SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V12I3P119
Volume 12 Issue 3, 202-222, March 2025

Original Article

Implementation of an Automatic Control System and an IOT Architecture for Wireless Monitoring Through Bidirectional Communication by RF and Wi-Fi for Registration in the Cloud of Parameters Needed in Greenhouse Operation

Igor Miguel Paredes Cornejo[1], Estefany Jared Rojas Chacón[1], Fernando Paul Salazar Collantes[1], German Alberto Echaiz Espinoza[1], Andrés Ortiz Salazar[2], Elmer Rolando Llanos Villarreal[3]

[1]*Professional School of Engineering Electronics, Universidad Nacional de San Agustin de Arequipa, Arequipa, Peru.*
[2]*Department of Computer Engineering and Automation, Federal University of Rio Grande do Norte (DCA-UFRN), Natal 59072-970, RN, Brazil.*
[3]*Department of Natural Sciences, Mathematics, and Statistics, Federal Rural University of Semi-Arid (DCME-UFERSA), Mossoró 59625-900, RN, Brazil.*

[1]*Corresponding Author : gechaiz@unsa.edu.pe*

Received: 06 December 2024 Revised: 06 March 2025 Accepted: 26 March 2025 Published: 31 March 2025

Abstract *- Food shortages in Peru are frequently exacerbated by roadblocks due to social protests and seasonal variations, restrict food production, and intensify shortages, particularly in regions with limited cultivable land. To address this challenge, an automated control system and IoT architecture for wireless monitoring were developed to replicate specific climatic conditions within greenhouses, allowing the cultivation of crops typically restricted to certain regions and providing a potential solution to improve agricultural productivity and mitigate food scarcity. The system regulates critical environmental parameters such as temperature, irrigation and luminosity, while wireless monitoring of temperature, water level, luminosity, and pH is achieved through RF communication, with data transmitted to a database via Wi-Fi and visualized in real-time using a cross-platform application. In addition, the NRF24L01 radio frequency communication modules were optimal for the application due to their ease of integration with the Arduino UNO development board, affordability, range, and market availability. The system was validated in a pilot-scale greenhouse. Further research is needed to evaluate its long-term scalability and efficiency in diverse agricultural settings, as these aspects remain critical to its broader applicability and impact.*

Keywords *- Greenhouse, Automatic control, IoT, Radio Frequency, Monitoring.*

1. Introduction

Food security is a critical issue facing many countries, particularly in regions where agricultural production is affected by climatic limitations or political instability. In Peru, road blockages due to political demonstrations often disrupt the supply of essential food products, leading to shortages and malnutrition. According to the National Disaster Risk Management Plan, approximately 3,862,572 residents in the provinces of Arequipa, Ayacucho, Huancavelica, Junin and Puno in Peru are exposed to frost periods with temperatures reaching 4°C for up to 30 days. [1] In addition, data from INDECI indicate that between 2003 and 2016, 127,833 hectares of crops were lost, and 682,990 hectares were affected due to frost and cold. These meteorological events significantly impact the health and food security of the affected populations. [2] Furthermore, the most affected population is generally those who live in places located more than 3,000 meters above sea level and whose subsistence resources depend mainly on agricultural activities and livestock raising. Adverse weather conditions severely impact these economic activities. [3] To address this challenge, developing innovative agricdeveloping is crucial the problem is a shortage of monitoring and control systems for creating smart greenhouses that monitor and control specific parameters in Peru, which hinders agricultural production and contributes to food insecurity. Thus, the research gap lies in designing systems that include simultaneous monitoring and control to replicate microclimates. Traditional agricultural methods are vulnerable to disruptions in food transportation networks caused by social and political instability; they are also vulnerable to drastic climate changes, resulting in shortages of essential foods.

The inability to grow food in climate-controlled conditions limits the availability of diverse products, especially in regions with limited cultivable land. This research project aims to develop and implement an automated control system and IoT architecture for wireless monitoring in greenhouses. The main objectives are:

- Establish an IoT architecture for wireless monitoring of greenhouse parameters, including temperature, humidity, brightness, and pH.
- Implement an automated control system for regulating temperature and water levels in hydroponic irrigation systems.
- Design and develop a user-friendly web application for real-time data visualization and parameter control.
- Validate the proposed system by constructing a pilot-scale greenhouse and evaluating its performance under controlled conditions.

The scope of this research includes the design, implementation, and validation of an IoT-enabled greenhouse monitoring and control system. The focus is establishing a communication network among sensors, actuators, and a central controller, enabling real-time monitoring and automated parameter adjustment. The system will be designed to accommodate various greenhouse configurations and crop requirements.

Implementing an IoT-based greenhouse monitoring and control system has significant potential to address food security concerns in Peru. Enabling the self-sufficient cultivation of diverse crops under controlled conditions can help mitigate the impact of transportation disruptions and enhance agricultural productivity. This technology potentially improves food availability, reduces malnutrition rates, and contributes to economic growth in rural communities.

The main contributions of this study are the following:

- Regulation of key environmental parameters, such as temperature, irrigation, and luminosity, to replicate specific conditions within greenhouses. A PI controller was used for irrigation and fuzzy control for temperature, while luminosity was managed based on predefined conditions.
- IoT infrastructure integration enables real-time monitoring of parameters such as temperature, water level, luminosity, and pH using Radio Frequency (RF) communication and data transmission via WiFi.
- The development of a Cross-platform application to show real-time monitoring of greenhouse conditions, improving data accessibility and interpretation.
- Proposal of a technological solution to enhance agricultural productivity in areas with limited cultivable land, addressing food scarcity issues in the Peruvian context.

1.1. State of the Art

The scientific article titled "IoT Architecture Based on Wireless Sensor Network Applied to Agricultural Monitoring: A Case Study of Cocoa Crops in Ecuador" presents a low-cost IOT architecture intended for agricultural monitoring of different types of crops. However, it focuses only on cocoa crops. The article develops a multiplatform application, which is the focus of the research. The basis of the research is that in Ecuador, the main source of income in rural areas (37% of the population) is agriculture, and due to one of its main problems, which is low productivity, the sustainable management of these resources is very important. In order to address this problem, the authors present an IOT architecture based on WSN. This architecture presents a design based on a sensor network. It is important to consider that the architecture should be accessible to farmers, so low cost is prioritized, and that the system should be non-invasive in the development of crops. The architecture presented is of the client-server type, obtaining three layers: an application layer, a service layer and a sensor layer. The sensor layer is composed of five nodes under a mesh topology, a coordinator node and a gateway. An important point in this layer is the measurement of humidity, temperature, PH, conductivity and luminosity. However, the sensors used do not provide high accuracy because of their low cost. Data transmission is done through Xbee S2C between each sensor node, and the data is transmitted to the coordinator node, which is sent to the Gateway through a LoRa module. Consequently, in the service layer, the information hosted in the cloud is managed and processed, obtaining tables, statistical graphs and interactive maps, obtaining reports and notifications according to the set parameters, thanks to the use of Framework MLib, which contains massive data preprocessing algorithms [4].

An important comparative point in the analysed article and the present research is that both present a multiplatform application to address the problem of low food productivity, and both work with an IOT architecture for data collection intended to be low cost. However, although the literature review emphasizes the processing and presentation of data, human intervention is still necessary to solve problems that arise over time; that is why, as a novelty, it was considered of vital importance to develop an automatic control system to control the main growth factors in a plant, avoiding human intervention for the control of factors. Similar to the nodes explained in the article, a transmitter module, a receiver module, and a control module were developed, working with wireless communications through Arduino and using Raspberry Pi to send the information to a database.

Similarly, the research article entitled “Technological architecture for monitoring environmental variables in a museum” presents a system for monitoring temperature in degrees Celsius and relative humidity percentage designed for a museum with completely sealed and covered rooms. The article presents an architecture composed of three levels: a physical level centered on DHT sensors, then a level to acquire data centered on Arduino and a final level of management by Raspberry Pi 3 B+. The purpose of the research is to notify when it is necessary to provide preventive maintenance to the books, extending this system with a control system to carry out preventive actions. In addition, the authors want to provide a system developed with modern technologies, low cost and energetically efficient. The reason why they chose to use “free” software and hardware is to allow access from any device. It is worth mentioning that the acquisition of variables is performed every 5 minutes because the rate of change of temperature and humidity is slow.

They calculate the average values for each variable. This information is stored in a database requested to perform a basic statistical analysis, data visualization, email reporting, and notification. In the physical layer, they use UTP twisted pair cable category 5, suitable for instrumentation and shielding against noise to connect the DHT11 and DHT22 sensors to an Arduino nano through serial communication with a maximum distance of 20 meters. For data processing and visualization, they use a Raspberry Pi 3 B+, which has the open-source Raspberry Pi OS and internet connection via Wifi without extra modules. In this configuration, Raspberry Pi 3 B+ works as a system server and another as a client using the local access program Node-Read, which can only be accessed via remote access. [5]

Although the article analysed is not oriented to agriculture, it presents a monitoring system that could easily be adapted to a greenhouse since both the museum and the greenhouse are closed environments. Now, the article uses DHT11 and DHT22 sensors, very similar to the DHT21 used in this research; likewise, Arduino and Raspberry Pi are used; however, the article uses the Arduino nano and Raspberry Pi 3 B+, which have fewer features and power than the Arduino UNO R3 and Raspberry Pi 4B used in this research, which can develop a control logic, which is not developed in the article presented. Because the authors do not develop a control, they prioritize obtaining reports and the generation of alarms. Another key point is that the authors use direct wired communication, not wireless technology, as presented in this research. This is a key factor because it limits the distance to 20 meters. Finally, although Raspberry Pi is used in both cases, the exposed article uses it as a local database. To access it remote access to the Raspberry Pi is made through a local WiFi network. While in this research, the Raspberry Pi is used as a gateway to store data on a web server. Additionally, remote access can be made through the local network by WiFi to visualize the data flow.

The article "LoRaWAN Based Internet of Things (IoT) System for Precision Irrigation in Plasticulture Fresh-Market Tomato" presents the development and evaluation of an IoT system based on LoRaWAN for precision irrigation in fresh-market irrigation in scheduling treatments were designed and tested. The study implemented and evaluated four irrigation treatments, including irrigation based on crop EvapoTranspiration (ET), soil matrix potential sensors (Watermark 200SS-5) set to -60 kPa (MP60) or -40 kPa (MP40), and the GesCoN fertigation Decision Support System (DSS). The results suggest that the IoT system can be implemented for automated and precision irrigation operations in vegetable and other horticultural crops, enhancing water use efficiency and sustainability. In terms of system performance, the LoRaWAN network successfully transmitted real-time data, having occasional signal losses (5.5% data loss rate) due to gateway positioning and network disconnections. The solenoid valves operated effectively; some response delays were noted due to low voltage power supply and potential clogging. Battery-powered data loggers maintained stable operation throughout the three-month experiment, with solar panels continuously recharging. [6]

Unlike this system, which focuses on soil moisture, the present study integrates both monitoring and automatization, offering a more comprehensive approach to environmental management. One of the limitations was its reliance on intermittent manual intervention for irrigation control. In contrast, the proposed system is fully automated. Furthermore, using NRF24L01 modules ensures reliable data transmission, addressing issues such as data loss and power limitation observed in the LoRaWAN implementation. Finally, in the study "An Automatic Irrigation System Using IoT Devices," an automated irrigation system was implemented. It was complemented by an Android application that allows manual motor control, enabling it to be switched on and off as an alternative. The implementation of this system utilizes an Arduino UNO, GSM modules, an Android application, and temperature and soil moisture sensors. [7]

This research project aims to advance the development of IoT-enabled greenhouse monitoring and control systems, tackling the critical issue of food security in Peru. By enabling the cultivation of diverse crops under controlled conditions, the system has the potential to mitigate the impact of transportation disruptions, improve agricultural productivity, and enhance food availability in regions facing food shortages. Unlike this system, which primarily focuses on irrigation automation, the proposed system monitors and controls a broader set of parameters, including pH, luminosity and water level. Additionally, the web-based application enhances functionality by offering real-time monitoring, remote reset and scalability for diverse greenhouse configurations. An important point of comparison between the analysed article and the present research is that both systems allow farmers to observe the status of the field or greenhouse from anywhere and at any time.

