
- Open: is use when you want to open the sketch from sketchbook.
- Save: save the current sketch in the sketchbook.
- Serial monitor: showing the data which have been sent from Arduino.

Code editor section:

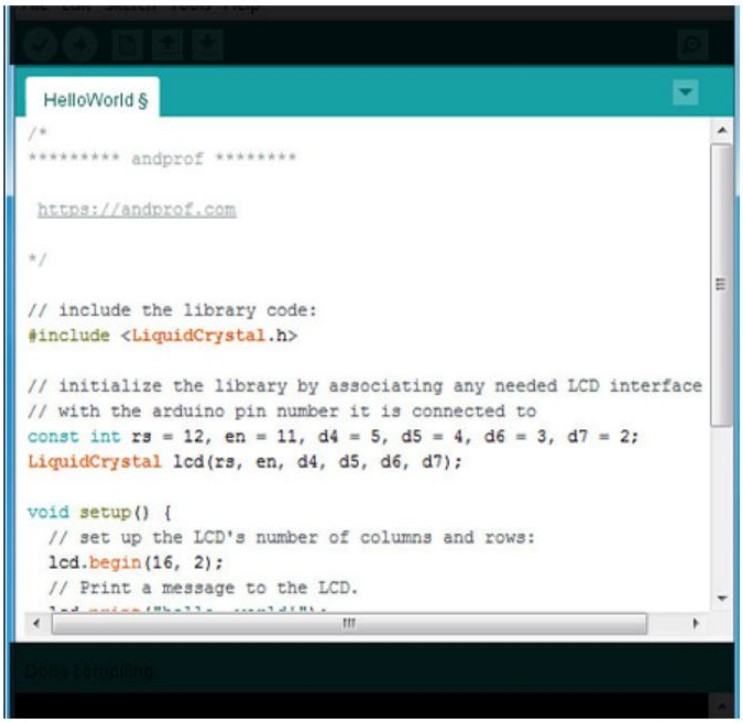


Figure II.15: Code editor section

Code editor is liberator of codes, it is the white space in the program, in which codes are been writing, and modifying on it.

Status bar section:

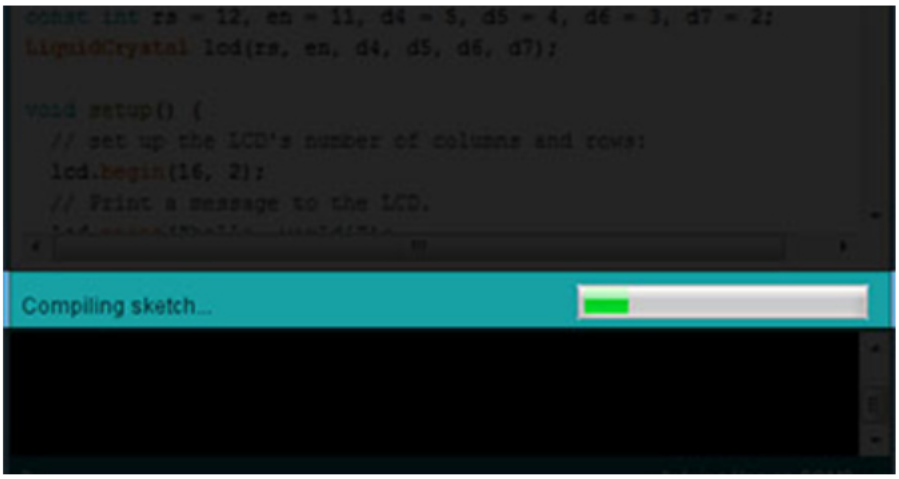


Figure II.16: Status bar section

Status bar is a space which can be found down the code editor, through it showing the status of operation's completion (compiling, uploading, …etc).

Program notifications section:

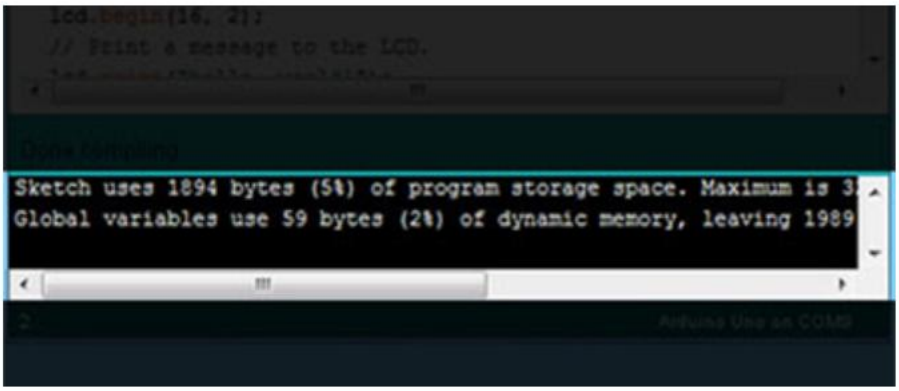


Figure II.17: Program notifications section

Program notifications: this program showing you the mistakes of codes and some problems that can be face you during the programing process. It clarifies to you the type of the mistake or the problem, which happened. It presents some instruction through it, which you have to apply to process the mistake or the problem.

Serial port & Board selections:

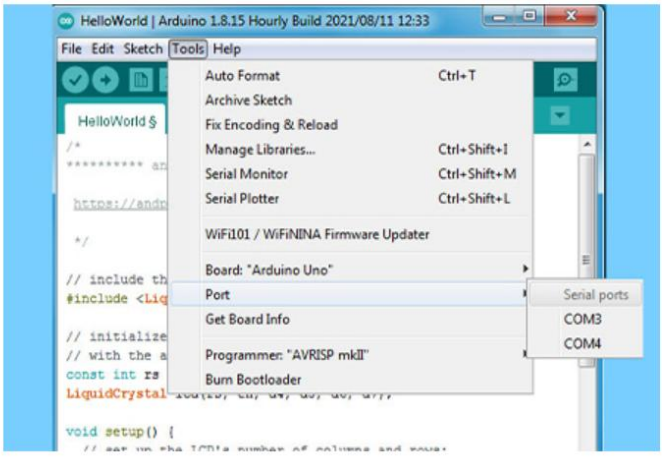


Figure II.18: Serial port & selections

Serial ports selections is a space in which the program showing you the type of the port which is used to connect the Arduino by computer. Board selections is a space in which the program showing you the type of the Arduino board.

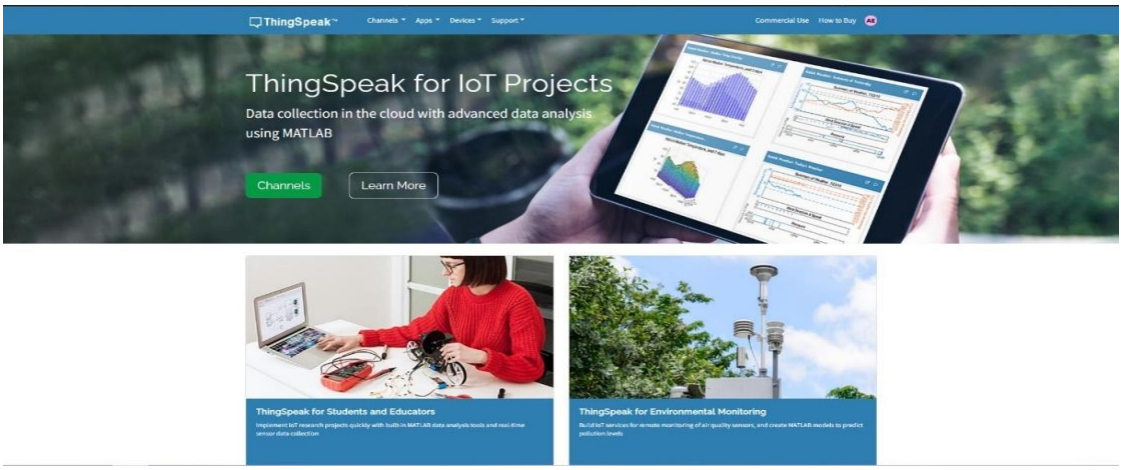
II.15.2. Cirkit Designer

Before proceeding with actual implementation, the Cirkit Designer platform was utilized as an electronic simulator to design the smart greenhouse circuit and test the interplay of the different components (ESP32 microcontroller, sensors, and actuators) as part of the project's design and preliminary validation phase. An enhanced development environment that combines theoretical design with practical programming is offered by the Cirkit Designer platform, a visual tool that makes it easier for users to generate electronic schematics and incorporate them into programming.

Figure II.19: Logo of Cirkit designer.

II.16. Thing Speak Cloud

ThingSpeak, offered by Math Works, is an IoT platform and cloud service that facilitates the collection, storage, analysis, and visualization of data from connected devices or sensors. It simplifies the development of IoT applications through an



intuitive interface and various features for data management and visualization.

Figure II.20: save data to ThingSpeak Cloud.

Key Features

- **Data Collection**

  Thing Speak collects data from IoT devices, sensors, or external sources using Restful APIs for secure data transmission. It also supports MQTT for efficient data transfer.

- **Channel-based Data Storage**

  Data is organized into channels, each representing a specific device or source. Users can customize fields within a channel to store different data types, such as temperature, humidity, or GPS coordinates.

- **Data Visualization:**

  Thing Speak offers tools for visualizing data through customizable charts, graphs, and gauges. This helps users monitor sensor readings, track trends, and detect patterns or anomalies.

Data Processing and Analysis

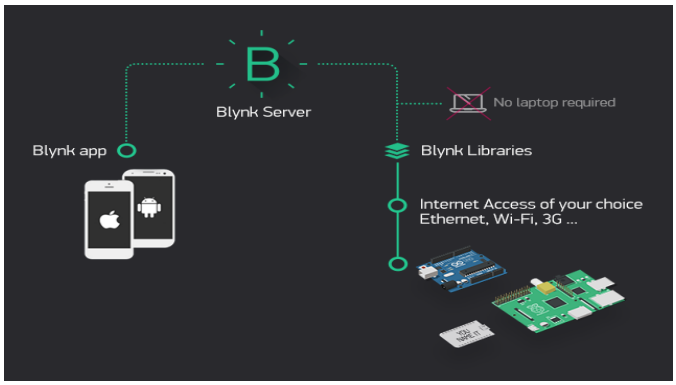


Figure II.21: Blynk architecture[13].

The platform supports MATLAB analytics, enabling advanced data processing and analysis. Users can run custom MATLAB scripts or use predefined functions to gain insights and trigger actions based on conditions.

- **IoT Integrations**
  Thing Speak integrates with popular platforms like IFTTT, allowing seamless data exchange, automation, and triggering actions based on specific events.
- **Real-time Data Access**
  The platform provides real-time access to data through APIs, making it easy to retrieve and integrate data for monitoring or external systems.
- **Open and Extensible**
  Thing Speak is an open platform that supports custom plugins and MATLAB code integration, allowing developers to extend its capabilities and implement specific data processing functionalities based on their needs.[3]

II.17. Blynk

A particular IoT platform called Blynk offers a wide range of capabilities for data storage and visualization, sensor data collection and presentation, and remote device control.

There are three main parts to it:

- The Blynk application that lets you use different widgets to build a unique interface.
- The Blynk Server, which can be used locally or through the Blynk cloud, controls communication between gadgets and cell phones.
- The Blynk library allows for the administration of incoming and outgoing commands as well as communication with servers and it is compatible with a variety of hardware platforms[13**]** **.**

II.17.1. Blynk IOT configuration

After signing up for a Blynk IoT account, navigate to the dashboard to create a new project. Assign a name to the project, select "ESP32" as the device type, and choose "Wi-Fi" as the connection method.

