Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 9, No. 3, June, 2025

IoT System with ESP32 for Smart Drip Irrigation and Climate Monitoring in Greenhouses

Joseph J. Correa-Quiroz [1], Max A. Toribio-Barrueto [1], Cristian Castro-Vargas [1*]

[1] *Facultad de Ingeniería, Universidad Privada del Norte, Lima-Trujillo, Perú.*

Abstract

The depletion of water resources and the need for sustainable agricultural practices require innovative technological solutions. This study develops an IoT-based smart drip irrigation and climate monitoring system for greenhouses using the ESP32 microcontroller. The methodology implements DHT11 sensors for temperature and humidity, GUVA-S12SD for UV radiation, capacitive soil moisture sensors, and HC-SR04 for water level measurement. Real-time data is displayed on an LCD screen and transmitted to the Arduino Cloud, enabling remote monitoring and control. Field tests showed a 35% reduction in water consumption compared to traditional methods, improving crop environmental conditions and reducing operating costs. The system operates in automatic and manual modes, adapting to various climatic conditions and user needs. The main innovation lies in its optimized water use efficiency through smart drip irrigation, ensuring precise humidity control and minimizing waste. Furthermore, its scalability allows integration with renewable energy sources, increasing its autonomy and sustainability. This approach fosters climate-resilient agriculture, aligned with the Sustainable Development Goals (SDGs), by promoting water conservation and efficient resource use.

Keywords:

Greenhouse Management;
Internet of Things (IoT);
Weather Irrigation;
Automated Irrigation;
Arduino Cloud; Esp32;
Drip irrigation.

Article History:

Received:	21	December	2024
Revised:	19	April	2025
Accepted:	24	April	2025
Published:	01	June	2025

1- Introduction

Climate change is a global phenomenon that significantly impacts food security and water resource management, particularly in agriculture [1]. In Latin America, prolonged droughts have reduced water availability, affecting agricultural productivity and promoting rural migration [2]. In Chile, for example, several regions have experienced severe droughts for more than a decade, drastically reducing agricultural production [3]. Among the main consequences of climate change are extreme temperatures, changes in rainfall patterns, and a reduction in water available for irrigation [4]. Vegetable and root crops, susceptible to climate variability, have shown declines in yield due to water and heat stress [5]. Furthermore, in high Andean areas, glacial retreat and soil erosion have aggravated the vulnerability of key agricultural ecosystems, affecting essential crops such as potatoes and corn [6]. In response to these challenges, climate-smart agriculture has emerged as a key strategy to improve water use efficiency and reduce greenhouse gas emissions [7].

Despite these advances, conventional irrigation systems still suffer from deficiencies in terms of water efficiency and adaptation to dynamic environmental conditions [8]. Irrigation automation using IoT technologies has proven to be a viable alternative for optimizing water use and improving agricultural productivity [9]. In particular, drip irrigation has been widely recognized as one of the most efficient strategies for agrarian irrigation, reducing water waste and ensuring

* **CONTACT**: cristian.castro@upn.pe

DOI: http://dx.doi.org/10.28991/ESJ-2025-09-03-01

even distribution to crop roots [10]. Several studies have addressed irrigation optimization using IoT. In Pérez-Baca et al. [11] study, an automated irrigation system was designed using humidity sensors and IoT technology to improve domestic gardens' water use efficiency. Another study proposed an irrigation system based on NodeMCU and the Blynk platform, optimizing water consumption through real-time monitoring [12]. In Rahim et al. [13] study, an irrigation system for taro yam crops was developed, achieving a 32.5% reduction in water consumption through automated soil moisture control. In the Peruvian context, an automatic irrigation system was implemented using an Arduino Uno and soil moisture sensors, achieving a 75% reduction in water consumption in dry soils and a 76.5% reduction in wet soils [14]. Similarly, in Canlas et al. [15] work, an IoT system was designed for irrigating date palm trees, optimizing water use through learning algorithms based on historical cases. Satra et al. [16] addressed the inefficient management of manual irrigation, which leads to high water consumption and labor demand. They designed an automatic irrigation system based on Arduino Uno, humidity sensors, and an SPDT relay to control water pumps. The results showed a significant reduction in water waste, promoting sustainable and low-cost practices in agricultural and gardening applications. Finally, in Abouelmehdi et al. [17] study, an environmental monitoring system based on IoT and renewable energy was explored, improving energy efficiency and agricultural productivity through the use of programmable sensors and controllers.