2. Materials and Methods

Figure 1 describes the general block diagram of the project.

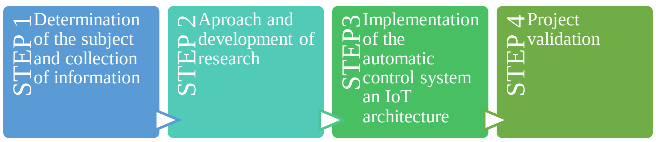

Fig. 1 General block diagram

Figure 2 describes the block diagram of the IoT architecture for monitoring.

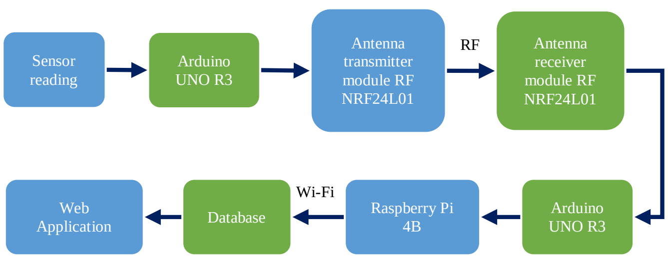

Fig. 2 Block diagram of the IoT architecture for monitoring

Figure 3 describes the block diagram for the level control of a hydroponics tank.

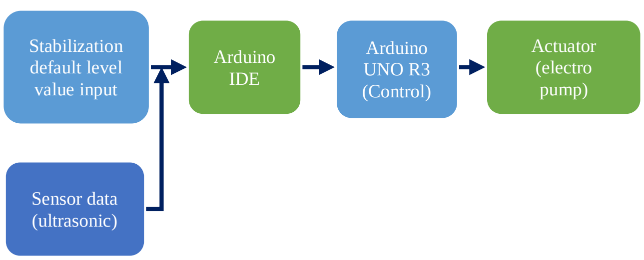

Fig. 3 Block diagram for level control of a hydroponics tank

Figure 4 describes a block diagram for temperature control.

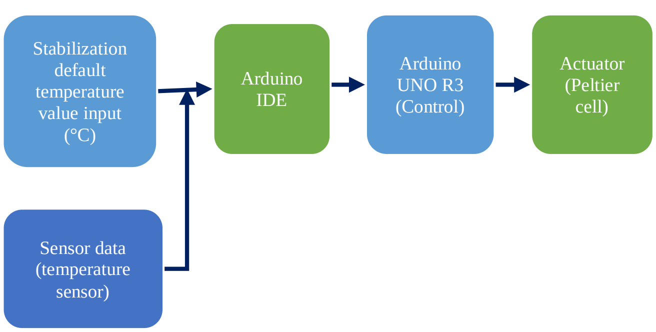

Fig. 4 Block diagram for temperature control

Figure 5 describes a block diagram for hourly brightness control.

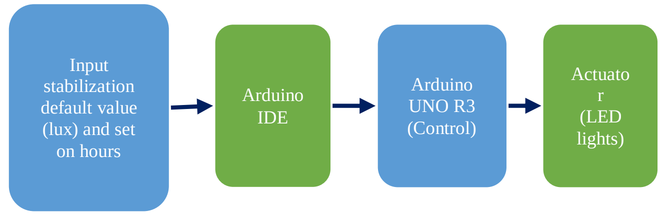

Fig. 5 Block diagram for hourly brightness control

2.1. Materials Required for the Implementation of the Project

List of materials required for implementation in Tables 1, 2 and 3. Arduino was used because it is an open-source hardware and software platform, so there is no need to purchase licenses. Raspberry Pi 4B was chosen because of its low price, operating system-agnostic design, Python capability and USB and Wi-Fi connections included. In the case of the selection of the BH1750, HC-SR04, DHT21 and Liquid pH Sensor +Hydroponic Electrode BNC sensors, the rationale is based on their low price, easy installation and programming. In addition, they are commercially available.

Table 1. List of materials required for implementation

Description	Unit of Measure	Quantity
Raspberry Pi 4B 8Gb + Vilros Case	Unit	1
Cooler Fan 12V 8cm	Unit	2
Cooler Fan 12V 4cm	Unit	1
RF module NRF24L01	Unit	2
Driver Bridge 'H' L298N	Unit	2
Motor Shield VHN2SP30	Unit	1
Luxmeter BH1750	Unit	1
DS3231 clock module	Unit	1
LM2596 DC-DC voltage regulator module	Unit	2
Ultrasonic sensor HC-SR04	Unit	2
Digital humidity and temperature sensor DHT21	Unit	2
LED Tape 5m without silicone	Unit	2
Liquid pH Sensor +Hydroponic Electrode BNC	Unit	1
Peltier Kit 12706 with cell, heatsink and thermal paste	Unit	1
Water Pump 12 V. 3.5 L/min 2 A	Unit	1
Switching power supply AC/DC 12 VDC 10 A	Unit	1
Switching power supply AC/DC 12 VDC 10 A	Unit	1
White PETG roll	Unit	1
ARDUINO UNO R3	Unit	4

Table 2. List of required software

Description	Unit of measure
ARDUINO IDE	Unit
MATLAB/Simulink (UNSA Student License)	Unit
Pusher	Unit
Visual Studio Code	Unit
Tinkercad	Unit
Thonny, Python IDE	Unit

Table 3. List of materials required for prototype structure

Description	Unit of measure	Quantity
Premium Acrylic Plate 4mm 1.25 x 1.85	Unit	1
MDF board	Unit	1
Aluminum angles	Unit	4
Screws	Unit	100
UTP cable	Meters	8
Sika acetic silicone	Unit	1
6″ PVC pipe	Unit	1

2.2. Mathematical Modeling and Simulation of Control Systems

2.2.1. Mathematical Model of the Level Control System

Figure 6 below shows the filling and draining scheme of a tank.

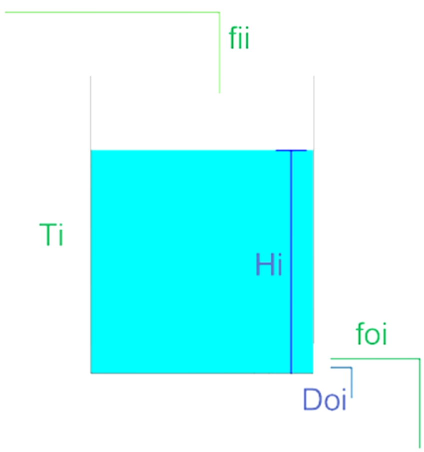

Fig. 6 Schematic diagram of filling and draining a tank

Therefore, the resulting expression of the Equation of Obtained Motion (EOM), considering the mass balance applied to the water level in the tank, is:

$$A\,T_i \frac{dh_i}{dt} = F_{ii} - F_{oi} \quad (1)$$

Where:

h_i water height in the tank $\boldsymbol{Ti}$

A_{T_i} tank base area $\boldsymbol{Ti}$

F_{i_i} Tank inlet flow rate $\boldsymbol{Ti}$

F_{o_i} tank outlet flow rate Ti

It is assumed that the inlet volumetric flow rate to the tank is directly proportional to the current applied to the pump, obtaining:

$$F_{i_i} = K_p I_p \quad (2)$$

In addition, according to Bernoulli's equation for small orifices, it is possible to obtain the tank outlet velocity.

$$\frac{1}{2} mv^2 = mgh$$

$$v_{o_i} = \sqrt{2gh_i} \quad (3)$$

Thus, the volumetric flow rate out of the tank is expressed as:

$$F_{o_i} = A_{o_i} v_{o_i} \quad (4)$$

Where:

A_{o_i} tank outlet orifice area Ti

v_{o_i} tank outlet flow velocity Ti

From the figure of the tank, the following equations are defined:

$$\Delta V_1 = f_{i1} - f_{o1}$$

Since the area of the base of the tank is constant, the equation develops as follows:

$$A_1 \Delta h_1 = f_{i1} - f_{o1}$$

$$A\dot{h}_1 = f_{i1} - f_{o1}$$

From equations (2) and (4), the above expression is rewritten as:

$$A\dot{h}_1 = K_p I_p - A_{o1} v_{o1}$$

Finally, replace equation (3) in the previous expression:

$$A\dot{h}_1 = K_p I_p - A_{o1}\sqrt{2gh_1}$$

$$\dot{h}_1 = \frac{K_p}{A} I_p - \frac{A_{o1}}{A}\sqrt{2gh_1} \quad (5)$$

Linearization

From equation (5), which defines the tank, the term $B_0 h$ is calculated by deriving equation (5) and evaluating at an average height of 15 cm. The linearized system has the form:

$$\dot{h} = u\,A - B_0 h$$

System parameters:

- $d_{o1} = 0.6\ cm$
- $A_{o1} = 0.2827\ cm^2$
- $g = 978 \frac{cm}{s^2}$
- $A = 888\ cm^2$

$$\dot{h} = \frac{K_p}{A} I_p - \frac{A_{o1}}{A} \sqrt{2gh}$$

$$\frac{d\dot{h}}{dh} = -\frac{A_{o1}}{2A} \sqrt{\frac{2g}{h}}\Bigg|_{h=15cm}$$

$$\frac{d\dot{h}}{dh} = -\frac{A_{o1}}{2A} \sqrt{\frac{2g}{15}}$$

$$B_0 = \frac{d\dot{h}}{dh} = -0.00182$$

System Model

The linearized system is:

$$\dot{h} = \frac{u}{A} - B_0 h$$

$$\dot{h} = \frac{u}{888} - 0.00182h$$

In addition, by Laplace, the transfer equation is obtained:

$$H(s) = \frac{h(s)}{u(s)} = \frac{0.619}{549.505s + 1}$$

Although nonlinearities can cause discrepancies between the model and the system's actual behaviour, particularly when operating far from the equilibrium point—resulting in control errors, instabilities, or unexpected saturations—the linearization method remains highly advantageous. It simplifies mathematical analysis and facilitates the design of controllers using tools developed for linear systems, such as transfer function analysis. This approach is especially useful in small operational ranges with less significant nonlinearities, offering a practical balance between accuracy and computational manageability. [8]

2.2.2. Simulation and Auto-Tuning of the PI Controller for the Level Control System

It is possible to use the Tune function of the Simulink software to perform the auto-tuning of the PID controller. This function provides estimated values of the PID according to the established transfer function, and it is also possible to observe a change depending on the desired response time. The block diagram is presented below in Figure 7:

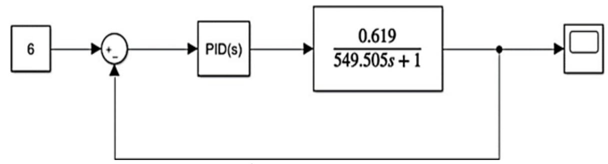

Fig. 7 Block diagram of the system implemented in simulink

In this software, there is a block designated as PID(s). Figure 8 shows where the tune configuration is performed on the block's form, in addition to making changes to the PID values to observe the result in the simulation.