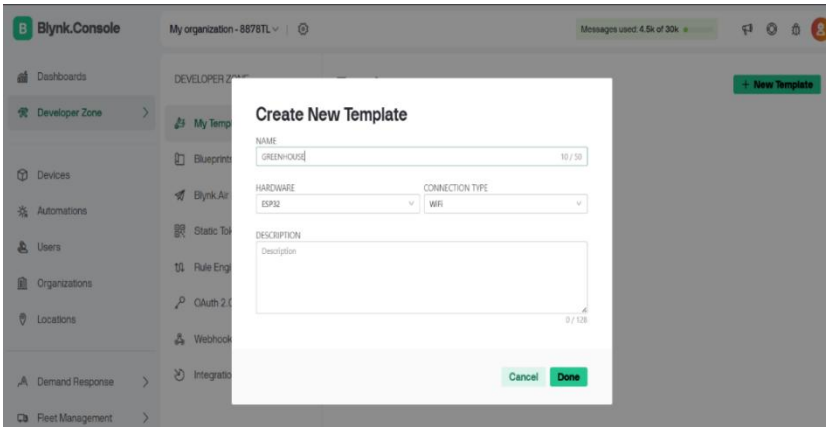


Figure II.22: Create a new Template in Blynk

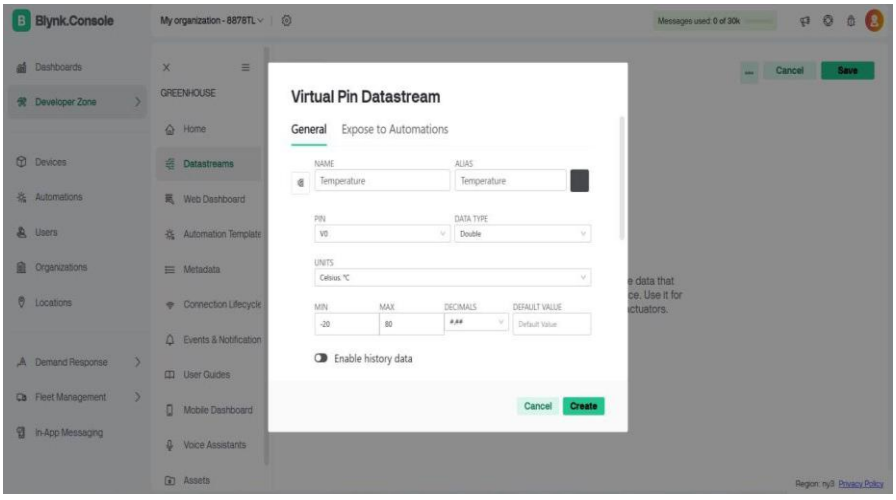


Figure II.23: Select Virtual Pin.

Virtual Pins (V0, V1, V2 …) in Blynk are software-based data channels used to send and receive data between your hardware (ESP32) and the Blynk Cloud. Unlike physical GPIO pins, Virtual Pins are flexible and used for:

- Sending sensor data.
- Controlling devices.
- Handling custom logic.

Steps to Configure Events:

1. Go to Blynk. Cloud → Events → Create New Event.
2. Define event names (should match your code):
    - fire_alert
    - dry_soil
    - high_temp
    - bad_air
    - low_water
3. For each event:
    - Set Event Name (e.g., fire_alert).
    - Add a Notification Action

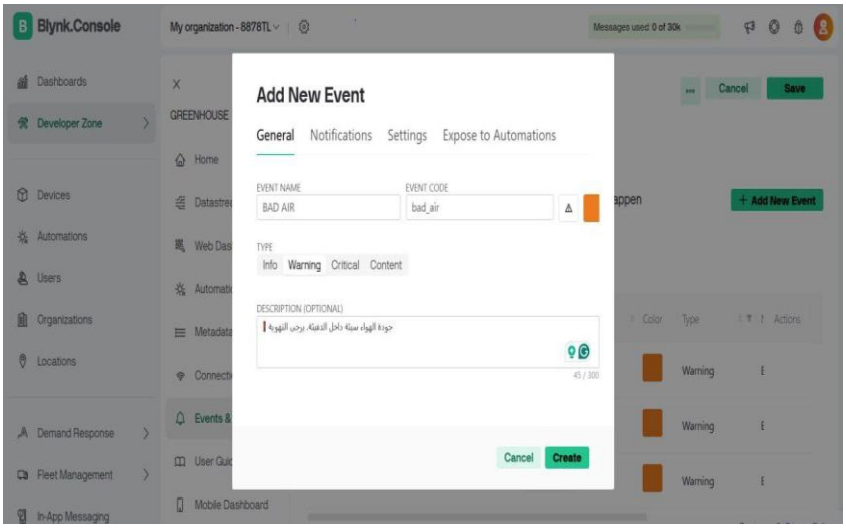


Figure II.24: Setting Up Notifications in Blynk.

II.18.Web Dashboard configuration

From the Tools menu, you can add button and gauge widgets to the interface from the Tools menu. Name them as you wish and select the default pin V0. Next, change the values from 1 to 0, and change the mode when the gauge switch changes input values from 0 to 100, or whatever suits your sensors. Finally, customize these widgets as you wish.

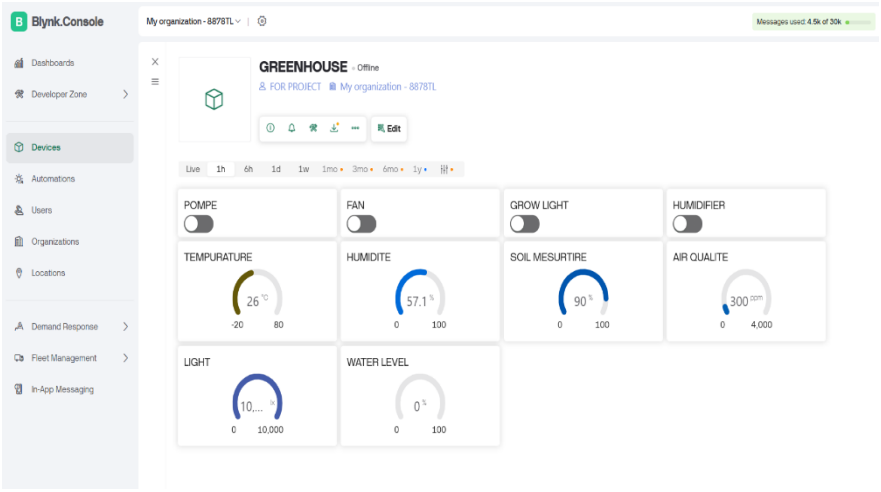


Figure II.25: Web Dashboard.

II.18. 1.Firmware configuration

In device info you find your Template ID, Template Name, and AuthToken, copy and paste into the code.

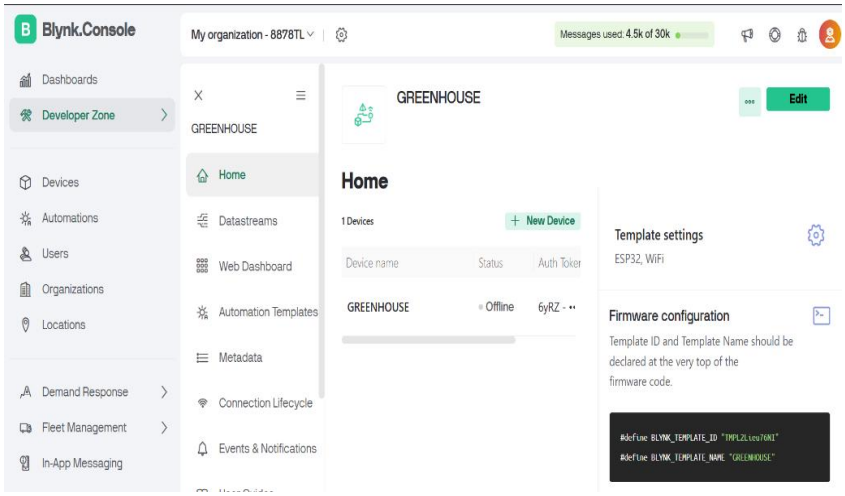


Figure II.26: Firmware configuration.

Conclusion

The most crucial parts we used for the greenhouse project were introduced in this chapter, including the ESP32 controller, a number of sensors (including the DHT22,

LDR, water level, and MQ135 gas sensors), and environmental control devices (such as the pump, fan, and humidifier). We also described the IDE software, the tools and platforms we used, including Cirkit Designer for circuit design and simulation prior to practical implementation, the ThingSpeak platform for data presentation and analysis, and the Blynk application for phone-based remote control.

Chapter III

Experimental Study

III.1. Introduction

This chapter is crucial because it makes it possible to move from the project's conception to its completion through a number of sequential phases. First, the first simulation was used with the aid of the Cirkit designer software. This has enabled us to assess the system's performance and forecast the project's behavior following implementation. The Arduinio IDE may then be used to easily program our ESP 32 microcontroller. The next phase is about practice, but only once the first phase's results have been validated. We then investigate various montages and use the findings to support the project's practicality and accuracy. Additionally, we used Blynk to create an Android application that shows parameters like temperature, humidity, light level, water level and moisture soil. This enables remote surveillance of the greenhouse and its condition. We also used Thing Speak platform to show the graphs of the evolution with time of temperature, humidity, soil moisture, light and water level.

III.2. Working principle of the smart IoT greenhouse

The system incorporates several sensors and actuators to monitor and control the greenhouse:

- Ventilation and lighting: a fan is used to circulate air and evacuate gases, an LDR sensor measures the light intensity, and lighting can also be controlled.
- Pump control: the water pump is managed through a 5V relay.
- Soil moisture: a soil moisture sensor is used to determine the amount of moisture in the soil.
- Water level: a water level sensor is used to monitor the amount of water inside the basin.
- Air temperature and humidity: The DHT22 sensor is used to track air temperature and humidity.
- Air quality and safety: an air quality sensor monitors the air, while a flame detector provides fire safety. If a gas is detected the ventilator turns ON. If a fire is detected, an SMS notification is sent to the user’s smartphone.

When the system is powered on, the ESP32 board connects to the ThinkSpeak platform and the Blynk application via the internet and the ThingSpeak and Blynk cloud. The following features are available:

- Real-time monitoring: the Blynk application displays real-time data, including temperature, water level, humidity, soil moisture, light level, and air quality.

- Automatic irrigation: if the soil moisture drops, the pump is automatically activated to irrigate the plants. However, if the water level in the basin falls, the pump will not start. Once soil is sufficiently moist, the pump turns off.
- Manual controls: The blynk application provides buttons to manually turn the pump, ventilators, and lights on or off.
- Remote access: all operations and monitoring can be performed remotely from anywhere via the Blynk application.
- Data visualization: ThingSpeak is used to display variations in temperature, humidity, water level, light level, and soil moisture over time.