Given this context, this study proposes an ESP32-based IoT system for smart drip irrigation and climate monitoring in greenhouses. The developed solution integrates temperature, air humidity, UV radiation, and soil moisture sensors, whose data are processed in real-time and sent to the Arduino Cloud platform for remote monitoring and control. Unlike other approaches, this system operates in dual mode (automatic and manual), allowing dynamic adjustments based on the specific greenhouse conditions. This approach optimizes water consumption, reduces operating costs, and allows for scalability. It integrates with renewable energy sources such as solar panels to improve its autonomy. In this way, the research contributes to developing sustainable agricultural practices aligned with the Sustainable Development Goals (SDGs) regarding efficient water management and sustainable agricultural production.

The article's structure is as follows: Section 2 presents the theoretical approach, Section 3 details the methodology, Section 4 analyzes the results and discussions, and Section 5 presents conclusions and recommendations for future research.

2- Theoretical Approach

The design and implementation of the proposed system are based on various theories and technological approaches that support the integration of the Internet of Things (IoT) into agricultural automation, with a special emphasis on drip irrigation. The main theoretical frameworks underlying this study are presented below.

2-1- Precision Agriculture and Drip Irrigation

Precision agriculture is based on optimizing agricultural resource use through advanced technologies such as sensors, artificial intelligence (AI), and data analytics, aiming to improve crop efficiency and productivity [18]. Within this paradigm, drip irrigation has established itself as an efficient technique for agricultural irrigation, allowing for uniform distribution of water directly to the root zone of plants and minimizing losses due to evaporation and runoff [19]. Recent studies have shown that integrating IoT with drip irrigation systems optimizes water use by automating water flow based on real-time data, reducing water consumption by up to 40% compared to conventional methods [20].

2-2- Internet of Things (IoT) in Agriculture

Agricultural IoT has revolutionized irrigation management by enabling the connection of sensors, embedded devices, and cloud platforms for remote monitoring and intelligent crop automation [21]. This technology facilitates the real-time collection and analysis of environmental variables such as soil moisture, temperature, UV radiation, and water level, providing more precise control and reducing manual intervention [22]. In the proposed system, sensors collect key data and send it to the cloud via ESP32, enabling remote management from mobile devices. This approach enables farmers to adjust irrigation parameters based on weather conditions and crop water needs, improving sustainability and operational efficiency [23].

2-3- Smart Irrigation Systems and Control Algorithms

Intelligent irrigation systems seek to maximize water efficiency by applying water only, when necessary, based on information obtained from environmental sensors [24]. The literature has shown that implementing automatic control algorithms can reduce water waste and improve crop yield by up to 30%, compared to manual irrigation [25]. More advanced approaches use artificial intelligence (AI) and machine learning-based models to predict soil water requirements and optimize irrigation timing [26]. This study uses predefined moisture thresholds to activate or deactivate the drip irrigation system based on the collected data, ensuring efficient water use.

2-4- Cloud Computing and Remote Accessibility

Cloud computing enables the remote storage, processing, and visualization of agricultural data in real-time, optimizing irrigation management and reducing the need for expensive local infrastructure [27]. This system transmits the collected information to the Arduino Cloud, where users can monitor and control irrigation using a mobile app or web interface. This approach improves the ability to respond to sudden weather changes and facilitates the implementation of automated irrigation strategies based on historical data and predictive analytics [28].

2-5- Integration of Renewable Energy in Agricultural Automation

Using renewable energy sources, such as solar panels, represents a key strategy for improving the sustainability and autonomy of intelligent irrigation systems in greenhouses [29]. Research has shown that combining IoT and solar energy can reduce energy consumption by up to 60% compared to conventional grid-connected systems [30]. This study considers the possibility of integrating solar panels into future versions of the system. This would allow the system to operate in rural areas with limited access to electricity, increasing its applicability in diverse agricultural settings.

2-6- Cybersecurity and Data Protection in Agricultural IoT

Transmitting agricultural data to the cloud requires the implementation of security and encryption mechanisms due to the risks associated with cyberattacks and unauthorized access [31]. This system employs secure encryption and authentication protocols, protecting crop information and preventing disruptions to irrigation automation. Recent studies have shown that incorporating advanced security algorithms into agricultural IoT systems can reduce vulnerability to cyberattacks by 70%, ensuring data integrity in critical environments [32].