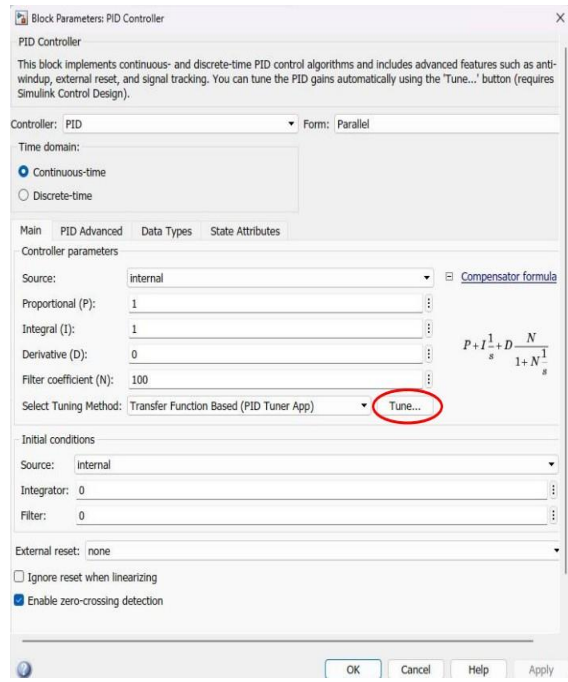

Fig. 8 PID block parameters in simulink

The values obtained, as well as the response time of the controller with these values, are presented below in Figure 9.

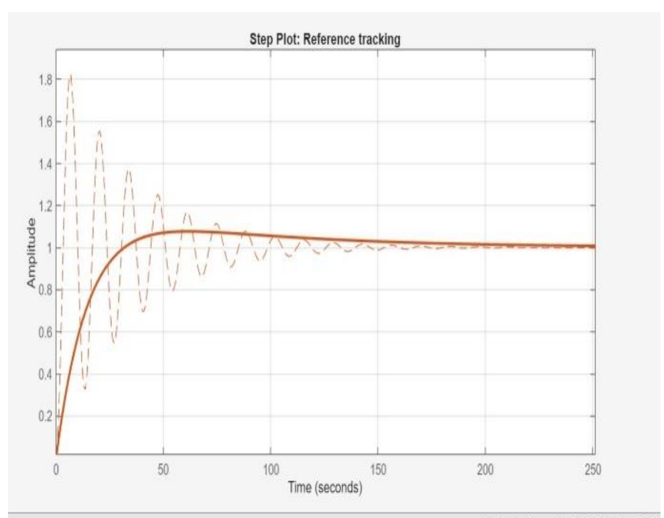

Fig. 9 Auto tuning of PID parameters

The proportional value of the controller allows the controller output to adjust proportionally to the current error. For instance, if the difference between the setpoint and the actual value is 1 unit, the proportional action will generate a controller output of 70 units. Meanwhile, the integral value of the controller is responsible for accumulating the error Over time. The PID tuning value indicates that the error's integral will be multiplied by 0.8 to adjust the controller output. Based on observations from the simulation, the system response shows an overshoot of approximately 1% and a settling time of around 4 minutes.

2.2.3. Temperature Control System Control Logic

Fuzzy control allows for the expression of logical values between true and false. For decision-making and system control, use fuzzy logic, meaning decisions are made based on various variables using approximate or fuzzy rules. [9, 10] These rules are used to make decisions in different scenarios, which are evaluated through a degree of truthfulness by comparing the rules and the established input. The fuzzy controller has two main features that support its selection as a control system. First, it does not require a detailed mathematical model of the system for design. Additionally, a fuzzy controller involves minimal computational load. [11]

The fuzzy controller design is applied in a temperature control system using a Peltier cell. This Peltier cell operates over a current range of 0 to 3.2 amps, where 0 amps represents the minimum temperature and 3.2 amps represents the maximum temperature. In this control system, the control action is represented by a PWM signal (0 to 255 pulses) generated by the Arduino UNO, allowing adjustment of the current range required for Peltier cell operation. The temperature sensor used is the DHT21 model. Inputs to the fuzzy controller are determined from the difference between the temperature setpoint established in the greenhouse and the temperature measured at a specific moment, as well as the rate of change of that difference. This enables the controller to adjust the output effectively to maintain the desired level of greenhouse temperature. Based on this theory, the following rules were applied for temperature control using fuzzy logic in Table 4:

Table 4. Structure of the fuzzy controller

Fuzzy Rules	Output (PWM Value on Current Control Driver (0-255)
Temperature between 22°C - 25°C	PWM response between 230 - 255
Temperature between 23.5°C - 26.5°C	PWM response between 160 - 230
Temperature between 25°C - 28°C	PWM response between 0 - 160

These rules present control of the PWM response given to the driver based on the value recorded in the temperature sensor; below is the graphical representation of the rules presented in Figure 10.

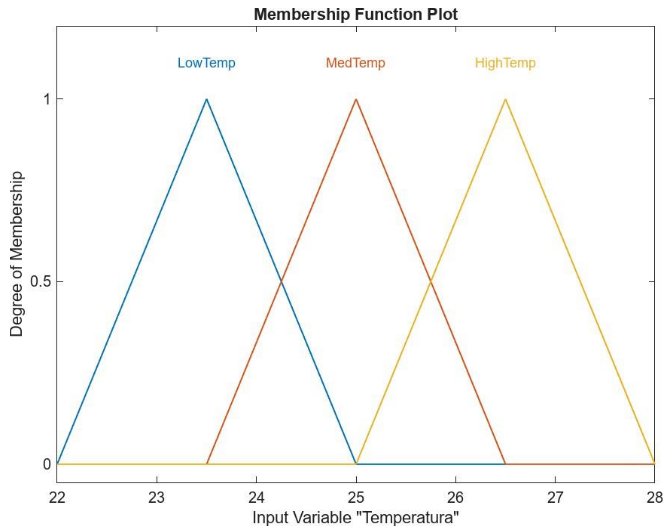

Fig. 10 Input configuration of the fuzzy controller

The figure below shows the configuration of the controller outputs represented with a PWM signal $(0 - 255)$. Three possible controller outputs are configured, which will be selected according to the decision of the fuzzy controller, considering the rules established above in Figure 11.

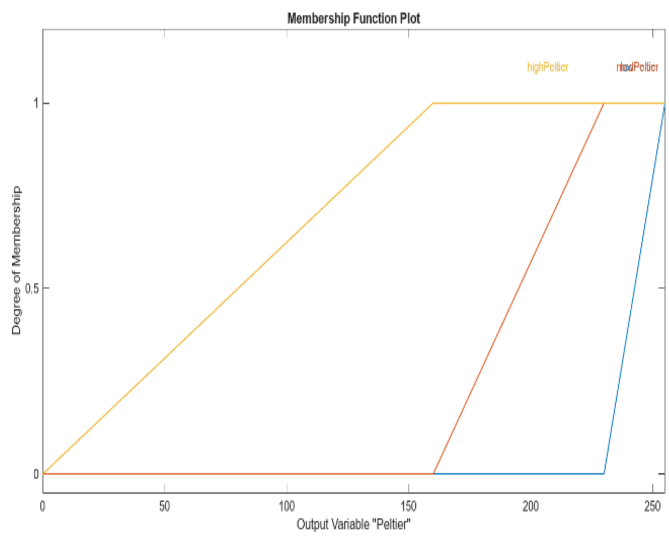

Fig. 11 Output configuration of the fuzzy controller

Likewise, the rules are assigned a weight, and the resolution of these rules is established:

- If the measured temperature is in the "LowTemp" range, then the controller output will be "highPeltier" $(230 - 255)$.
- If the measured temperature is in the "MedTemp" range, then the controller output will be "medPeltier" $(160 - 230)$.
- If the measured temperature is in the "HighTemp" range, then the controller output will be "lowPeltier" $(0 - 160)$.

Figure 12 presents fuzzy controller rules.

	Rule	Weight
1	If Temperatura is LowTemp then Peltier is highPeltier	1
2	If Temperatura is MedTemp then Peltier is medPeltier	1
3	If Temperatura is HighTemp then Peltier is lowPeltier	1

Fig. 12 Fuzzy controller rules

Figure 13 shows the controller's response to different temperature values, considering the abovementioned rules. The setpoint considered in the control system is 25° C. According to the simulation before this temperature, the controller output will be 220 pulses.

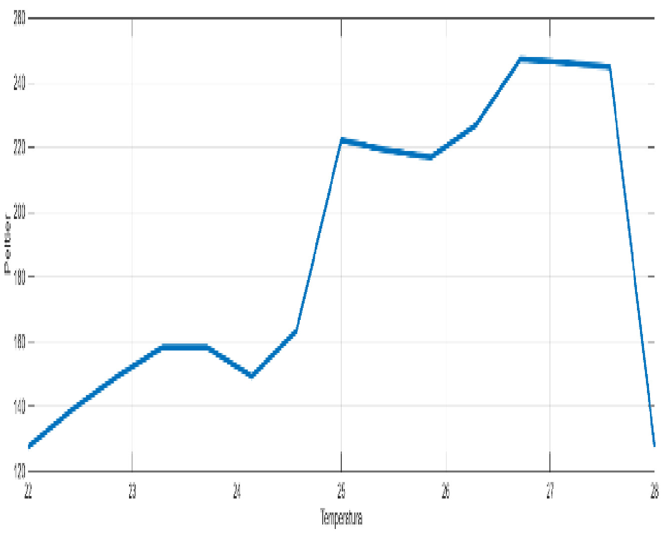

Fig. 13 Response of the fuzzy controller to temperature changes

2.3. Electronic Design

2.3.1. Emitter Module

The transmitter module represents the initial stage in the process of real-time parameter monitoring. The purpose of this module is to measure water pH, temperature, humidity, luminosity, and water level and then transmit these readings via a radiofrequency module to a remote receiver. Monitoring the water level is essential, as a hydroponic irrigation system is being considered, where the water is maintained at a preset level (setpoint) and flows continuously to prevent stagnation.

The transmitter module includes an Arduino Uno programming board, an NRF24L01 RF module, an AM2302 temperature sensor, an HC-SR04 ultrasonic sensor, a PH-4502C pH sensor, and a BH-1750 luminosity sensor. The actual wiring of the transmitter module is shown in Figure 14. In addition, the schematic diagram of the emitter module is shown below, according to the actual connection with its respective pins. Figure 15 presents a schematic diagram of the emitter module. From the datasheet of the RF module NRF24L01 from NORDIC SEMICONDUCTOR, Table 5 describes the function of the pins used for data transmission.

Table 5. Structure of the fuzzy controller

No	Pin Name	Pin Function	Description
1	CE	Digital Input (7)	Chip Enable. Enables RX or TX mode.
2	CSN	Digital Input (8)	SPI (Serial Peripheral Interface) chip selection.
3	SCK	Digital Input (13)	SPI watch.
4	MOSI	Digital Input (11)	SPI slave data input
5	MISO	Digital Output (12)	SPI slave data output, with a choice of three states.
6	IRQ	Digital Output (Not connected).	Maskable interrupt pin. Active low
7	VCC	Power 3.3 V	Power supply (+1.9V-+3.6V DC)
8	GND	Power (GND)	Ground (0V)

Data transmission via the MOSI (Master Out Slave In) pin is a unidirectional operation used in devices that follow the SPI communication protocol. In this protocol, the master device (Arduino) sends data to the slave device (NRF24L01) [12].

After correctly configuring the pins, the Arduino sends a clock pulse on the SCK pin. With each clock pulse, the least significant bit of the data in the output register is placed on the MOSI pin. If the bit is 1, the MOSI pin goes high; if the bit is 0, the MOSI pin goes low.

This process repeats for each bit of data. Besides collecting data from sensors, the Arduino receives a reset command from the web application, which does not interfere with the normal operation of the system. Once the data transmission is complete, the CSN pin is set high, signaling to the NRF24L01 module that the transfer has finished and the module can process the received information.

2.3.2. Receiver Module

The purpose of this module is to receive data from the sender module and transmit this data to a database and a web application. An Arduino UNO was used as the programming board for this module, an NRF24L01 was used as the RF module, and additionally, a Raspberry Pi 4B was used to send the data received from the transmitter to a database so that these data can be viewed in a web application. [13] Figure 16 presents the actual connection diagram of the receiver module, and Figure 17 shows a schematic diagram of the receiver module.