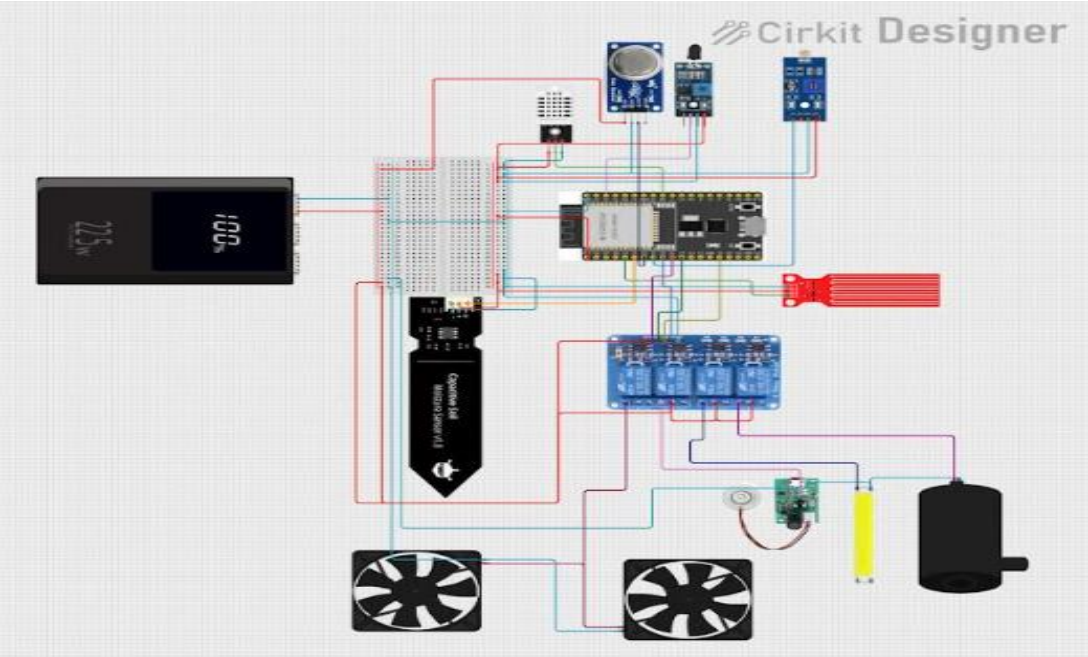


Figure III.1: Circuit diagram of the system in Cirkit.

III.3 Soil moisture sensor with ESP32

III.3.1 Interface soil moisture sensor with ESP32

We must connect the VCC pin to 5V, the GND pin to ground, and the analog output pin to one of the ESP32's GPIO pin 35. Depending on its 12-bit ADC resolution, the ESP32 interprets the sensor's analog voltage, which changes based on the soil moisture content in the range of the ADC values from 0 (dry soil) to 4095 (wet soil).

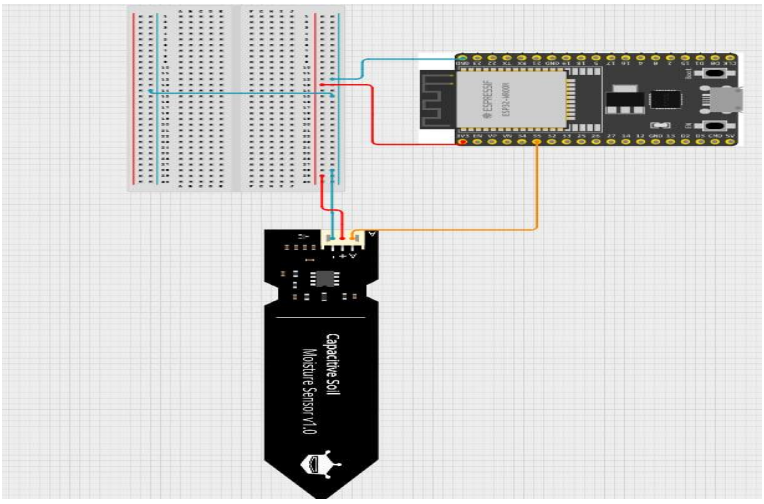


Figure III.2: Soil moisture wiring with ESP32

III.3.2.Soil moisture sensor code

Since the sensor has a distinct reading in both wet and dry conditions, we must calibrate this sensor before we can utilize it correctly. The code is in the following in both completely dry and wet conditions in order to calibrate the sensor:

ESP32-WROOM-DA Module Verify

sketch_may23c.ino

```
#define BLYNK_TEMPLATE_ID ""
#define BLYNK_TEMPLATE_NAME "SMARTGREENHOUSE"
#define BLYNK_AUTH_TOKEN ""
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <ThingSpeak.h>
char ssid[] = "";
char pass[] = "";
WiFiClient client;
unsigned long myChannelNumber = ;
const char* myWriteAPIKey = "";
#define soilPin 35
float soilMoisturePercent;

void readSoil() {
  int soilValue = analogRead(soilPin);
  soilMoisturePercent = map(soilValue, 2660, 1040, 0, 100);
  soilMoisturePercent = constrain(soilMoisturePercent, 0, 100);
}
```

Figure III.3: Soil moisture calibration Code.

The analog value from a soil moisture sensor is read by this code, which then turns it into a percentage that shows the soil moisture level. Reading the analog data from the sensor and

storing it in the variable soil Moisture Value. This value, which falls from 2660 (dry) and 1040 (wet), is then transformed into a percentage scale from 0% (totally dry) to 100% (entirely wet) using the map function. The outcome is kept as a float in the soil Moisture Percent variable. The instructions soil Moisture Percent = constrain (soil Moisture Percent, 0, 100) make the percentage remaining within the acceptable range of 0 to 100%, avoiding any values that are out of the range due to noise or erratic readings.

III.4. DHT22 sensor with ESP32

III.4.1. Interface DHT22 sensor with ESP32

It is simple to connect the DHT22 sensor to the ESP32 board. The following are the connections:

Attach the DHT22's VCC pin to the ESP32's 3.3V pin.

Attach the DHT22's DATA pin to the ESP32's GPIO pin 4.

Attach the DHT22's GND pin to one of the ESP32's GND pins.

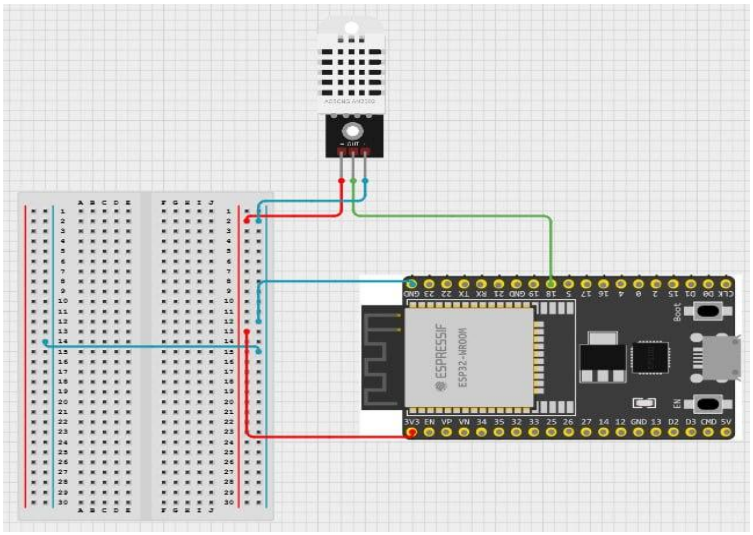


Figure III.4: DHT22 wiring with ESP32.

III.4.2. DHT22 sensor code

This code continually receives data from the DHT22 sensor and shows the current temperature and humidity readings on the serial monitor.

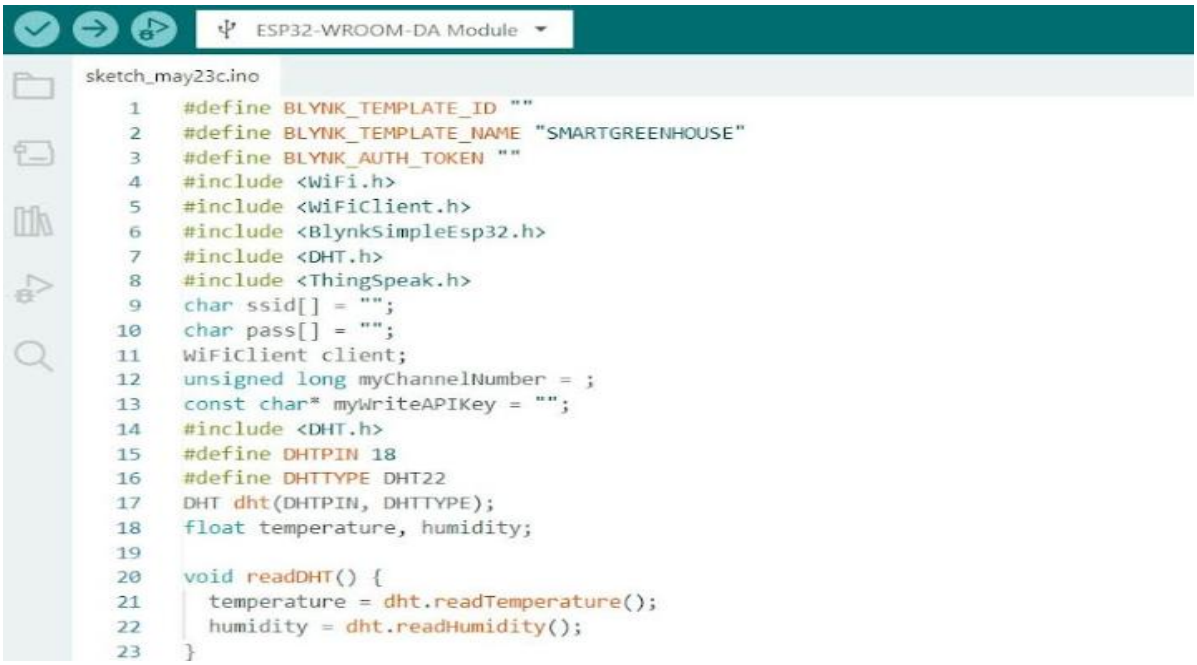


Figure III.5: Code to read from DHT22 sensor.

The #define DHTPIN 18 line indicates that the ESP32's GPIO 18 is connected to the DHT22's data pin. The #define DHTTYPE DHT22 line specifies the kind of sensor being utilized (DHT22). The DHT dht(DHTPIN, DHTTYPE) creates a DHT sensor object using the defined pin and sensor type. While the dht.readHumidity() function reads the relative humidity and stores it in the humidity variable, the dht.readTemperature() function receives the current temperature from the sensor and stores it in the temperature variable.

III.5. Humidifier connection

III.5.1. Interface Humidifier connection

The humidifier is controlled by one of the relays. If the humidity sensor readings are low, the ESP32 sends a signal to the relay, which then powers the humidifier.

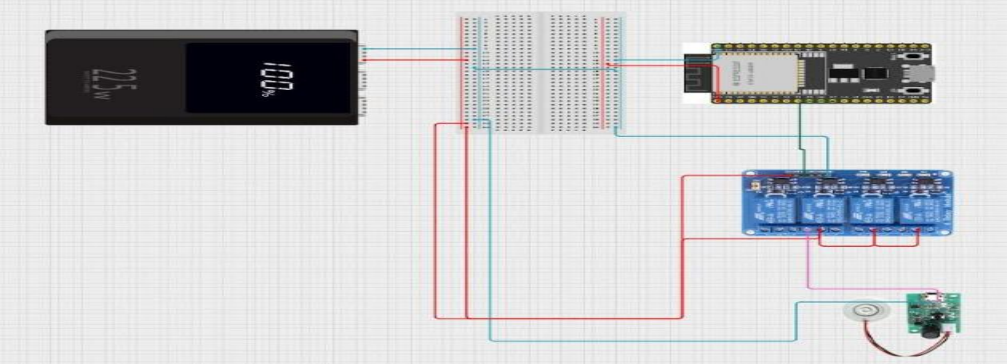


Figure III.6: Humidifier, relay and ESP32 connections.