3- Research Methodology

As can be seen, the cascade method will be used, as shown in Figure 1. We will start with selecting the work area and identifying the variables essential for the research. Materials will be identified and specified in the second block, and the electronic components and sensors necessary to implement the system will be defined. In the third phase, the circuit design will be developed, which will include the schematic design, the PCB design, and the creation of a 3D model of the circuit to validate the layout of the components. Next, the algorithm and programming will be developed, incorporating the applied mathematical equations in the code for monitoring and controlling the variables. Finally, the model will be designed, which involves the construction of a physical model of the project to validate the integration of all the elements in a controlled environment.

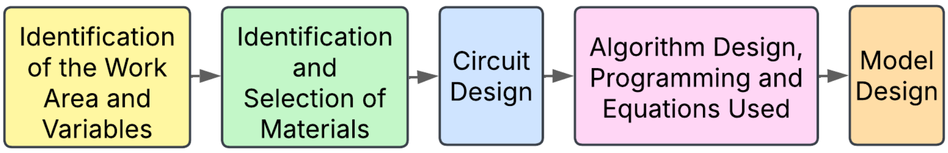


Figure 1. Block Diagram of Project Methodology

3-1- Identification of the Work Area

The area selected for the implementation of the irrigation and climate monitoring system is the district of Trujillo, Peru, as shown in Figure 2, a region characterized by an arid and semi-arid climate that presents significant challenges for agriculture due to the limited availability of water. Agricultural areas, essential for production, face increasing pressures due to urban expansion and the scarcity of water resources, which makes the use of advanced technologies necessary for optimizing irrigation and controlling the microclimate in greenhouses. This base map of the Trujillo district, extracted from Google Maps, delimits the geographical contours of the study area, allowing areas with agricultural potential to be identified. In particular, green areas are observed that, although not explicitly categorized on this map, suggest the presence of agricultural land or open spaces. These areas are relevant for future studies that evaluate their feasibility for the installation of greenhouses and the automation of irrigation, thus contributing to agricultural sustainability in water scarcity.

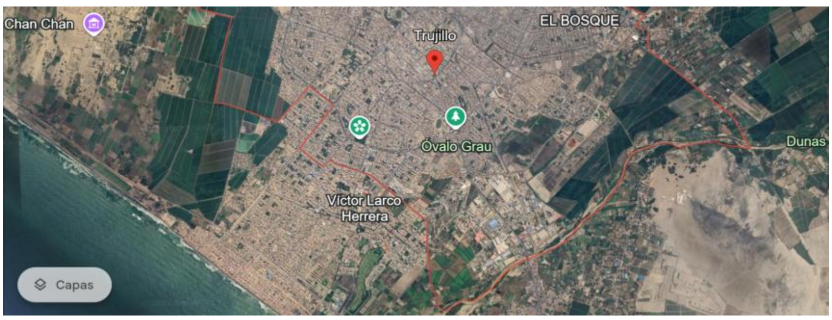


Figure 2. Thematic Map of Vegetation Cover in the Province of Trujillo

3-2-Identification and Selection of Materials

In this project, the ESP32, connected via Wi-Fi to the Arduino Cloud, acts as the system's brain, monitoring various systems [33] and automatically adjusting irrigation in a greenhouse. In addition to sensors that measure critical variables such as air temperature and humidity (via the DHT11) and UV radiation (via the GUVA-S12SD), the ESP32 is connected to a soil moisture sensor to determine irrigation needs. An ultrasonic sensor (HC-SR04) is also employed to measure the water level in the irrigation tank, ensuring that sufficient water is available. The system is completed by a relay controlling the irrigation pump and an LCD display showing key real-time information. The ESP32 processes the data from the sensors and triggers the relay to adjust irrigation based on the needs of the crop. See Table 1 and Figure 3.