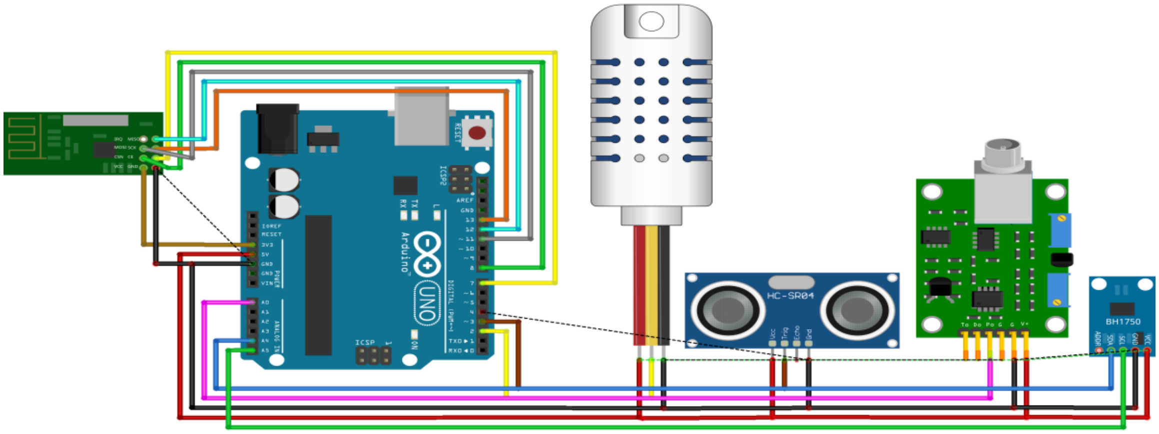

Fig. 14 Actual wiring diagram of the emitter module

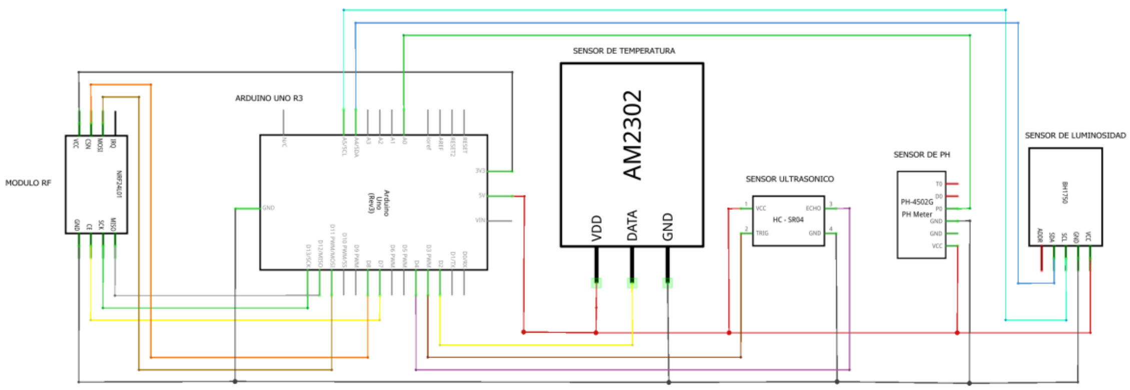

Fig. 15 Schematic diagram of the emitter module

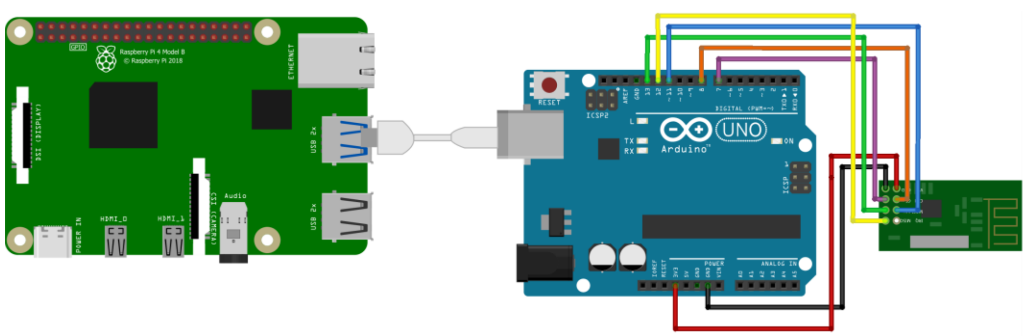

Fig. 16 Actual connection diagram of the receiver module

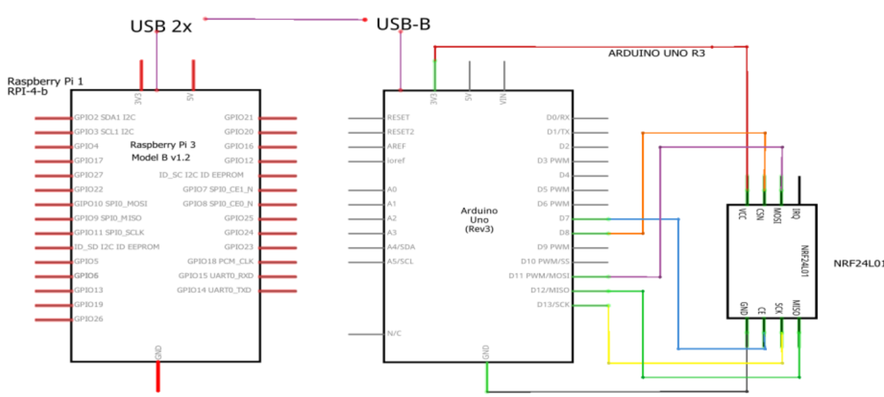

Fig. 17 Schematic diagram of the receiver module

2.3.3. Control Module

This module manages the water supply control for the main tank based on data from the ultrasonic sensor HC-SR04; this sensor was chosen primarily due to its simplicity in measuring water levels, while other alternatives, such as capacitive or pressure-based sensors, offer enhanced precision, they can be more complex to integrate for a prototype. PID control adjusts the intensity supplied to the water pump via an L298N driver. Although higher-performance drivers, such as the BTS7960, offer greater current capacity and efficiency, the L298N remains a reliable option. The pump operates at a voltage level of 12V, requiring a power source that can provide this voltage. Lighting control utilizes a ZS-042 RTC1 clock module, which regulates the light intensity in the prototype at specific times of the day through automatic monitoring. Operation times are preset in the code, and when the module registers a specified time, the light intensity is adjusted by the L298N driver. The system uses an Arduino UNO board for control and code implementation. Additionally, temperature control is managed using a separate Arduino UNO and a Peltier cell. This cell adjusts the temperature via a VHN2SP30 motor shield, which modulates a PWM signal through fuzzy control embedded in the Arduino code. This shield is chosen due to the amperage limitations of the L298N drivers. Fans circulate air, helping to distribute the temperature generated by the Peltier cell. Figure 18 presents an actual wiring diagram of the control module.

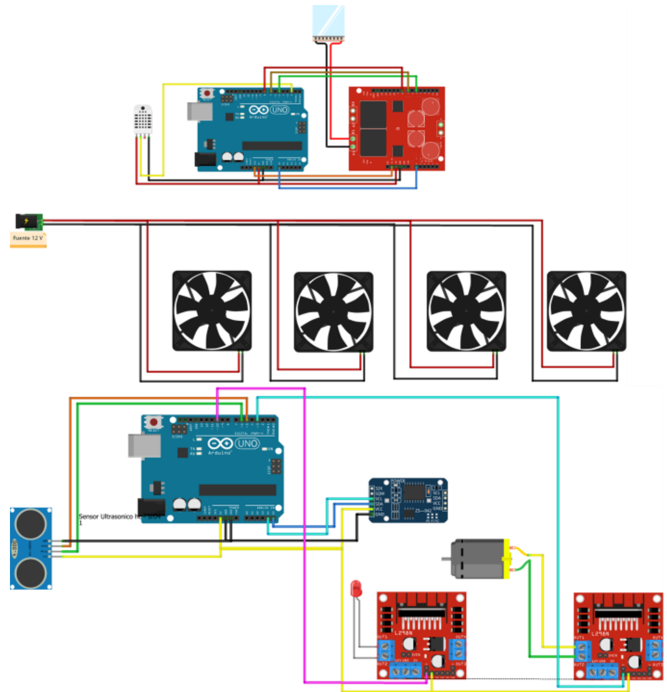

Fig. 18 Actual wiring diagram of the control module.

Figure 19 and 20 presents a schematic diagram of the control module.

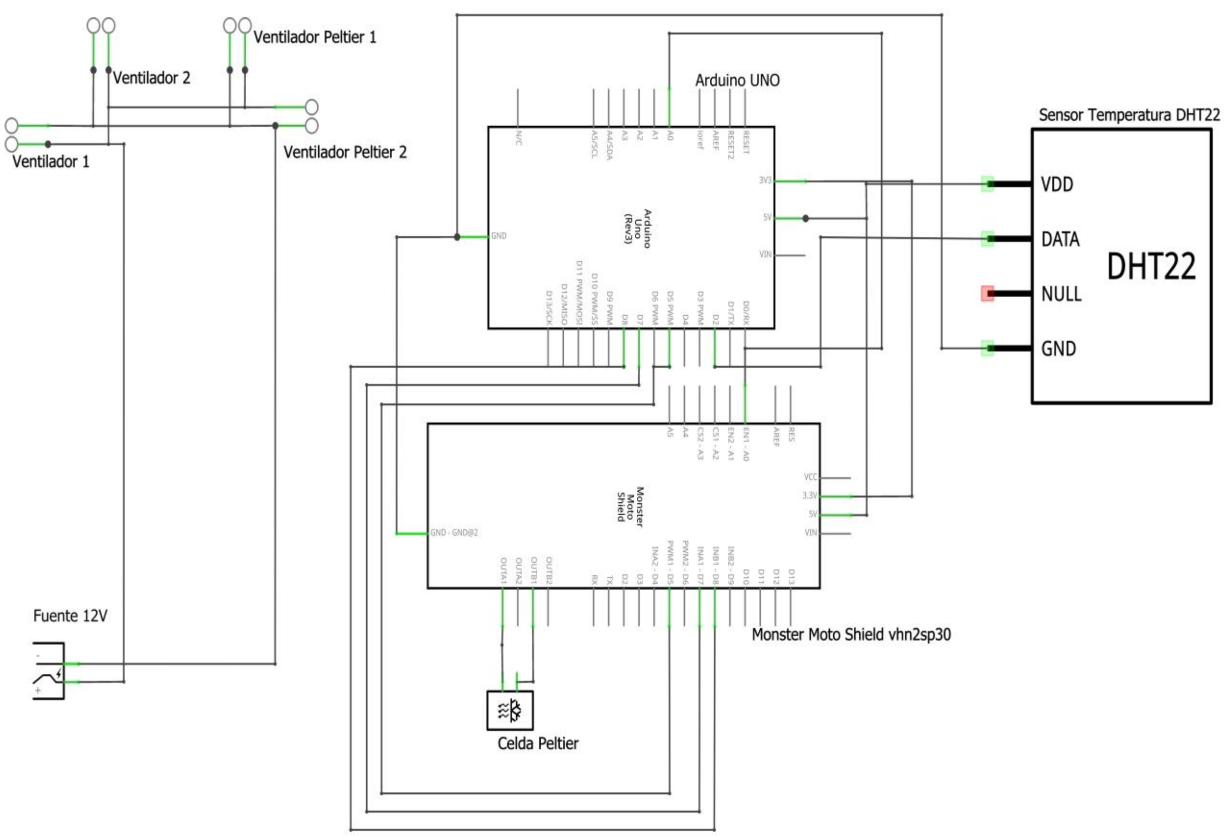

Fig. 19 Actual wiring diagram of the control module. (Part 1)