III.5.2. Humidifier code

ESP32-WROOM-DA Module

sketch_may23c.ino

```
#define BLYNK_TEMPLATE_ID ""
#define BLYNK_TEMPLATE_NAME "SMARTGREENHOUSE"
#define BLYNK_AUTH_TOKEN ""
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <ThingSpeak.h>
char ssid[] = "";
char pass[] = "";
WiFiClient client;
unsigned long myChannelNumber = ;
const char* myWriteAPIKey = "";
#define humidifierRelay 27
bool humidifierManual;

void controlHumidifier() {
  if (systemManual) {
    digitalWrite(humidifierRelay, humidifierManual ? LOW : HIGH);
  } else {
    digitalWrite(humidifierRelay, (humidity < 40) ? LOW : HIGH);
  }
}
```

Figure III.7 : Humidifier control code.

definehumidifierRelay 27 line sets GPIO 27 connected to the pin of the relay controlling the humidifier. Bool humidifierManual is a Boolean variable used to store the manual ON/OFF state of the humidifier. The controlHumidifier() function controls the humidifier relay based on the system mode:

- Manual mode (systemManual is true):

  The relay is turned ON or OFF depending on the value of humidifierManual. If humidifierManual is true, the relay is set to LOW (ON); if false, it is set to HIGH (OFF).

- Automatic mode (systemManual is false):

The relay is controlled by the humidity reading. If the humidity is below 40%, the relay is set to LOW (turning the humidifier ON). If the humidity is 40% or higher, the relay is set to HIGH (turning the humidifier OFF)

III.6. Air quality sensor MQ135 with ESP32

III.6.1. Interface air quality sensor MQ135 with ESP32

The MQ135 sensor is used to measure the concentrations of dangerous gases such as ammonia (NH_3) and carbon dioxide (CO_2) as well as to detect overall air quality. The analog output of the MQ135 is connected to the GPIO pin of the ESP32. When powered on, the MQ135 sensor detects the concentration of certain gases in the air and outputs an analog voltage. The ESP32 reads this analog value. A larger value denotes lower air quality (a higher concentration of dangerous gases). If it surpasses a predetermined threshold that is established following calibration, a ventilation mechanism, such a fan, can be turned ON.

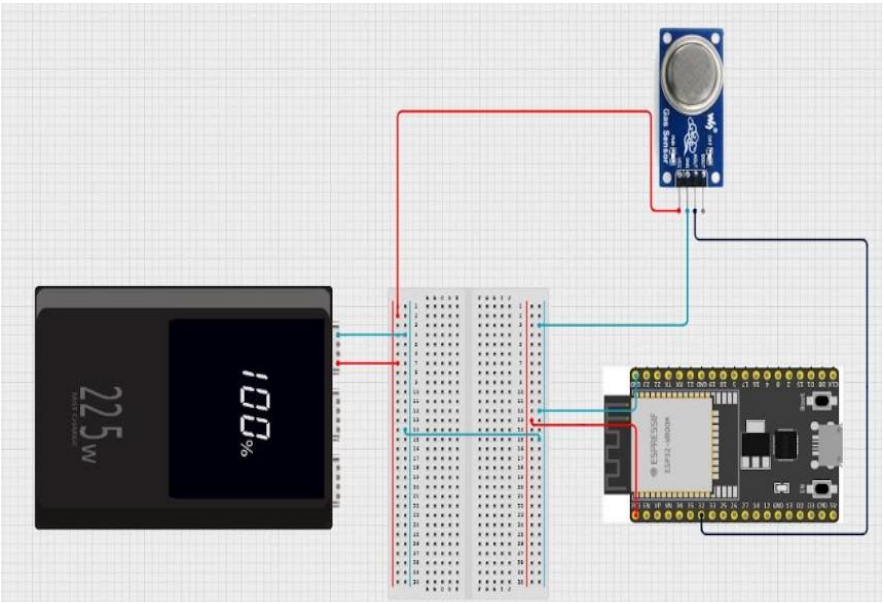


Figure III.8: MQ 135 wiring with ESP32.

III.6.2. Air quality sensor MQ135 Code

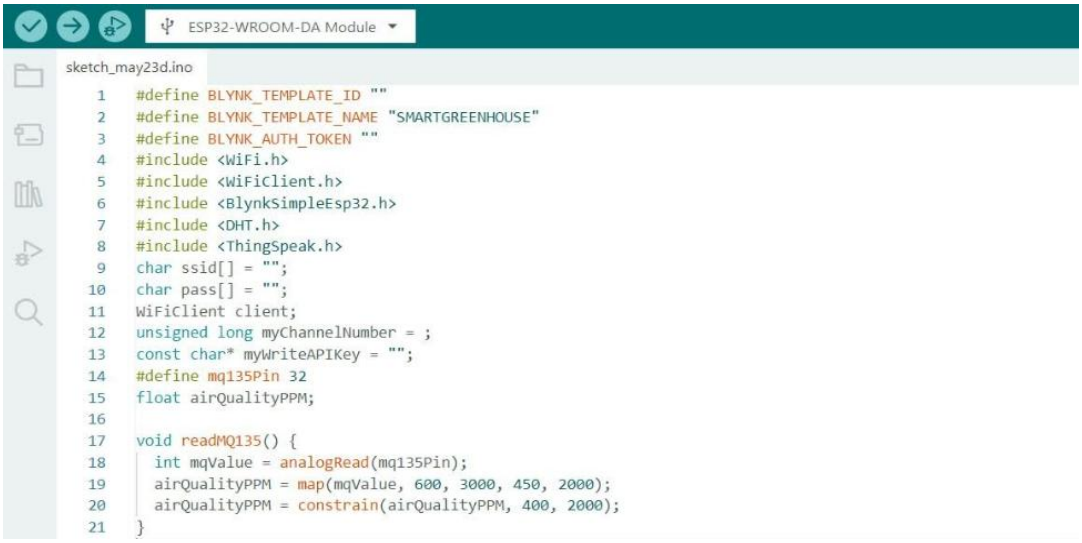


Figure III.9: Code to read Air quality from sensor MQ135.

An analog input pin for the MQ135 sensor is defined as mq135Pin is connected to the the GPIO 32 pin of the ESP32. A variable airQualityPPM is declared to store the air quality value in parts per million (PPM). The function readMQ135() reads the analog value from the MQ135 sensor using analogRead(mq135Pin).The map() function, mapping values from 600-3000 (sensor output range) to 400-2000 (PPM range). The constrain() function is used to ensure the PPM value stays between 400 and 2000.

III.7. Light sensor LDR

III.7.1. Interface light sensor LDR

The light intensity within the greenhouse is measured by the LDR sensor. The sensor is based on the idea that its resistance varies with light intensity, declining with increasing light intensity and rising with decreasing light intensity.

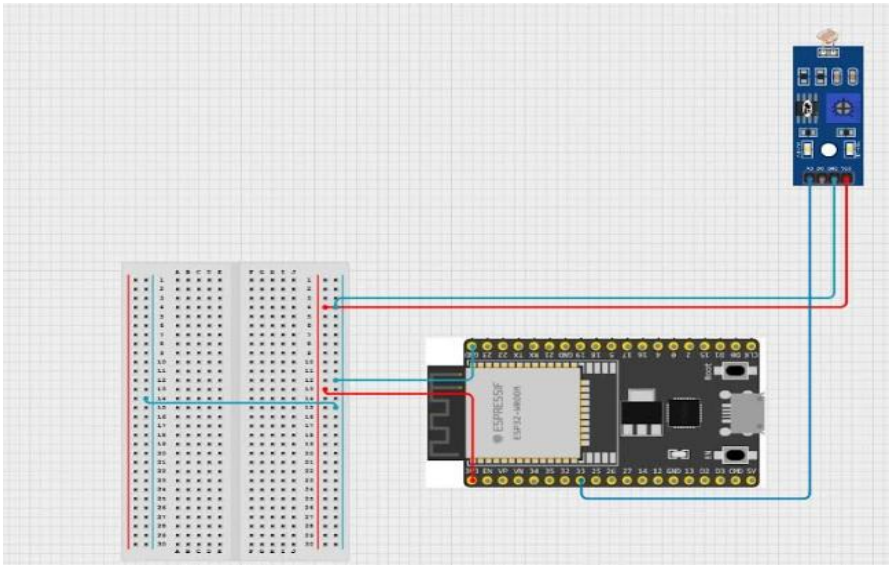


Figure III.10: LDR wiring with ESP32.

III.7.2.Light sensor LDR code

The code below controls the light sensor.

The ldrPin is connected to the GPIO 33 pin of the ESP32. lightLux is a float variable to store the light level in lux. void readLDR() function reads the light sensor and converts the value. analogRead(ldrPin) reads the analog value from the LDR pin (range:0-4095). Map() converts the raw value to a range of 0 to 10,000 lux.

ESP32-WROOM-DA Module

sketch_may23c.ino

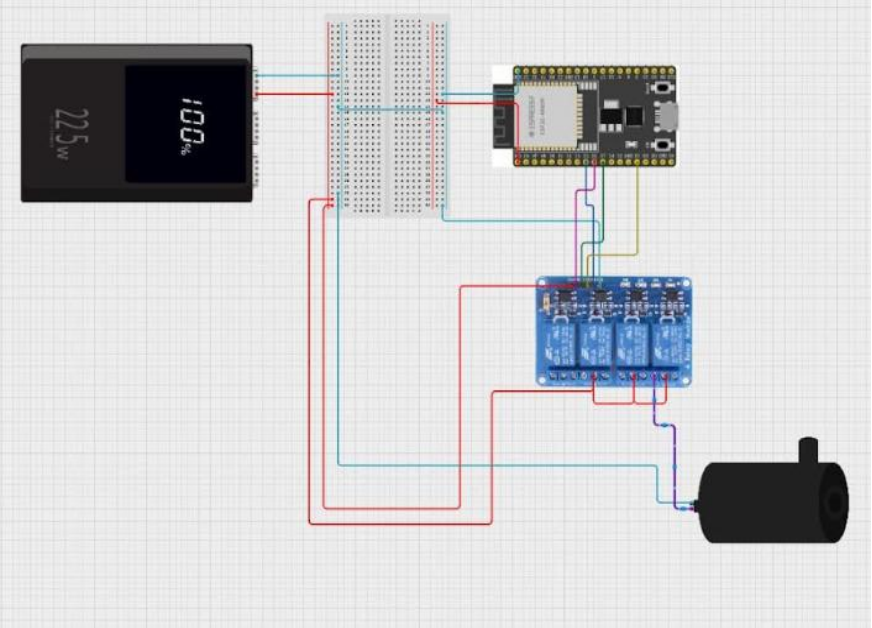
```
#define BLYNK_TEMPLATE_ID ""
#define BLYNK_TEMPLATE_NAME "SMARTGREENHOUSE"
#define BLYNK_AUTH_TOKEN ""
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <ThingSpeak.h>
char ssid[] = "";
char pass[] = "";
WiFiClient client;
unsigned long myChannelNumber = ;
const char* myWriteAPIKey = "";
#define ldrPin 33
float lightLux;

void readLDR() {
  int ldrValue = analogRead(ldrPin);
  lightLux = map(ldrValue, 4095, 0, 0, 10000);
}
```

Figure III.11: code to read from light sensor LDR.