Table 1. Technical specifications of the ESP32 DevKit V1

Attribute	Specifications
CPU	Dual Core Tensilica Xtensa LX6 (32 Bit)
Operating voltage	3.3 V - 5 V
Input / Output	30 pin GPIO/ADC/DAC/PWM/SPI/I2C
Wi-Fi	802.11 b/g/n/e/i (2.4 GHz)
Bluetooth	4.2 (BLE)

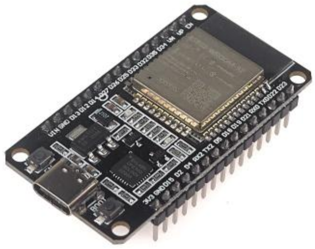


Figure 3. ESP32 DevKit V1

The DHT11 sensor, a low-consumption digital device, measures two critical variables: relative humidity and temperature. It uses a capacitive sensor to detect changes in ambient moisture, measuring variations in the capacitance of humid air. In addition, it incorporates a thermistor that changes its resistance depending on temperature, allowing it to provide accurate readings within a range of 0 to 50 °C [34], as shown in Table 2 and Figure 4.

Table 2. Technical specifications of the DHT11

Attribute	Specifications
Operating voltage	3 V - 5.5 V
Humidity accuracy	±5% RH
Temperature range	0 °C - 50 °C
Connection pins	Vcc - Gnd - Data (Digital)

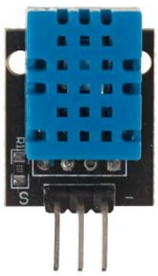


Figure 4. DHT11 sensor

The GUVA-S12SD sensor is designed to detect ultraviolet (UV) radiation in the wavelength range from 240 nm to 370 nm, covering both the UV-B and UV-A spectrum [35]. This sensor converts the UV radiation it receives into an analog signal that can be read by a microcontroller (such as an Arduino or ESP32) through an analog-to-digital converter (ADC). The analog output is proportional to the detected UV intensity, which allows for obtaining a precise value of the levels of ultraviolet radiation present, as seen in Table 3 and Figure 5.

Table 3. Technical specifications of the GUVA-S12SD

Attribute	Specifications
Operating voltage	2.7 V - 5 V
Forward current	1 mA
Operating temperature	-30 °C to 85 °C
Spectral detection range	240 - 370 nm
Connection pins	Vcc - Gnd - Data (Analog)

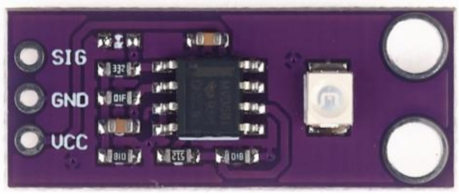


Figure 5. GUVAS12-SD sensor

The Capacitive Soil Moisture Sensor V1.2 uses capacitive measurement to detect moisture levels in the soil. Unlike resistive sensors, it does not have parts exposed to the ground that corrode over time, which gives it more excellent durability. The sensor measures the soil's ability to store electrical charge, which varies depending on moisture content [36]. The output is an analog value that reflects the moisture level: when the soil is drier, the reading is higher, and when the soil is wetter, the reading is lower; its characteristics are illustrated in Table 4 and Figure 6.

Table 4. Technical specifications of the capacitive soil moisture sensor V1.2

Attribute	Specifications
Operating voltage	3.3 V - 5 V
Operating current	5 mA
Connection pins	Vcc - Gnd - Data (Analog)

Figure 6. Capacitive soil moisture sensor V1.2

The HC-SR04 ultrasonic sensor emits ultrasonic pulses through its "Trigger" pin, which bounce off an object and are received back at the "Echo" pin. The sensor measures the time it takes for the pulse to return and calculates the distance using a respective formula, considering that the speed of sound is approximately 343 m/s under normal conditions. The sensor can measure distances between 2 cm and 400 cm with an accuracy of ±3 mm, operating at a frequency of 40 kHz [37]. Its output is a digital signal that microcontrollers can process for applications such as obstacle detection or level measurement. The characteristics are identified in Table 5 and Figure 7.

Table 5. Technical specifications of the ultrasonic sensor

Attribute	Specifications
Operating voltage	5 V
Operating current	15 mA
Measuring range	2 cm to 400 cm
Working Frequency	40 khz
Connection Pins	Vcc - Gnd -Trigger (Digital) - Echo (Digital)

Figure 7. HC-SR04 sensor

Table 6 and Figure 8 show the qualities of the water pump that uses an electric motor that drives an impeller to move the liquid from the pump inlet to a pressure outlet, which raises the liquid to a maximum height of 3 meters. This pump can move 240 liters per hour, making it suitable for water circulation applications, irrigation systems, aquariums, and water drainage in various situations [38]. It is quiet (less than 40 dB) and can operate with liquids at temperatures up to 60°C, making it versatile for different types of fluids and environments. It has a long lifespan of up to 30,000 hours, making it an efficient and durable option.