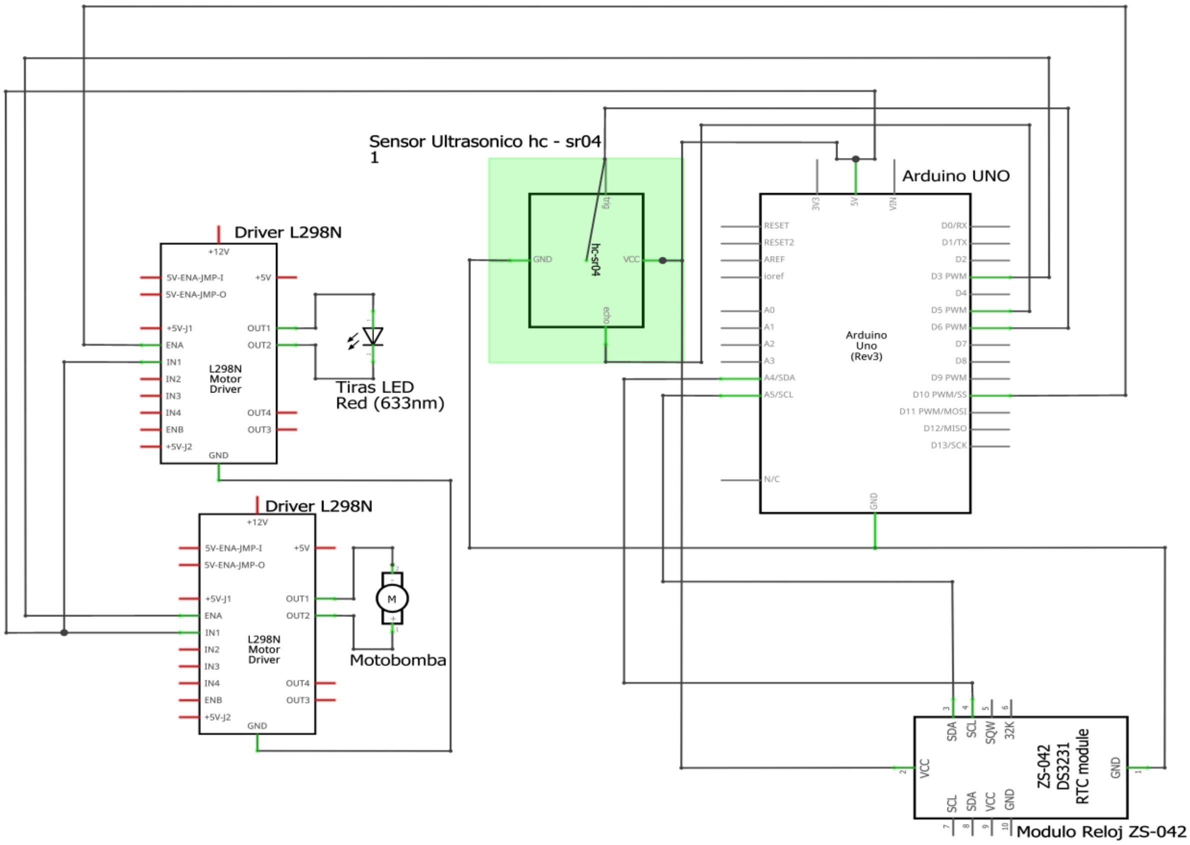

Fig. 20 Actual wiring diagram of the control module. (Part2)

3. Materials and Methods

3.1. Mechanical Design

3.1.1. Pilot Prototype

Figures 21 and 22 show, from different perspectives, the plan of the pilot prototype implemented to validate the research project. This design was made using the free software Tinkercad.

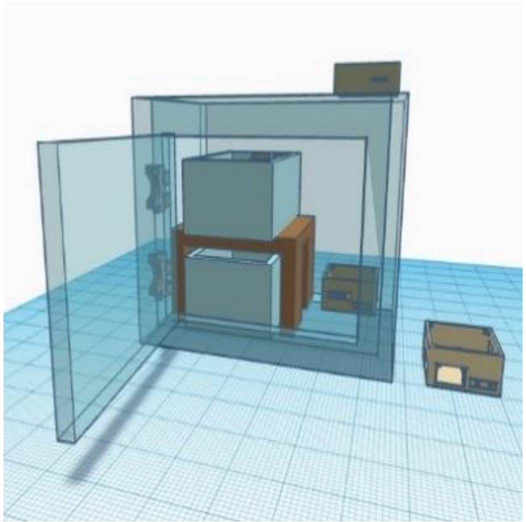

Fig. 21 Front view of the pilot prototype

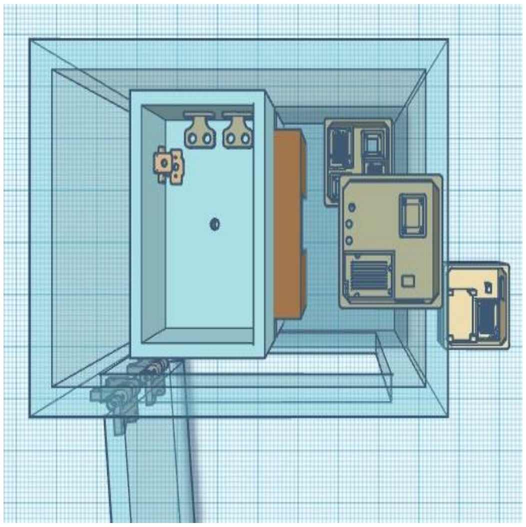

Fig. 22 Top view of the pilot prototype

3.1.2 Module Structures

Supports were designed and fabricated for the emitter, receiver, and control modules to ensure proper organisation and functionality. These supports were modeled using Tinkercad software and 3D-printed with an Artillery Genius Pro 3D printer. Each support was tailored to securely house the components of its respective module, providing structural stability and facilitating proper ventilation where necessary.

Emitter Module Holder

The support for the emitter module was designed to accommodate all components related to this module. Special attention was given to structural integrity and accessibility.

Figure 23 shows the top view of the emitter module holder, highlighting its layout and organization.

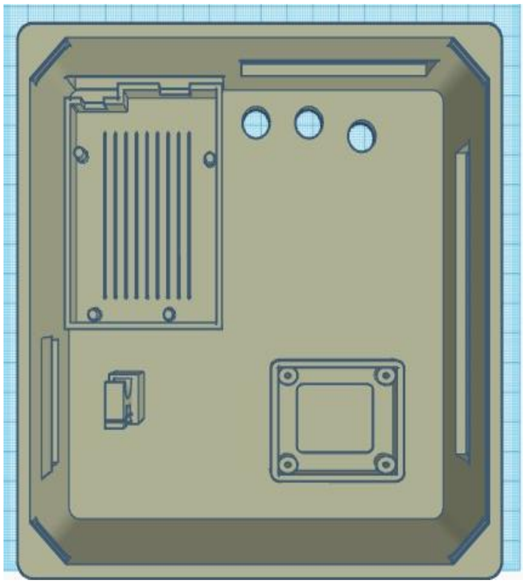

Fig. 23 Top view of emitter module

Receiver Module Holder

The receiver module holder was similarly designed to ensure the proper placement of all associated components. Ventilation considerations were incorporated where needed.

Figure 24 provides a top view of the receiver module holder, illustrating its design features.

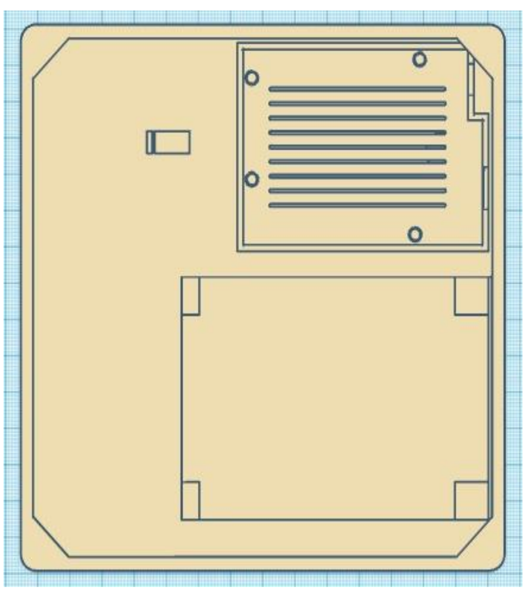

Fig. 24 Top view of the receiver module

Control Module Holder

The support for the control module includes additional features, such as ventilation holes in the lower section, to improve airflow and prevent overheating of sensitive components.

Figure 25 displays the top view of the control module holder, showing its arrangement and functional elements.

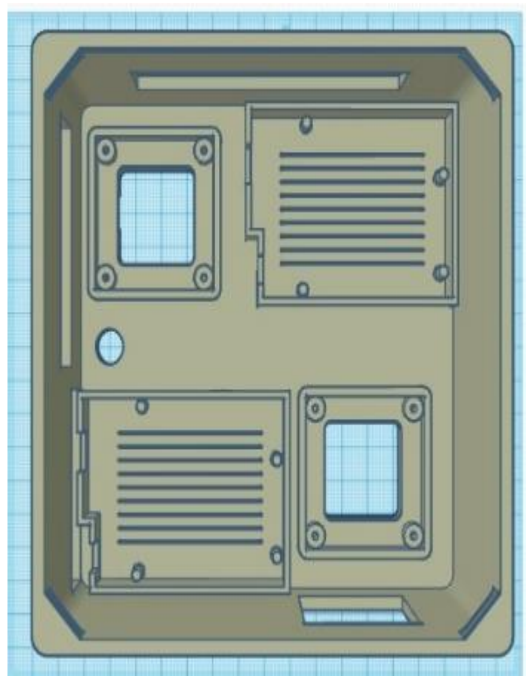

Fig. 25 Top view of the control module

3.2. Software Design

3.2.1. Script for the Transmitter and Receiver Modules

The script for the transmitter and receiver modules was designed to efficiently handle the sensor data transmission to the receiver module.

Although parameters such as temperature, humidity, and PH change slowly over time (taking minutes to have a significant variation), the general data-sending rate was set at 500 ms.

This is because the collection of water level data for control, as this process must be fast. The HC-SR04 sensor has a minimum waiting time between measurements of 20 ms, so a time of 30 ms was considered for obtaining the level, luminosity, humidity and temperature, adding up to 120 ms; for the PH, the same 30 ms was taken, however, as the average of 10 measurements is taken, a time of 300 ms is taken.

Finally, 80 ms was given to store the data in variables and send them through the serial port. This adds up to 500 ms.

Figure 26 provides the flow diagram for the transmitter module, illustrating the logic and processes involved.