III.8. Water Pump Control System using ESP32

III.8.1. Interface Water Pump Control System using ESP32

The relay module receives a signal from the ESP32 and controls the water pump's power supply to turn it ON or OFF. This enables manual or automatic pump control.



FigureIII.12: Water Pump wiring with ESP32.

III.8.2. Water Pump code

ESP32-WROOM-DA Module

sketch_may23c.ino

```
#define BLYNK_TEMPLATE_ID ""
#define BLYNK_TEMPLATE_NAME "SMARTGREENHOUSE"
#define BLYNK_AUTH_TOKEN ""
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <ThingSpeak.h>
char ssid[] = "";
char pass[] = "";
WiFiClient client;
unsigned long myChannelNumber = ;
const char* myWriteAPIKey = "";
#define pumpRelay 25
bool pumpManual, systemManual;

void controlPump() {
  if (waterLevelPercent < 20) {
    digitalWrite(pumpRelay, HIGH);
    return;
  }

  if (systemManual) {
    digitalWrite(pumpRelay, pumpManual ? LOW : HIGH);
  } else {
    digitalWrite(pumpRelay, (soilMoisturePercent < 30) ? LOW : HIGH);
  }
}
```

Figure III.13: code to control Water Pump.

pumpRelay is set to GPIO 25 of ESP32, which controls the water pump relay. controlPump() function decides when to turn the water pump on or off:

if the water level percentage is less than 20%, the pump is turned off by setting the relay pin HIGH, and the function exits.

If systemManual is true (manual mode), the pump is controlled directly by the pumpManual variable.

If not in manual mode, the pump is turned on if the soil moisture percentage (soilMoisturePercent) is less than 30%.

III.9. LED Light

III.9.1. Interface LED Light

The ESP32 reads data from the LDR to measure ambient light intensity inside the greenhouse.

Automated lighting: if the light intensity drops below a certain threshold, the ESP turns ON the LED light. When the natural light is sufficient, the LED light is turned OFF. The ESP32 sends signals to relay to turn the LED light ON or OFF.

Manual lighting: Through Blynk application, users can control the LED light remotely using a smartphone.

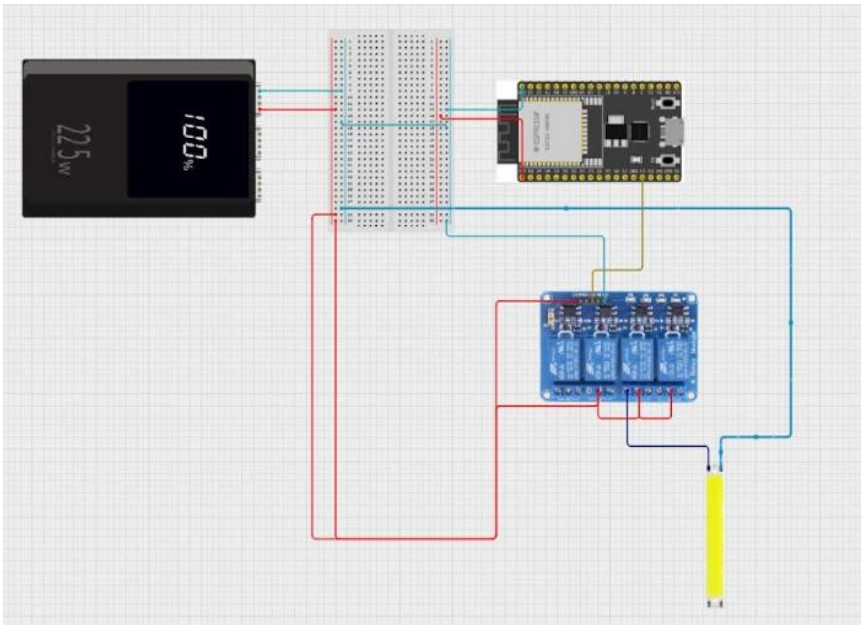


Figure III.14: LED Light wiring with ESP32

III.9.2. LED Light Code

ESP32-WROOM-DA Module

sketch_may23c.ino

```
#define BLYNK_TEMPLATE_ID ""
#define BLYNK_TEMPLATE_NAME "SMARTGREENHOUSE"
#define BLYNK_AUTH_TOKEN ""
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <ThingSpeak.h>
char ssid[] = "";
char pass[] = "";
WiFiClient client;
unsigned long myChannelNumber = ;
const char* myWriteAPIKey = "";
#define lightRelay 13
bool lightManual;

void controlLight() {
  if (systemManual) {
    digitalWrite(lightRelay, lightManual ? LOW : HIGH);
  } else {
    digitalWrite(lightRelay, (lightLux < 3000) ? LOW : HIGH);
  }
}
```

Figure III.15: code to control LED Light.

#define lightRelay 13 sets GPO 13 of ESP32 connected to the lightrelay. controlLight() function decides when to turn the LED light ON or OFF.

If systemManual is true (manual mode), the relay turns ON; if false, it turns OFF.

If not in manual mode, the system checks the measured light intensity (lightLux). If it is below 3000 lux, the relay turn ON the LED light; if above 3000, the light is turned OFF.

III.10. Water level sensor

III.10.1. Interface Water level sensor

The water level sensor determines if water is present or not in the basin. Depending on whether or not water is detected, it gives the ESP32 a digital HIGH or LOW signal. After that, the ESP32 can turn OFF the pump or turn ON the pump.

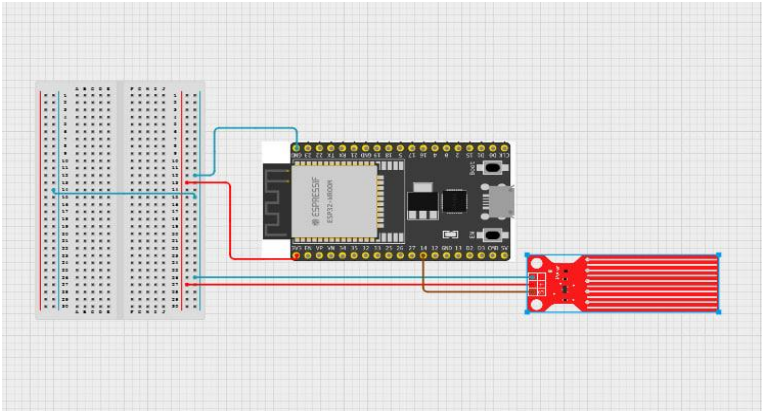
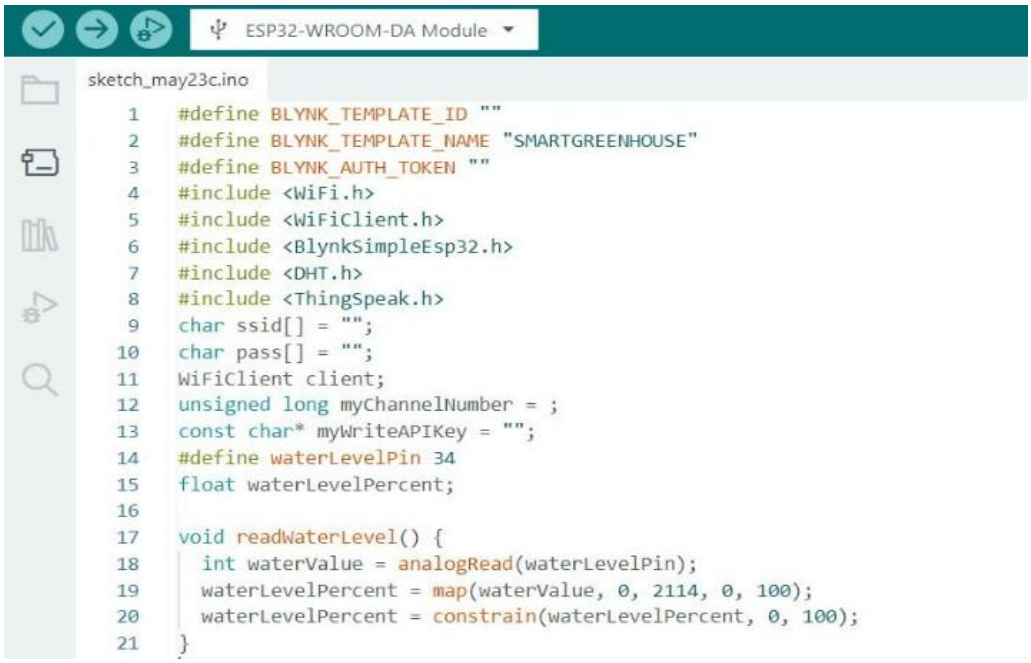


Figure III.16: Water level sensor wiring with ESP32.

III.10.2. Water level sensor code



FigureIII.17: code to read from Water level sensor.

#define waterLevelpin 34 assigns GPIO pin 34 to the water level sensor.readWaterLevel() function reads the analog value from the water level sensor connected to GPIO 34 pin of ESP32. The raw sensor value (from o to 2114) is mapped to a percentage scale (0 to 100%). The constrain function ensures the result stays within 0-100%.

III.11. Flame sensor

III.11.1. Interface Flame sensor

A flame sensor picks up radiation from a flame, including ionization signals, ultraviolet (UV), and infrared (IR) radiation. To guarantee safe operation, it recognizes the existence of a flame and notifies the control system.

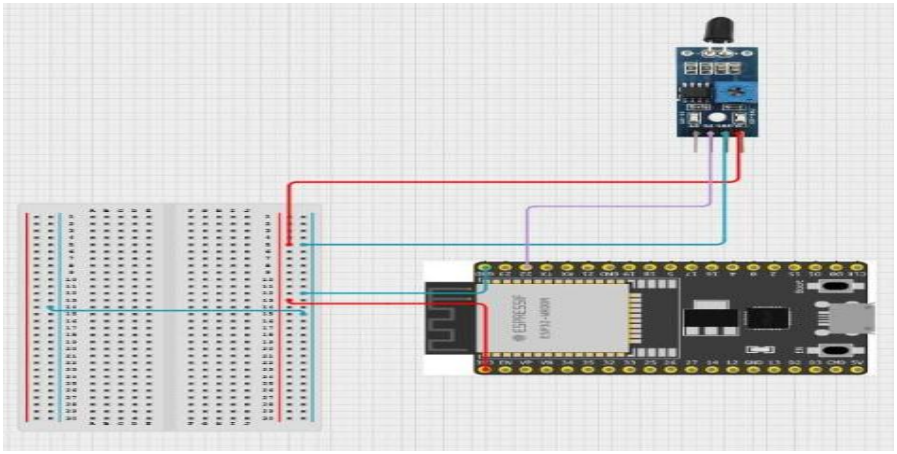


Figure III.18: flame sensor wiring with ESP32.