Table 6. Technical specifications of the water pump.

Attribute	Specifications
Operating voltage	12 V
Noise level	<40 dB
Fluid temperature	0 °C to 60 °C
Working Liquid	Water, oil, gasoline

Figure 8. Water pump

Table 7 and Figure 9 show the characteristics of the single-channel relay that allows the control of high-power devices (such as motors or pumps) using a low-voltage signal from a microcontroller, such as an ESP32. When the input pin (IN) receives a control signal (5V), the relay switches, allowing current to flow between the COM and NO (Normally Open) terminals [39]. If the signal goes off, the circuit is interrupted, returning to the NC (Normally Closed) position. This is useful for automating processes such as turning devices on and off in irrigation systems.

Table 7. Technical specifications of the relay

Attribute	Specifications
Operating voltage	5 V DC
Maximum current	10 A (NO), 5 A (NC)
Maximum load current	10 A at 250 VAC / 15 A at 125 VAC
Connection pins	Vcc, Gnd, In, NO, Com, NC

Figure 9. Relay

The 16x2 LCD screen in Figure 10, controlled by the HD44780 controller, can display up to 32 characters in two rows of 16 characters each [40]. Its operation is based on sending commands and data from the microcontroller, which is responsible for managing the position of the cursor and the content to be displayed. The backlight makes it easy to read in low-light environments, and its 4- or 8-bit interface makes it compatible with various microcontrollers, such as the ESP32. Powered at 5V, according to Table 8, this screen is ideal for viewing information simply and effectively.

Table 8. Technical specifications of the LCD 16×2

Attribute	Specifications
LCD size	16×2 characters
Operating voltage	5 V
Operating current	20 mA (with blacklight on)
Used Pins	VSS, VDD, RS, E, D4, D5, D6, D7

Figure 10. LCD 16×2

3-3- Circuit Design

In the circuit design of Figure 11, the key stages from data collection to action execution are identified, efficiently integrating sensors, controllers, actuators, and communication systems. To begin with, in the Acquisition Stage, data is collected from various sensors, where the DHT11 measures the temperature and humidity of the air, the HC-SR04 detects the water level in the tank, the GUVA-S12SD measures ultraviolet radiation, and the Capacitive soil moisture sensor monitors soil moisture. All this information is processed in the Control Stage by the ESP32, which acts as the central controller and determines the necessary actions based on the readings received. In the Actuator Stage, the relay activates or deactivates the water pump, depending on the humidity levels detected in the soil, and a power supply ensures the operation of these components. Information is presented in real-time on a 16x2 LCD in the Visualization Stage, including the Arduino Cloud platform, where the user can view the system's status through a web interface. Finally, in the Communication Stage, the ESP32 transmits data to the cloud via Wi-Fi, allowing the user to monitor and control the system remotely from any device with internet access.