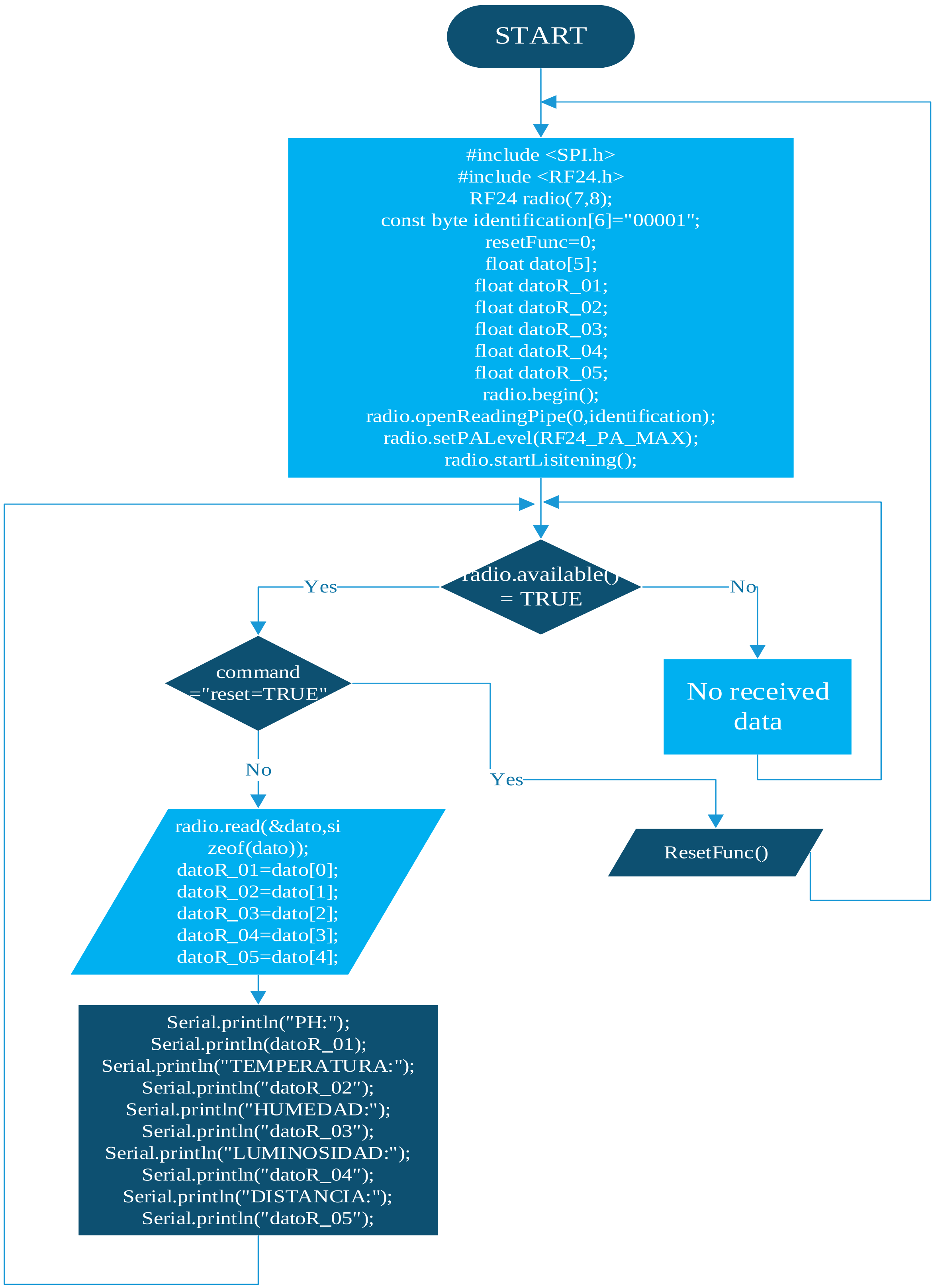

Fig. 26 Flow diagram transmitter module

Figure 27 provides the flow diagram for the receiver.

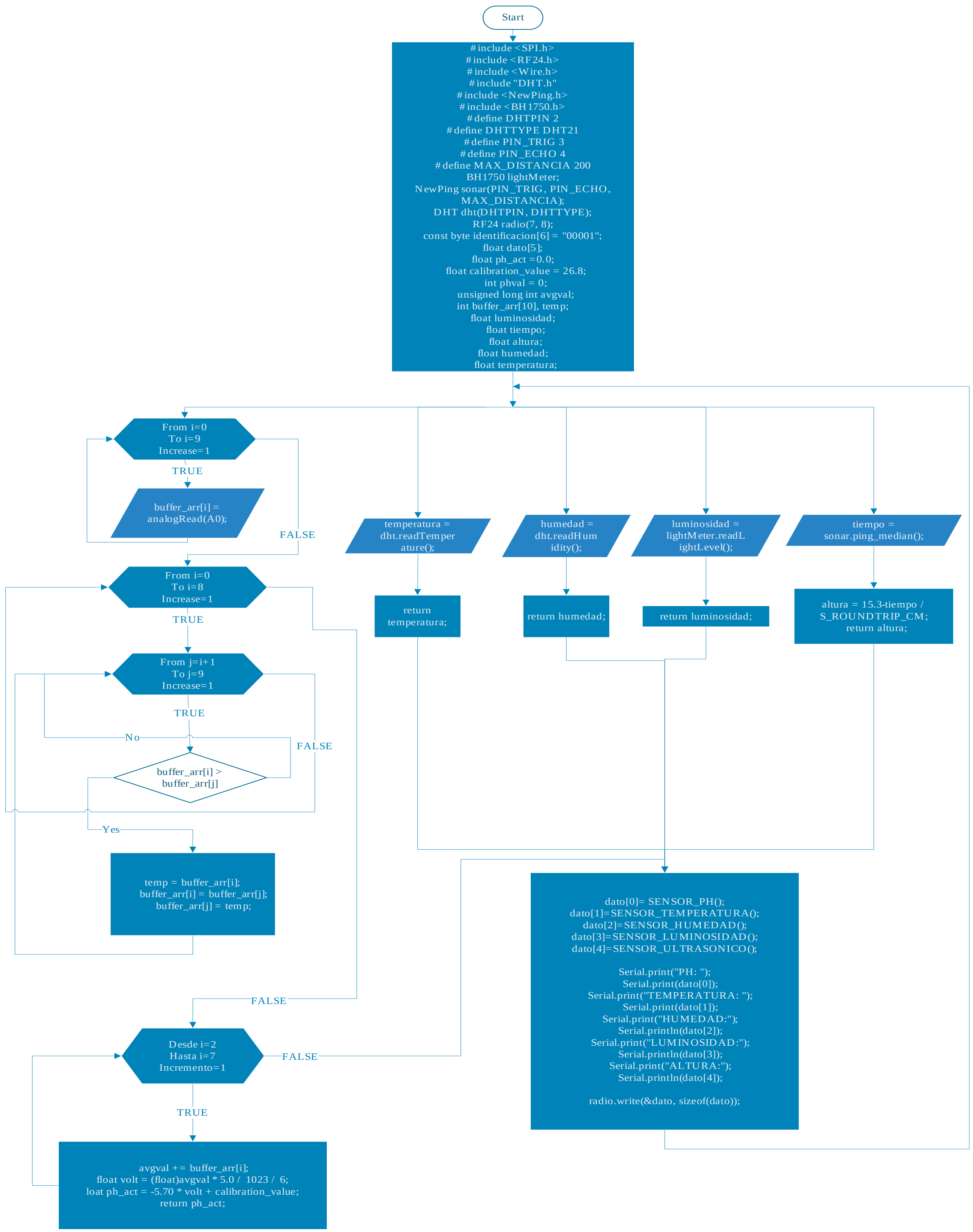

Fig. 27 Receiver module flow diagram

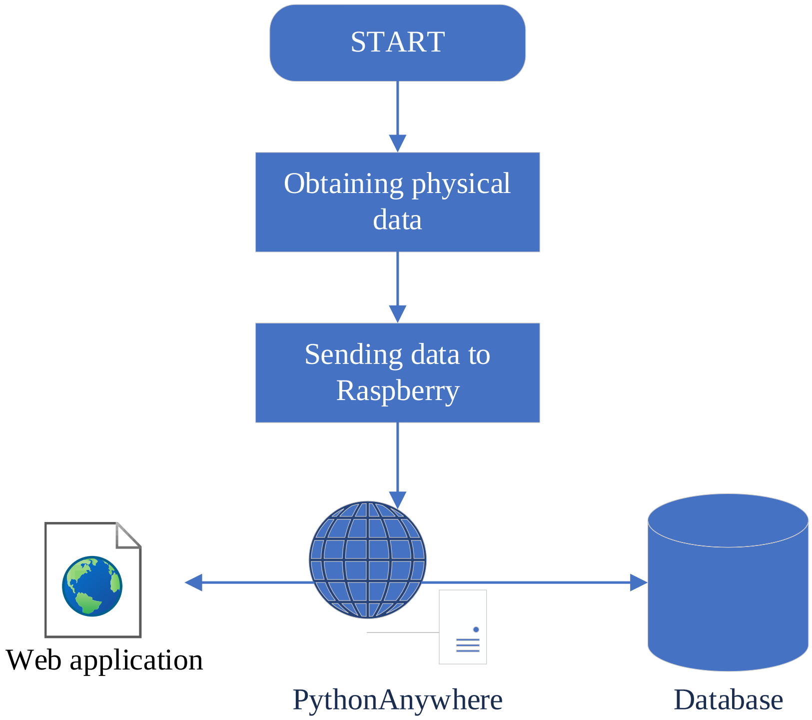

Fig. 28 Flowchart for sending data to the database Web application

3.2.2. Programming for Sending Data to the Database and Web Application

To initiate the Python script on the Raspberry Pi, it must be configured to run automatically upon booting. This is achieved by adding a specific line of code to the Raspberry Pi's system startup file, ensuring the program starts as the device powers on. The Raspberry Pi receives sensor readings from the Arduino (receiver module) via serial communication. Then, these readings are sent to a web server hosted on PythonAnywhere. The server processes the data and performs two key tasks:

- Storing the data in a database (SQLite).
- Updating the web application for real-time visualization and analysis.

The server is hosted on PythonAnywhere, which provides a free infrastructure for Python application deployment. The Raspberry Pi communicates with the server to transmit greenhouse monitoring data, which is subsequently stored in an SQL database. The database maintains an organized structure with columns for a Unique Identifier (ID), timestamp, pH level, temperature, luminosity, and water level.

Pusher was used alongside HTML and JavaScript for the web application to enable real-time updates. Pusher is integrated into both the server and web application, ensuring that any changes on the server are automatically reflected in the web interface. When the Raspberry Pi invokes the *"/update_parameters"* endpoint on the server, Pusher triggers an "update" function via the "parameters" channel, enabling seamless updates in the web application. Figure 28 illustrates the flowchart for sending data to the database and web application, providing a visual representation of the process.

3.2.3. Programming for the Control Module

The control module was programmed to manage the operation of actuators and monitor system feedback in real time. Figure 29 presents the flow diagram of the control module, detailing the processes for receiving sensor input, making control decisions, and executing actuator commands.

4. Results and Discussion

4.1. Results

4.1.1. Explanation of the Construction of the Pilot Prototype

The pilot prototype was constructed using a 3 mm acrylic sheet for the walls, base, and roof, providing a lightweight yet durable structure. An MDF sheet was used as a base support, enhancing stability. Aluminum angles were added to reinforce the corners of the structure. Two plastic containers were integrated into the design.

The lower container serves as a water reservoir, while the upper container houses the lettuce plants along with the pH and ultrasonic sensors. An MDF base was constructed and securely positioned over the reservoir to support the upper container. Additionally, a door was installed on the front of the prototype to facilitate easier access to the internal sensors and components. In Figure 30, the complete construction of the pilot prototype is shown.