III.11.2. Flame sensor Code

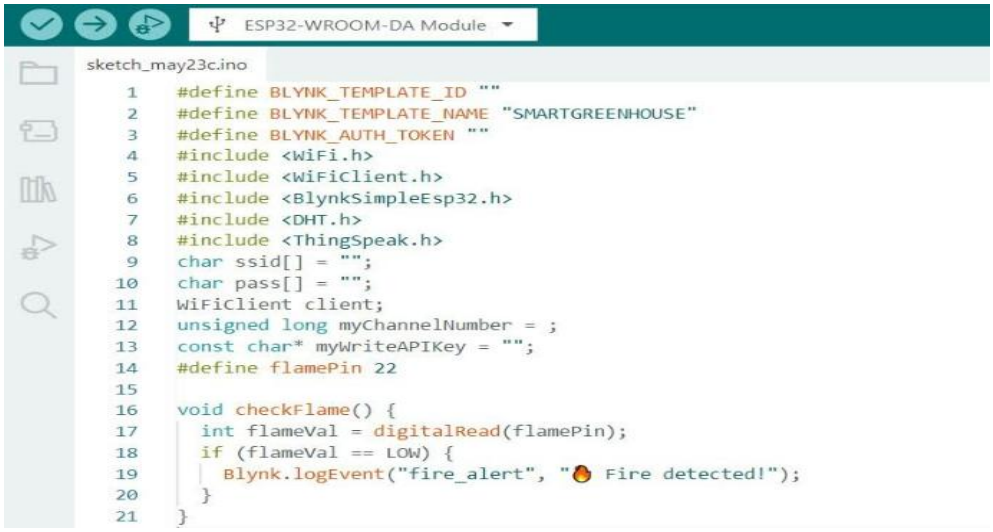


Figure III.19: code to read the flame sensor.

#define flamPin 22 sets GPO 22 pin of ESP32 connected to the flame sensor. checkFlame() reads the digital value from the flame sensor. If the sensor output LOW, it means a flme (fire) is detected. When the flame is present, the code sends a fire alert event to the Blynk platform with the message “fire detected”.

III.12. Fan Control

III.12.1. Interface Fan Control

The ESP 32 monitors environmental conditions and activates fans as needed. The relay ensures the switching of the fans.

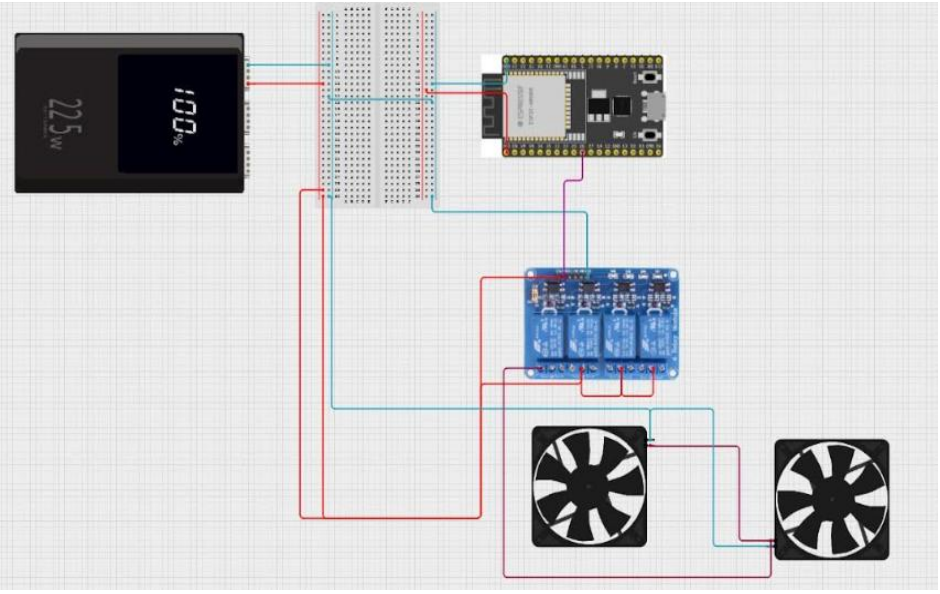


Figure III.20: Fan wiring with ESP32.

III.12.2. Fan Control code

ESP32-WROOM-DA Module

sketch_may23c.ino

```
#define BLYNK_TEMPLATE_ID ""
#define BLYNK_TEMPLATE_NAME "SMARTGREENHOUSE"
#define BLYNK_AUTH_TOKEN ""
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <ThingSpeak.h>
char ssid[] = "";
char pass[] = "";
WiFiClient client;
unsigned long myChannelNumber = ;
const char* myWriteAPIKey = "";
#define fanRelay 26
bool fanManual;

void controlFan() {
  if (systemManual) {
    digitalWrite(fanRelay, fanManual ? LOW : HIGH);
  } else {
    digitalWrite(fanRelay, (temperature > 30 || airQualityPPM > 600) ? LOW : HIGH);
  }
}
```

Figure III.21: Code to control the Fan.

#define fanRelay 26 sets GPIO pin 26 connected to the fan relay. Bool fanManual; variable tracks manual control state.

Manual mode:

If systemManual is true, the fan is controlled directly by the fanManual variable (via Blynk application). If fanManual is true, the relay is set to LOW (turning the fan ON). If fanManual is false, the relay is set to HIGH (turning the fan OFF).

Automatic mode:

If systemManual is false, the fan is controlled automatically base on environmental conditions:

The fan turns ON (relay set to LOW) if the temperature is above 30º or air quality is above 600.

Otherwise, the fan is OFF (relay set to HIGH).

```
  lightLux = map(ldrValue, 4095, 0, 0, 10000);

  // Read water level
  int waterValue = analogRead(waterLevelPin);
  waterLevelPercent = map(waterValue, 0, 2114, 0, 100);
  waterLevelPercent = constrain(waterLevelPercent, 0, 100);

  // Read soil moisture
  int soilValue = analogRead(soilPin);
  soilMoisturePercent = map(soilValue, 2660, 1040, 0, 100);
  soilMoisturePercent = constrain(soilMoisturePercent, 0, 100);

  // Send data to Blynk
  Blynk.virtualWrite(V0, temperature);
  Blynk.virtualWrite(V1, humidity);
  Blynk.virtualWrite(V2, soilMoisturePercent);
  Blynk.virtualWrite(V3, airQualityPPM);
  Blynk.virtualWrite(V4, lightLux);
  Blynk.virtualWrite(V6, waterLevelPercent);

  // Print to Serial Monitor
  Serial.print("Temp: "); Serial.print(temperature); Serial.print("°C | ");
  Serial.print("Hum: "); Serial.print(humidity); Serial.print("% | ");
  Serial.print("Soil: "); Serial.print(soilMoisturePercent); Serial.print("% | ");
  Serial.print("Air: "); Serial.print(airQualityPPM); Serial.print(" ppm | ");
  Serial.print("Light: "); Serial.print(lightLux); Serial.print(" Lux | ");
  Serial.print("Water: "); Serial.print(waterLevelPercent); Serial.println("%");
}
```

ESP32-WROOM-DA Module

sketch_may26a.ino

```
#define BLYNK_TEMPLATE_ID "TemplateID"
#define BLYNK_TEMPLATE_NAME "ProjectName"
#define BLYNK_AUTH_TOKEN "AuthToken"
#include <DHT.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
char ssid[] = "SSID";
char pass[] = "Password";
#define DHTPIN 18
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
#define SOIL_PIN 35
#define MQ135_PIN 19
#define LDR_PIN 21
#define LEVEL_PIN 14
#define FLAME_PIN 22
const int SOIL_WET = 1040;
const int SOIL_DRY = 2660;
const int AIR_CLEAN = 200;
const int AIR_DIRTY = 1000;
bool flameNotified = false;
bool levelNotified = false;

void readSensorsAndSendToBlynk() {
  // Read temperature and humidity
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();

  // Read air quality (MQ135)
  int mqValue = analogRead(mq135Pin);
  airQualityPPM = map(mqValue, 600, 3000, 450, 2000);
  airQualityPPM = constrain(airQualityPPM, 400, 2000);

  // Read light intensity (LDR)
  int ldrValue = analogRead(ldrPin);
```

Figure III.22-23: Code to send Data to Blynk and serial monitor.

ESP32-WROOM-DA Module

sketch_may26a.ino

```
#define BLYNK_TEMPLATE_ID ""
#define BLYNK_TEMPLATE_NAME "SMARTGREENHOUSE"
#define BLYNK_AUTH_TOKEN ""

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <ThingSpeak.h>

char ssid[] = "";
char pass[] = "";
WiFiClient client;

unsigned long myChannelNumber = 2962289;
const char* myWriteAPIKey = "XRGJJP0D8652OQGM";

void sendToThingSpeak() {
  ThingSpeak.setField(1, temperature);
  ThingSpeak.setField(2, humidity);
  ThingSpeak.setField(3, soilMoisturePercent);
  ThingSpeak.setField(4, airQualityPPM);
  ThingSpeak.setField(5, lightLux);
  ThingSpeak.setField(6, waterLevelPercent);

  int status = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

  if(status == 200) {
    Serial.println("Data sent to ThingSpeak successfully");
  } else {
    Serial.println("Problem sending to ThingSpeak. HTTP error code " + String(status));
  }
}
```

Figure III.24: Code to send Data to ThingSpeak platform.

III.13. Results

Output Serial Monitor X

Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM6') New Line 115200 baud

```
Temp: 30.00°C | Hum: 52.60% | Soil: 89.00% | Air: 504.00 ppm | Light: 8720.00 Lux | Water: 69.00%
Temp: 30.00°C | Hum: 52.40% | Soil: 90.00% | Air: 501.00 ppm | Light: 8715.00 Lux | Water: 69.00%
Temp: 30.00°C | Hum: 52.30% | Soil: 89.00% | Air: 501.00 ppm | Light: 8710.00 Lux | Water: 69.00%
Temp: 30.00°C | Hum: 52.30% | Soil: 90.00% | Air: 499.00 ppm | Light: 8676.00 Lux | Water: 69.00%
Temp: 30.00°C | Hum: 52.40% | Soil: 90.00% | Air: 514.00 ppm | Light: 8669.00 Lux | Water: 70.00%
Temp: 30.00°C | Hum: 52.60% | Soil: 90.00% | Air: 497.00 ppm | Light: 8673.00 Lux | Water: 70.00%
Temp: 30.00°C | Hum: 52.70% | Soil: 90.00% | Air: 499.00 ppm | Light: 8673.00 Lux | Water: 69.00%
Temp: 30.00°C | Hum: 52.80% | Soil: 90.00% | Air: 517.00 ppm | Light: 8703.00 Lux | Water: 69.00%
Temp: 30.10°C | Hum: 52.80% | Soil: 90.00% | Air: 501.00 ppm | Light: 8710.00 Lux | Water: 70.00%
Temp: 30.00°C | Hum: 52.80% | Soil: 90.00% | Air: 457.00 ppm | Light: 8752.00 Lux | Water: 70.00%
Temp: 30.10°C | Hum: 52.90% | Soil: 90.00% | Air: 491.00 ppm | Light: 8246.00 Lux | Water: 70.00%
Temp: 30.10°C | Hum: 51.40% | Soil: 90.00% | Air: 452.00 ppm | Light: 8764.00 Lux | Water: 70.00%
Temp: 30.10°C | Hum: 50.50% | Soil: 90.00% | Air: 467.00 ppm | Light: 8752.00 Lux | Water: 70.00%
Temp: 30.10°C | Hum: 49.20% | Soil: 90.00% | Air: 451.00 ppm | Light: 8757.00 Lux | Water: 70.00%
Temp: 30.10°C | Hum: 48.10% | Soil: 90.00% | Air: 445.00 ppm | Light: 8752.00 Lux | Water: 70.00%
Temp: 30.10°C | Hum: 47.40% | Soil: 90.00% | Air: 446.00 ppm | Light: 8791.00 Lux | Water: 70.00%
Temp: 30.10°C | Hum: 46.80% | Soil: 90.00% | Air: 456.00 ppm | Light: 8835.00 Lux | Water: 71.00%
Temp: 30.10°C | Hum: 46.40% | Soil: 90.00% | Air: 447.00 ppm | Light: 8866.00 Lux | Water: 71.00%
Temp: 30.20°C | Hum: 46.00% | Soil: 90.00% | Air: 451.00 ppm | Light: 8891.00 Lux | Water: 71.00%
Temp: 30.20°C | Hum: 45.90% | Soil: 90.00% | Air: 447.00 ppm | Light: 8881.00 Lux | Water: 71.00%
```

Building sketch Ln 172, Col 41 ESP32-WROOM-DA Module on COM6

Figure III.25: Reading the sensor data in the serial monitor.