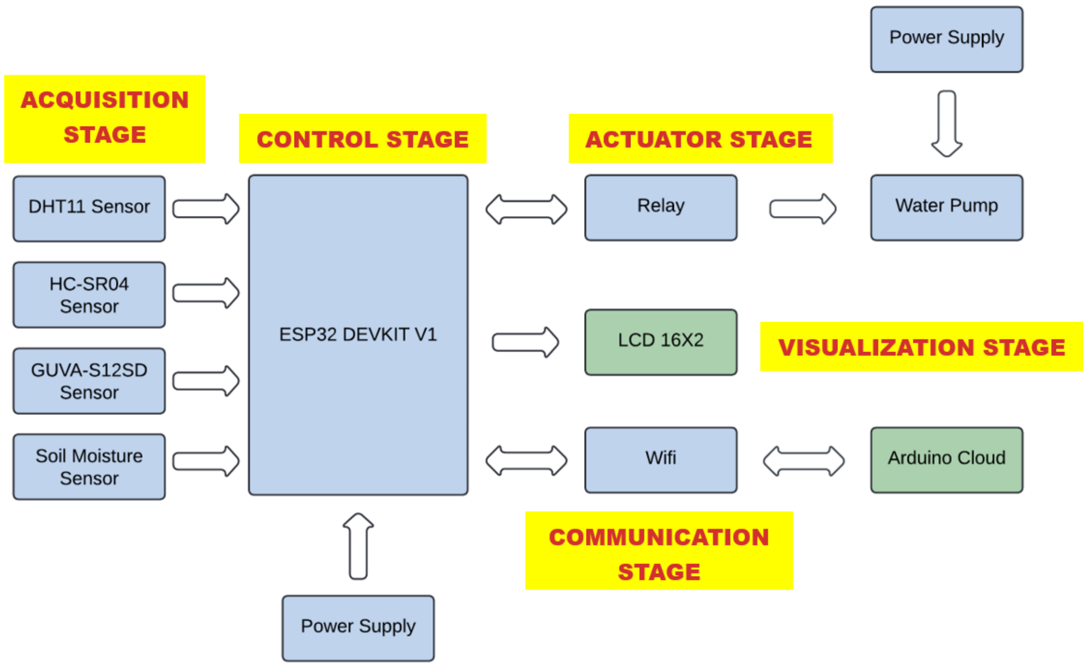


Figure 11. The architecture of the project in a block diagram

Figure 12 shows the detailed circuit schematic, highlighting the connections between the ESP32 and the different sensors. This schematic provides a more technical view of how the signals are distributed, ensuring that each sensor and actuator is correctly wired to the appropriate pins on the microcontroller to ensure smooth and accurate system operation.