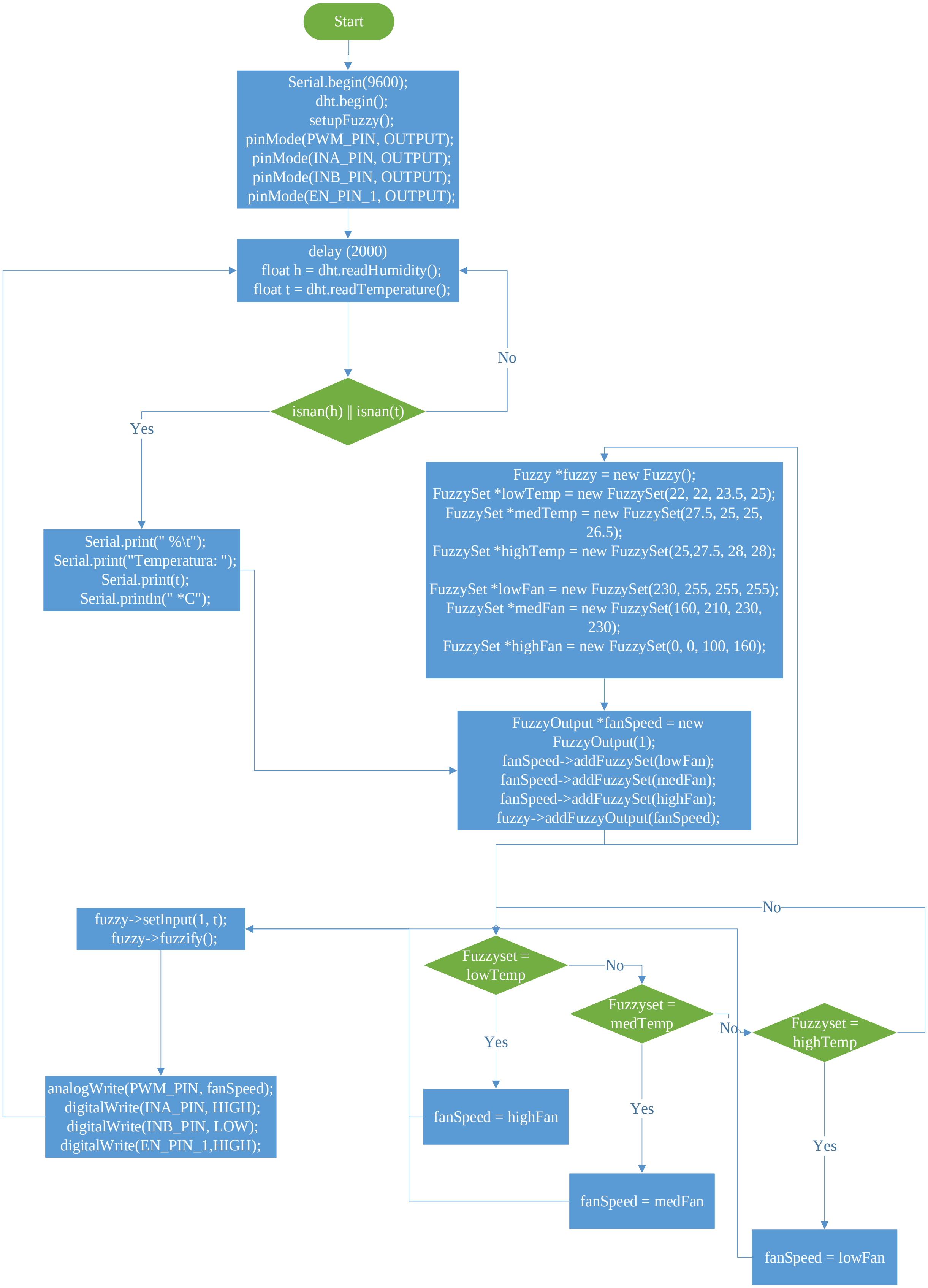

Fig. 29 Flow diagram of the control module

Fig. 30 Construction of the pilot prototype

The operational phase of the prototype, including the placement of sensors and the lettuce plants, is depicted in Figure 31.

Fig. 31 Pilot prototype in operation

Figure 32 illustrates the temperature and ventilation control system implemented within the prototype.

Fig. 32 Temperature and ventilation control system

The level control system is presented in Figure 33.

Fig. 33 Level control system

Finally, the monitoring system for both water level and pH values is demonstrated in Figure 34, emphasizing the integration of sensors and their placement within the structure.

Fig. 34 Level and pH monitoring system

4.1.2. Testing of the Level Control with Setpoint at 5 cm

The actual response of the level control system for a setpoint of 5 cm is analyzed below. A comparison between the simulated response in Simulink and the real system response demonstrates a good agreement regarding settling time and overshoot. Water level measurements were collected from the Arduino serial port over a seven-minute period, enabling the creation of a graph that shows the water level in centimeters as a function of time. The simulation produced a settling time of approximately 5 minutes and an overshoot of 7.6%. In contrast, the real system achieved a settling time of 3 minutes and 46 seconds, with an overshoot of 14%. The slight discrepancy in overshoot is likely caused by disturbances not considered in the system modeling, such as noise from the ultrasonic sensor. Despite this, the differences observed are minimal and do not significantly impact the system's behavior, validating the proposed simulation model.

In Figure 35, the temporal response of the level control system with a setpoint of 5 cm is displayed.

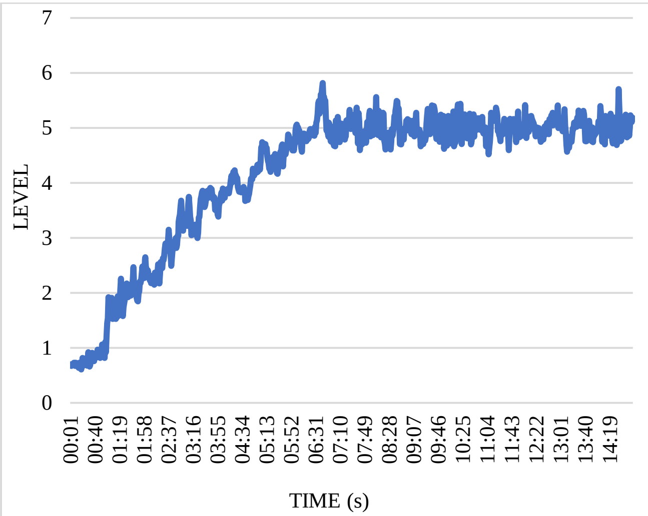

Fig. 35 Temporal response of the level control system (SP: 5 cm)

Figure 36 provides the simulation of the temporal response for the same setpoint.

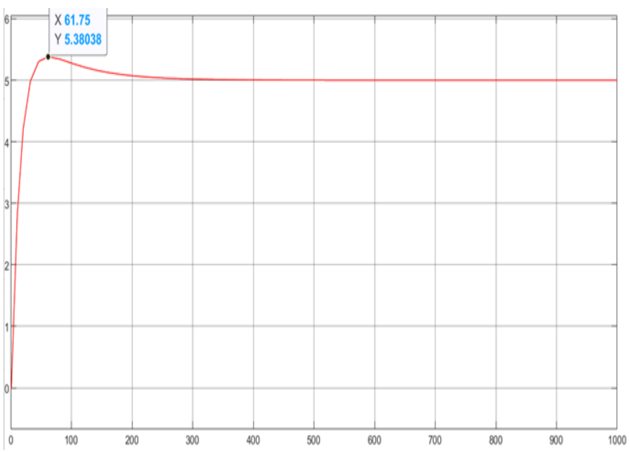

Fig. 36 Capture of simulation on Simulink of the time response of the level control system (SP: 5 cm)

4.1.3. Testing of the Level Control with Setpoint at 8 cm

The response of the level control system to a setpoint of 8 cm, implemented with a PI controller, is presented below. The time response shows that the system exhibits no overshoot and achieves a settling time of approximately 9 minutes.

The simulation, however, indicates a settling time of 5 minutes and 48 seconds with an overshoot of 7.5%. This divergence may arise from real-world disturbances not considered in the simulation. Nonetheless, the overall performance aligns closely with the expected behavior.

Figure 37 shows the temporal response of the level control system at a setpoint of 8 cm.

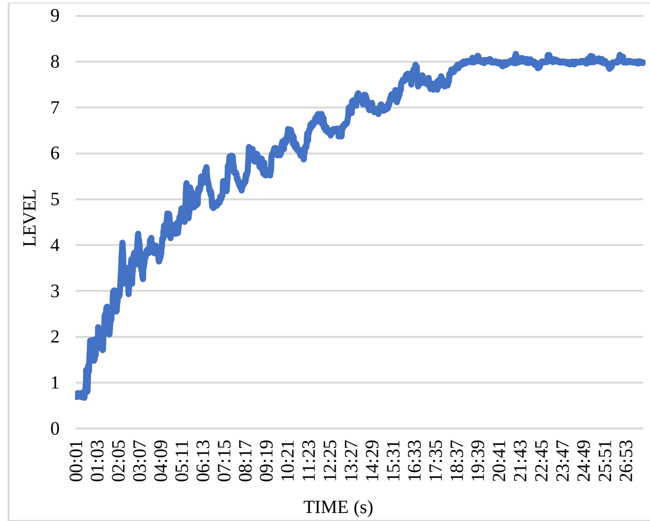

Fig. 37 Temporal response of the level control system (SP: 8 cm)

Figure 38 presents the simulated time response for the same setpoint, offering insight into the modeled system dynamics.

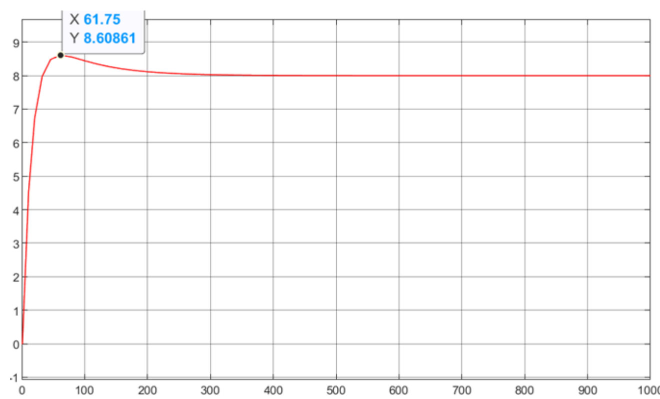

Fig. 38 Capture of simulation on Simulink of the time response of the level control system (SP: 8 cm)

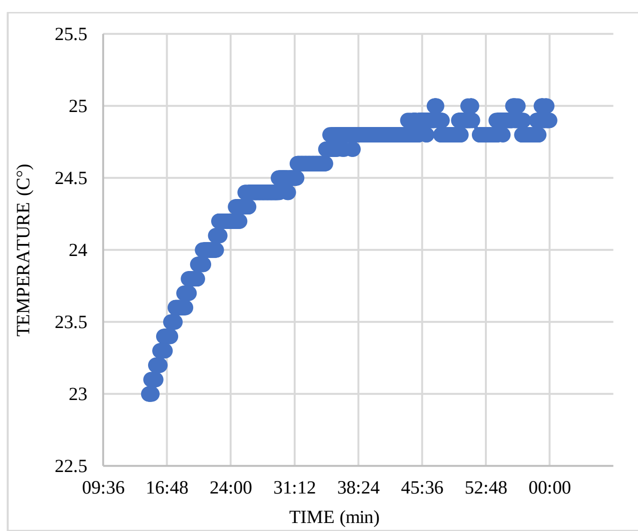

Fig. 39 Temporal response of the temperature control system (SP: 25°C)

4.1.4. Testing of Fuzzy Temperature Control with Setpoint at 25°C

The temperature control system's response to a setpoint of 25°C, implemented using a fuzzy controller, is evaluated here. The time response shows no overshoot and a settling time of approximately 30 minutes. This extended settling time is attributed to factors such as ambient temperature, Peltier cell dimensions, and heatsink efficiency. Figure 39 illustrates the temporal response of the temperature control system at the specified setpoint. Additionally, Figure 40 depicts the operating ranges of the Peltier cell, including variations in pulses and current against temperature changes.

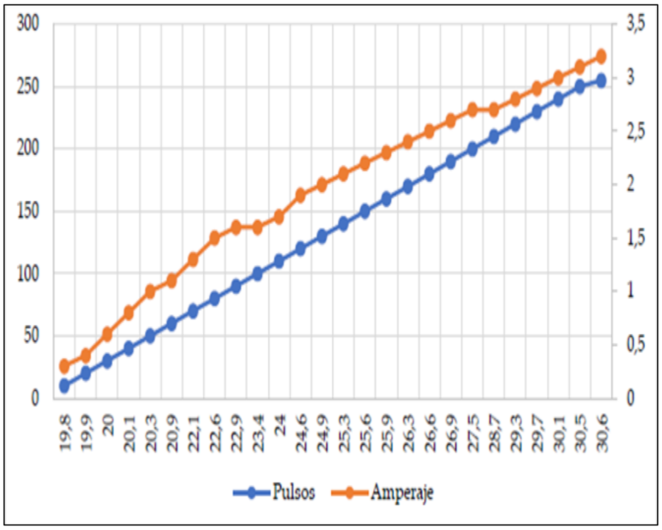

Fig. 40 Variation of pulses and amperage against temperature changes

4.1.5. Tests Performed on the Web Application

The web application was developed for real-time monitoring of environmental parameters measured by sensors connected to the Arduino. Sensor data is transmitted to the Raspberry Pi, which forwards it to a database for logging. This setup enables users to view the recorded measurements through the application interface.