III.13.1. Greenhouse results captured in ThingSpeak platform

Graph III.1: Data of temperature.

Graph III.2: Data of the humidity.

Graph III. 3: Data of soil moisture.

Graph III. 4: Data of air quality.

Graph III.5: Data of the Intensity of illumination.

III.13.2. Greenhouse results captured in Blynk application

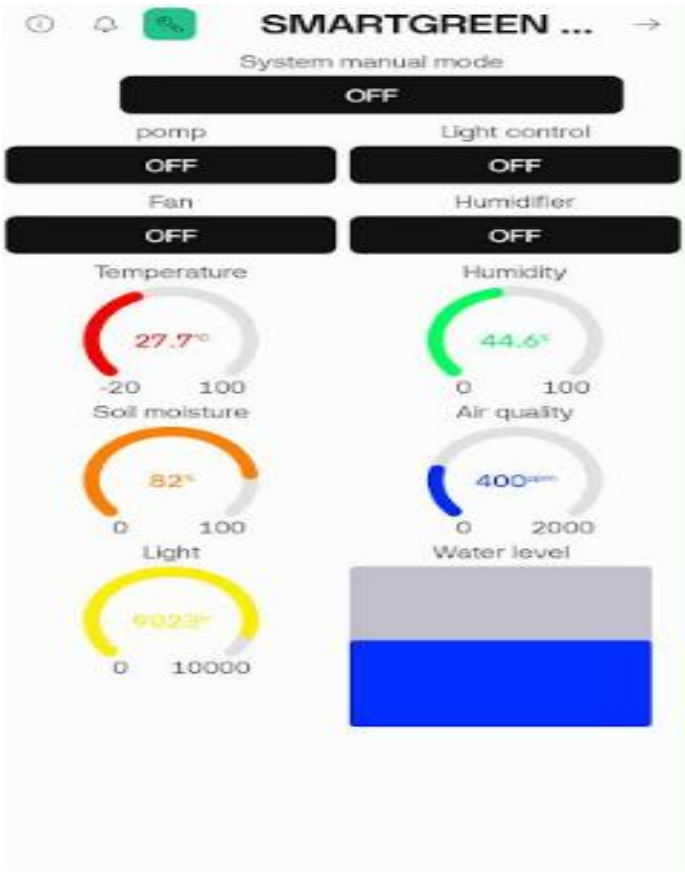


Figure III.26: Greehouse sensor Data in Blynk app.

The efficiency of the smart IoT greenhouse system in tracking environmental elements crucial to plant growth is evident. The indicators display excellent relative humidity (44.6%), good soil moisture (82%), and a moderate temperature (27.7°C). Additionally, the app shows a high light level (9023 lux) and an air quality measurement (400 ppm), both indicate good growing circumstances for plants. However, the manual operation system and all of its services (such as the pump, light control, fan, and humidifier) are currently off, which means that the system is either in monitoring mode or operating automatically according to programs. The basin's high water level is a reliable indicator of the system's water resources.

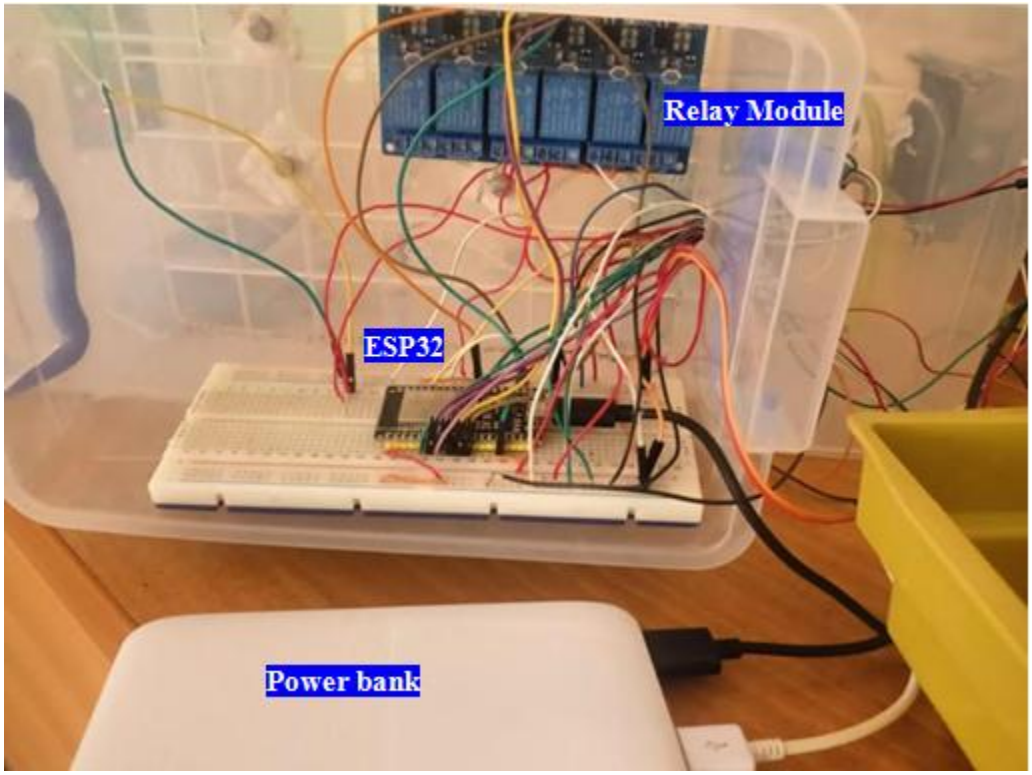


Figure III.27: Prototype view1.

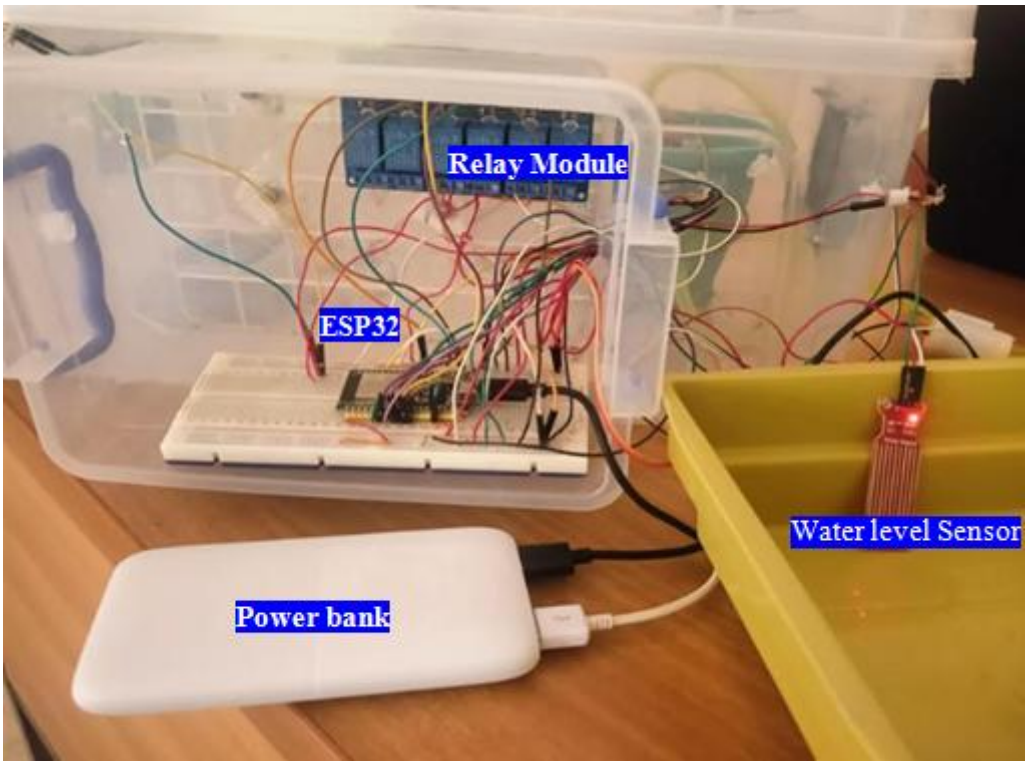


Figure III.28: Prototype view2.

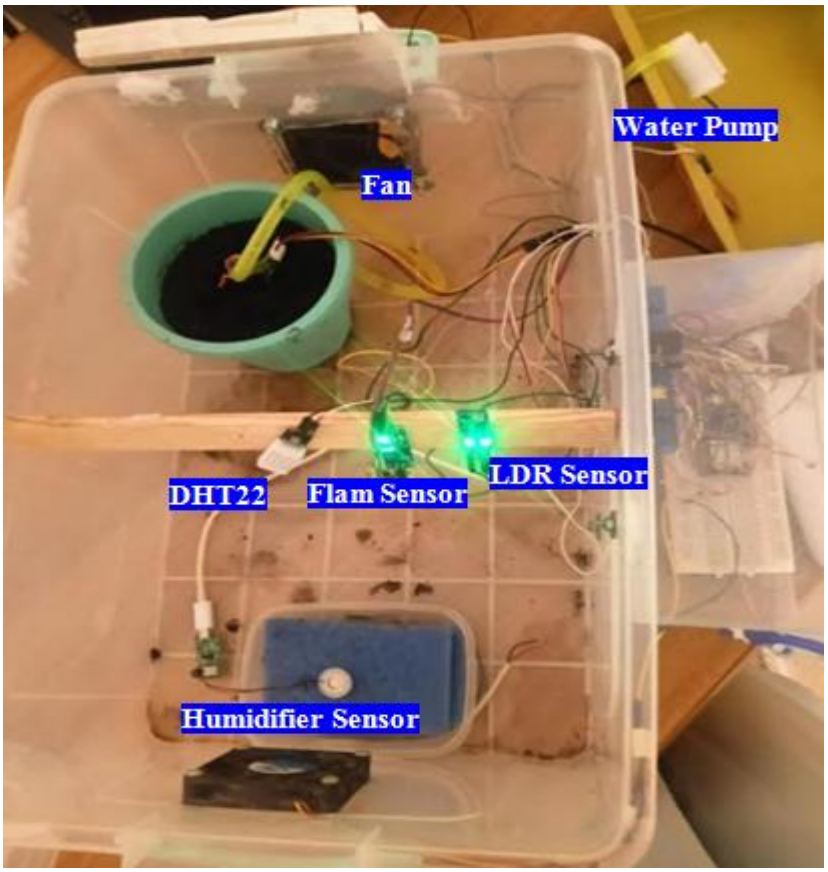


Figure III.29: Prototype view3.