A key feature of the web application is its remote restart function, which ensures a reliable and continuous flow of data even in the event of failures or data saturation. The application is accessible on any mobile device, offering flexibility and ease of use. During the tests, a temperature setpoint of 25°C and a water level setpoint of 8 cm were configured. The application reported actual measured values of 24.6°C for temperature and 7.95 cm for water level, demonstrating its precision and functionality. Additionally, the application successfully monitored other parameters, including:

- pH: 7.8
- Luminosity: 170 Lx
- Humidity: 41.4%

The web interface features a central button labelled "Reset Device." This button allows the receiver module to be remotely reset directly from the web application, further enhancing system reliability and user control.

Figure 41 shows the web application.

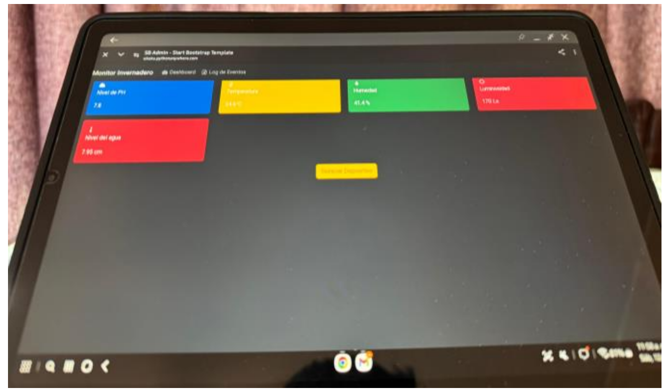

Fig. 41 Operation of the web application

4.2. Discussion

Lettuce was selected as the primary hydroponic crop due to its rapid growth, low water consumption, and adaptability to hydroponic systems. Compared to spinach and Swiss chard, it is more resilient to pH and temperature variations, reducing crop failure risks. Unlike basil, it follows a predictable growth cycle without requiring frequent pruning. Its compact root and leaf structure suit the 3D-designed hydroponic pots, ensuring efficient use of space. Its lower susceptibility to diseases and frequent harvesting make it a practical and sustainable choice for hydroponic cultivation.

Following the completion of the research project and the corresponding tests and validation of the pilot-scale prototype, several observations have been identified that could enhance the system's performance and applicability in future iterations.

The first observation concerns the use of ultrasonic sensors. These sensors often produce measurement spikes, leading to inaccuracies. For future research, adopting more precise level measurement technologies, such as hydrostatic pressure sensors, is recommended, which provide higher accuracy. The main reason for choosing this sensor is the low cost and practicality of its use. Other models considered are, for example, the differential pressure sensor MPX5010DP, which measures the pressure exerted by a column of water on its inlet; this pressure is directly proportional to the height of the water in a container, so it needs an additional mechanical design; this sensor has a price in the Peruvian market of S/ 105.00. Another option is the TFMini-S LiDAR sensor, which measures distances using laser detection technology; this sensor has a price in the Peruvian market of S/ 310.00. On the other hand, the HC-SR04 sensor is priced at S/ 8.00. Additionally, the 3D-designed hydroponic pots proved suitable for lettuce cultivation due to the plant’s limited root and leaf extension. However, the pot dimensions may need to be reconsidered and adapted for other crop species. Another noteworthy aspect is the web application's data display and update frequency, set at 500 ms. Such a rapid update rate in greenhouse monitoring systems is unnecessary, as the measured parameters typically evolve slowly. For instance, in the García Cortés et al. study, "Low-cost wireless system for monitoring temperature and relative humidity variables in a greenhouse," an update interval of 5 minutes was used, which suffices for similar applications [14].

Regarding the NRF24L01 RF modules, challenges were observed in data transmission over distances with obstacles obstructing the line of sight. To address these limitations, future implementations could consider adding RF modules using LoRaWAN technology, as demonstrated in the work by Guillermo et al., "IoT Architecture Based on Wireless Sensor Network Applied to Agricultural Monitoring: A Case Study of Cacao Crops in Ecuador." LoRaWAN technology offers extended transmission ranges of up to 10 km while maintaining low power consumption, making it an excellent alternative for agricultural applications. [4, 15]

The NRF24L01 RF module was chosen for this research due to its data rate, latency, cost, and easy installation and programming. The LoRa E220-900T22D module has a maximum data transmission rate of 62.5 Kbps with a high latency greater than 100 ms, and the NRF24L01 2000 Kbps module with a low latency of less than 5ms, which represents a noticeable difference in data transmission rate, crucial when a fast sending rate is needed for a closed-loop control system because rapid changes in the sensors must be sent immediately; in addition to the importance when monitoring in real-time. In addition, the price in the Peruvian market LoRa E220-900T22D module is S / 79.00, compared with the price of the S / 28.00 NRF24L01 module. The price difference is more than double.

The tests performed revealed certain discrepancies between the system's responses (e.g., temperature, water level, pulses, and amperage) and simulation results. These differences are attributable to environmental conditions and external factors influencing real-world data acquisition. Simulations, by contrast, represent idealized scenarios unaffected by such factors. Despite these variations, the experimental results closely align with the simulations, with only minor perturbations observed. Further testing is recommended for temperature control to identify the specific temperature ranges characteristic of the implementation environment. This analysis could help refine existing control rules or establish new ones tailored to local conditions. Additionally, uniform heat sinks are advised to ensure efficient heat transfer. Future works will explore the possibility of enabling setpoint configuration for control systems directly through the web application, further enhancing the system's versatility and user interaction.

5. Conclusion

Upon completing this research project, an automatic control system for a greenhouse was successfully implemented. Additionally, through integrating an IoT architecture, wireless monitoring of essential greenhouse parameters was achieved using RF and Wi-Fi communication. This approach ensures the creation of specific microclimates, addressing the lack of greenhouses contributing to regional food shortages. The monitored data is stored in the cloud and displayed via a web application.

The research project was validated by constructing a pilot-scale prototype, which was adapted along with the sensor and actuator parameters for the hydroponic cultivation of lettuce. During the implementation phase, the NRF24L01 radio frequency communication modules were identified as optimal for the application due to their ease of integration with the Arduino UNO development board, affordability, range, and market availability. However, alternative antennas are recommended for other applications.

The actuators selected for temperature, water level, and luminosity control include a Peltier cell, a 12 VDC water pump, and LED strips, respectively. Corresponding sensors were chosen: the DHT21 for temperature, an ultrasonic sensor for water level, the BH1750 lux meter for luminosity, and the PH4502C sensor for pH. All parameters were configured within suitable ranges for hydroponic lettuce cultivation. For future applications involving other crops, parameter adjustments will be necessary. The RF communication system was thoroughly tested using the NRF24L01 modules, ensuring reliable data transmission from the transmitter to the receiver. The transmitted data was visualized in the web application via a Wi-Fi connection using a Raspberry Pi 4B development board. Thermal insulation is recommended for the RF module to maintain optimal performance at low temperatures. Additionally, twisted wires should connect the RF module to the Arduino, and a 100 nF ceramic capacitor is suggested to stabilize the 3.3 V power supply.

Future improvements include replacing the ultrasonic sensor for water level measurement, which introduces noise due to signal peaks. External factors observed during testing differed slightly from simulations but did not significantly impact results, validating the system’s reliability and potential for further refinements by incorporating these factors in future simulations. The free PythonAnywhere platform was used to store the data, which requires a service restart every three months to guarantee its functionality, as it is free to use. This does not affect the functioning of the system or the SQLite database. This service restart is only to validate that the application is in use. Additionally, a restart mechanism was incorporated through bidirectional communication, and a button was added to the web application. In the control phase, a mathematical model of the tank system was developed, and a PI controller for water level management was tuned using

Simulink's PID Tune tool. Feedback was provided via an ultrasonic level sensor. An open-loop system was employed for brightness control, using LED strips programmed to operate 11 hours per day. Brightness monitoring was performed using the BH1750 lux meter, achieving an approximate reading of 288 lux. A Peltier cell was configured to operate continuously for temperature control, maintaining a temperature range of 20°C at low power and 25°C at maximum power. This range aligns with the optimal growth conditions for lettuce, which span from 15°C to 35°C.

Acknowledgments

The authors thank the National University of San Agustín, their alma mater, for providing the knowledge necessary to carry out this project.

References

[1] National Disaster Risk Management Plan - PLANAGERD 2014-2021, National Center for Disaster Risk Assessment, Prevention, and Reduction (CENEPRED), 2021. [Online]. Available: https://sigrid.cenepred.gob.pe/sigridv3/documento/417

[2] Jorge Luis Chávez Cresta, *Statistical Compendium 2017*, National Institute of Civil Defense, gob.pe, 2017. [Publisher Link]

[3] A. Perea Flores, '6-Friaje_Heladas', Lima, 2018. [Online]. Available: https://www2.congreso.gob.pe/sicr/cendocbib/con5_uibd.nsf/322B8DD56F06DD6E0525832700584752/$FILE/6-Friaje_Heladas.pdf

[4] Juan Carlos Guillermo et al., "IoT Architecture Based on Wireless Sensor Network Applied to Agricultural Monitoring: A Case of Study of Cacao Crops in Ecuador," *International Conference of Information and Communication Technologies for Adapting Agriculture to Climate Change*, Cali, Colombia, pp. 42-57, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Leonardo Fernández Díaz et al., "Technological Architecture for Monitoring Environmental Variables in a Museum," *Colombian Journal of Advanced Technologies*, vol. 2, no. 38, pp. 1-8, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Haozhe Zhang et al., "Lorawan Based Internet of Things (Iot) System for Precision Irrigation in Plasticulture Fresh-Market Tomato," *Smart Agricultural Technology*, vol. 2, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Jalalu Guntur et al., "An Automatic Irrigation System Using IOT Devices," *Materialstoday: Proceedings*, vol. 68, pp. 2233-2238, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jissie Vaquero López, and Carlos Luis Milian del Valle, "Nonlinear Control Using Approximate Linearization Applied in the Nonlinear System Position of a Ring on a Rotating Ring," *Cuban Journal of Computer Science*, vol. 10, no. 2, pp. 89-99, 2016. [Google Scholar] [Publisher Link]

[9] Rafael J. García et al., "Design of a Fuzzy Control Strategy Applied to the Ultra-Freezing Process of Food," *Ingeniare. Chilean Engineering Journal*, vol. 25, no. 1, pp. 70-84, 2017. [Google Scholar]

[10] Chiara Bersani et al., "Internet of Things Approaches for Monitoring and Control of Smart Greenhouses in Industry 4.0," *Energies*, vol. 15, no. 10, pp. 1-30, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] R. Jimenez Moreno, O. Aviles Sanchez, and O.L. Ramos Sandoval, "Analysis of the Implementation of a Fuzzy Controller on Different Hardware Architectures," *Neogranadine Science and Engineering*, vol. 23, no. 1, pp. 77-87, 2013. [Google Scholar]

[12] Yongchao Song, Jiping Bi, and Xuan Wang, "Design and Implementation of Intelligent Monitoring System for Agricultural Environment in IoT," *Internet of Things*, vol. 25, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Dawid Witczak, and Sabina Szymoniak, "Review of Monitoring and Control Systems Based on Internet of Things," *Applied Sciences*, vol. 14, no. 19, pp. 1-27, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] José de Jesús García Cortés et al., "Low-Cost Wireless System for Monitoring Temperature and Relative Humidity in a Greenhouse," *Ingeniantes*, vol. 2, no. 2, pp. 61-67, 2019. [Google Scholar]

[15] Mohammad Mansour et al., "Internet of Things: A Comprehensive Overview on Protocols, Architectures, Technologies, Simulation Tools, and Future Directions," *Energies*, vol. 16, no. 8, pp. 1-39, 2023. [CrossRef] [Google Scholar] [Publisher Link]