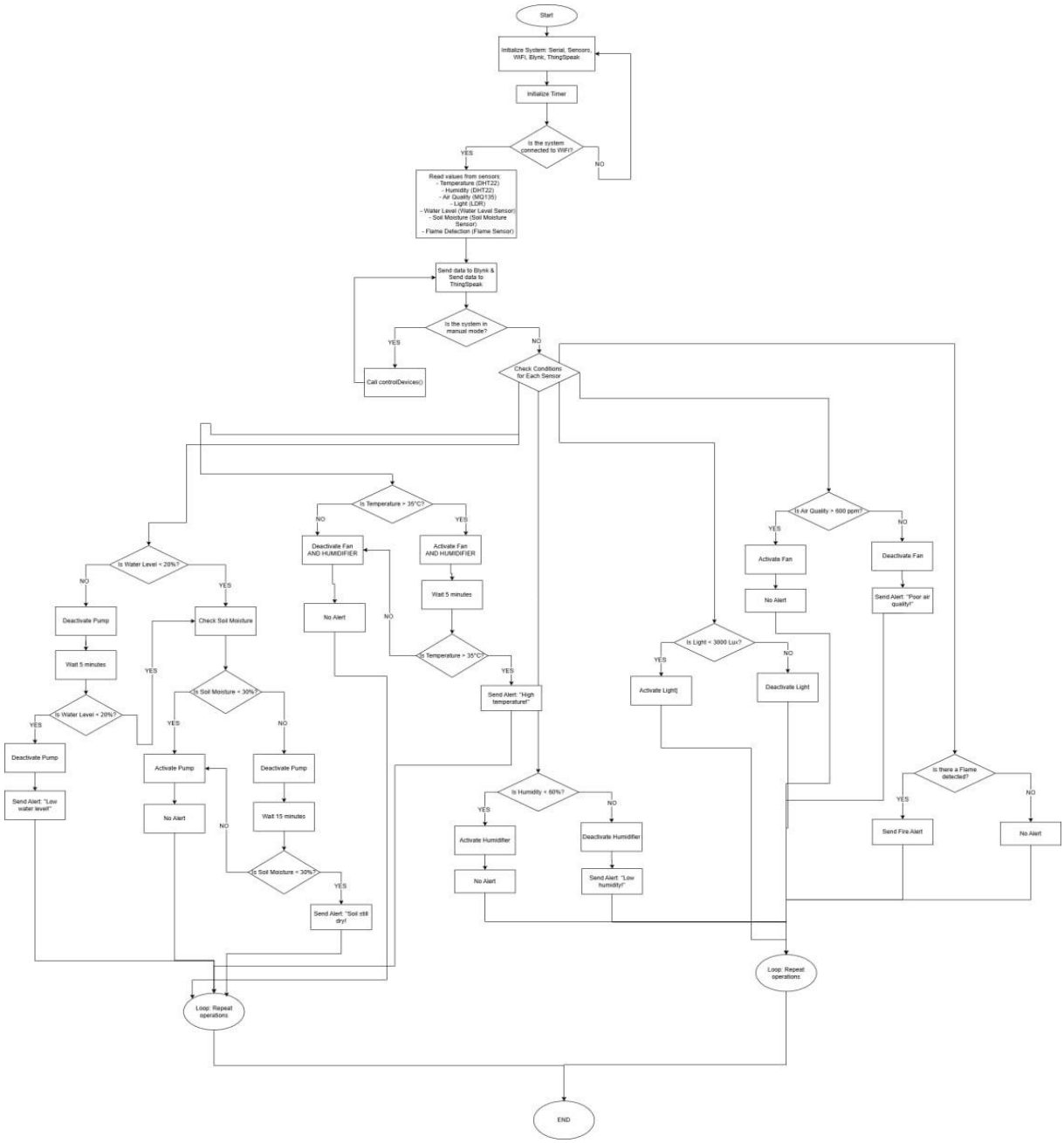


Figure III.30: Diagram of the system

Conclusion

In this chapter, the steps involved in designing and implementing a smart greenhouse system using the EP32 controller are explained, with a detailed explanation of the mechanism for connecting the various sensors and devices, and the method for sending data to the ThingSpeak platform, in addition to remote control using the Blynk application. The software

code used was also presented and its basic functions explained. This chapter forms the practical basis on which the results were analyzed and discussed.

GENERAL CONCLUSION

GENERAL CONCLUSION

At the end of this project, we can say that the experience of working on the smart greenhouse using IoT technologies was important and useful for us. We identified the idea and the issue that the project is trying to solve, where we studied the necessary components, designed the system, wrote the code, and tested the project on the Cirkit Designer simulator, until we reached the implementation of an integrated model that monitors and controls the environmental conditions inside the greenhouse automatically. This work relied on the ESP32 controller, with a set of sensors such as the DHT22 temperature and humidity sensor, soil moisture sensor, air quality sensor, water level sensor in addition to devices such as a fan, pump and humidifier to adjust the conditions inside the greenhouse.

The use of platforms such as ThingSpeak and Blynk is considered the most important features of the project, as the system was linked to these platforms to display data online and the possibility of remote control through a mobile phone or computer, which makes the system more intelligent and flexible.

During this project, we learned a lot, not only from the technical side, such as programming, connecting and designing the system, but also in terms of organizing time, searching for solutions to the issues we faced, and practical thinking in linking technology to agriculture.

References

[1] Gaikwad A, Ghatge A, Kumar H, Mudliar K. Monitoring of Smart Greenhouse. Int Res J Eng Technol. 2016;3(11):573–4.

[2] kherfi A, Zabi A. An automatic watering system for a smart greenhouse [Master's thesis]. El Oued : University Echahid Hamma Lakhdar – El Oued; 2024.

[3] Mehani TE, Mimouni MR. Management of an intelligent greenhouse system [Master's thesis]. Ouargla : University Kasdi Merbah Ouargla, Faculty of Applied Sciences; 2023.

[4] MEGTIT T, DAHMANE D. Réalisation d'une serre agricole intelligente et contrôlable à distance par Internet [Mémoire de master]. Tlemcen: Université Abou Bakr Belkaid; 2017/2018

[5] Benfatma H, Boudjenah SO. Intelligent management of a hydroponic greenhouse powered by a PV system using Internet of Things [Master's thesis]. Tiaret (DZ): Université Ibn-Khaldoun de Tiaret, Faculté des Sciences Appliquées; 2022.

[6] Idoje G, Dagiuklas T, Iqbal M. Survey for smart farming technologies: Challenges and issues. Comput Electr Eng. 2021;92.

[7] Isyraf MA, Abdul Razak B. Remote irrigation monitoring for smart greenhouse. Perak : Universiti Teknologi PETRONAS; 2021 Sep.

[8] Dunn AM, Hofmann OS, Waters B, Witchel E. Cloaking malware with the trusted platform module. In: Proceedings of the 20th USENIX Security Symposium; 2011. p. 395–410

[9] SHAMSHIRI RR. Principles of Greenhouse Control Engineering. Serdang: Universiti Putra Malaysia, Institute of Advanced Technology; 2006/2007.

[10] AMIR S. Conception et réalisation d'un système d'irrigation intelligent [Mémoire de master]. Tizi-Ouzou: Université Mouloud Mammeri de Tizi-Ouzou; 2019/2020.

[11] ABDI MS, BOUMAKEL ODZ. Etude et réalisation d'un système d'irrigation automatisé avec monitoring [Mémoire de master]. Ouargla: Université Kasdi Merbah Ouargla, Faculté des Sciences Appliquées; 2019/2020.

[12] Salman M, Pek E, Lamaddalena N. Field guide to improve water use efficiency in small-scale agriculture: The case of Burkina Faso, Morocco and Uganda. Rome: Food and Agriculture Organization of the United Nations; 2019. p. 1–81.

[13] LABBAS A, MECHMACHE A. Serre agricole intelligente [Mémoire de master]. Tlemcen: Université Aboubakr Belkaïd – Tlemcen, Faculté de Technologie; 2022/2023.

[14] Pereira GP, Chaari MZ, Daroge F. IoT-Enabled Smart Drip Irrigation System Using ESP32. IoT. 2023;4(3):221–243. doi:10.3390/iot4030012

[15] IOT Greenhouse Monitoring System. 20 p.

[16] MENNAL A, BOUHENBEL A. Les réseaux de capteurs sans fil "WSN" : Étude et réalisation d'un prototype à base d'Esp8266 nodeMCU [Mémoire de master]. Khemis Miliana: Université Djilali Bounaâma de Khemis Miliana, Faculté des Sciences et de la Technologie; 2017/2018.

[17] BARHOUM I. Fire alarm system using Arduino [Mémoire de master]. Ouargla: University Ouargla-Merbah Kasdi, Science and Technology; 2022/2023.

[18] khaldi O, Ouadjih K. Smart agriculture monitoring using Internet of Things [Master's thesis]. Annaba: Université Badji Mokhtar - Annaba; 2021.

[19] LALLAM ME, BENALI S. Réalisation d'un onduleur triphasé piloté à travers une interface graphique [Mémoire de master]. Tlemcen: Université Aboubakr Belkaïd – Tlemcen, Faculté de Technologie; 2021/2022.

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed KHIDHER -Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي
جامعة محمد خيضر بسكرة
كلية العلو م والتكنولوجيا
قسم الهندسة الكهربائية

تصريح شرفي

خاص بالالتزام بقواعد النزاهة العلمية لانجاز بحث

(ملحق القرارالقرار 1082 المؤرخ في 27 ديسمبر 2020)

أنا الممضي أدناه،: ..... لياد عبد الرحمان .....

الصفة: طالب ☑

الحامل لبطاقة التعريف الوطنية رقم: ...211.1260.96.. الصادرة بتاريخ:....2024/12/01.....

المسجل بكلية: ... ..العلوم والتكنولوجيا.....قسم: ....الهندسة الكهربائية.....

والمكلف بإنجاز: مذكرة ماستر

تحت عنوان: ....Smart IOT greenhous.....

.....

أصرح بشرفي أني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات المهنية والنزاهة الأكاديمية المطلوبة في انجاز البحث وفق ما ينصه القرار رقم 1082 المؤرخ في 27 ديسمبر 2020 المحدد للقواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.

التاريخ: ..2025/06/23.....

إمضاء المعني بالأمر

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed KHIDHER -Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي
جامعة محمد خيضر بسكرة
كلية: العلو م والتكنولوجيا
قسم الهندسة الكهربائية

تصريح شرفي

خاص بالالتزام بقواعد النزاهة العلمية لانجاز بحث

(ملحق القرارالقرار 1082 المؤرخ في 27 ديسمبر 2020)

أنا الممضي أدناه،: .....بن بوزيد مصطفى.....

الصفة: طالب ☑

الحامل لبطاقة التعريف الوطنية رقم: ..2104365500.. الصادرة بتاريخ:..2024/09/10..

المسجل بكلية: ... العلوم والتكنولوجيا.....قسم: .....الهندسة الكهربائية.....

والمكلف بإنجاز: مذكرة ماستر

تحت عنوان: .....Smart IOT greenhouse.....

.....

أصرح بشرفي أني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات المهنية والنزاهة الأكاديمية المطلوبة في انجاز البحث وفق ما ينصه القرار رقم 1082 المؤرخ في 27 ديسمبر 2020 المحدد للقواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.

التاريخ: .....2025/05/23.....

إمضاء المعني بالأمر