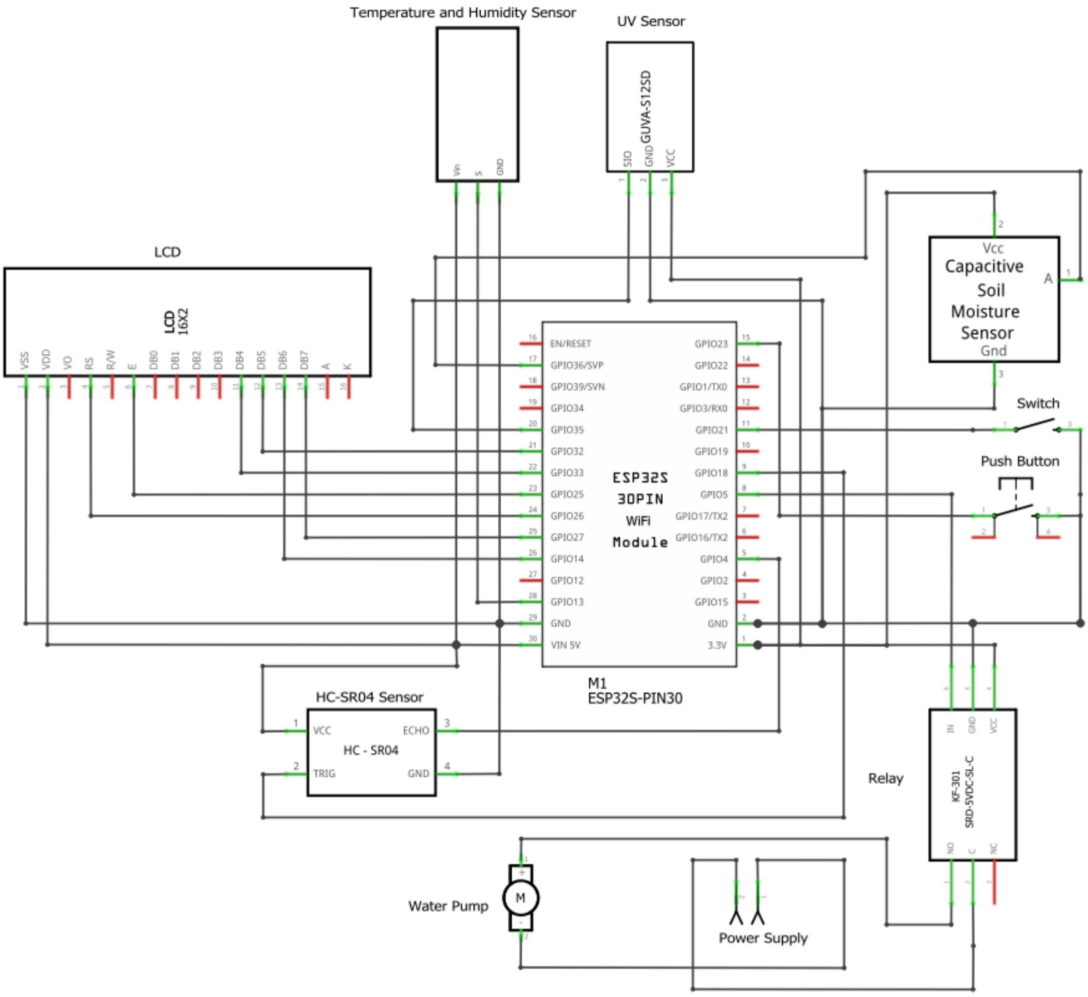


Figure 12. Schematic circuit of the proposed system

Figure 13 illustrates the physical circuit diagram, highlighting the connections of the sensors, water pump, relay, and LCD screen with the ESP32. This allows a visual understanding of how all the components are integrated into the system. The connections are clearly distributed, ensuring proper communication between the modules.

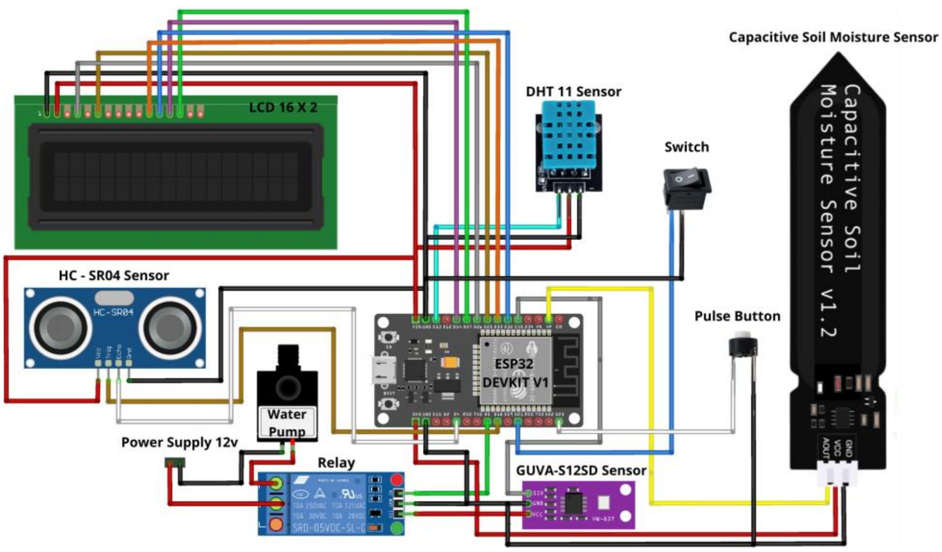


Figure 13. Physical block diagram of the circuit

Figure 14 shows the design of the printed circuit board (PCB). The PCB facilitates the integration and connection of all components in a compact manner, optimizing space and reducing errors in manual connections. The design allows the circuit to be replicated more efficiently and professionally.

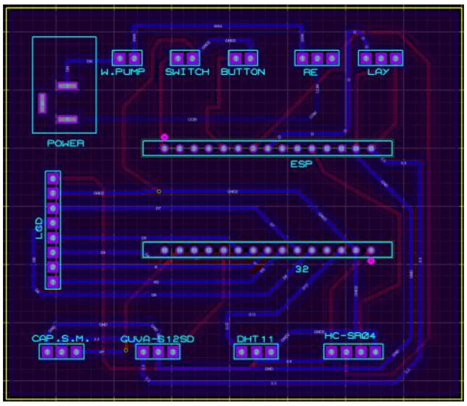


Figure 14. PCB design of the proposed system

Figure 15 shows the circuit's 3D simulation, showing how the components will be physically on the board. This view provides a clear picture of the assembly and helps anticipate possible adjustments in the placement of the elements to improve functionality and accessibility.

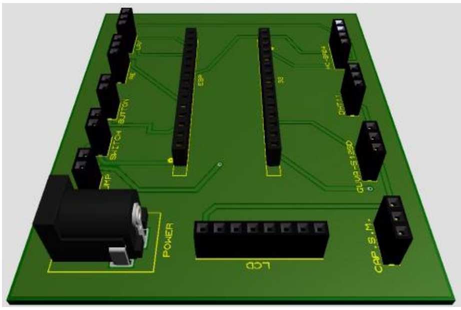


Figure 15. 3D simulated circuit for the proposed system

3-4- Algorithm Development and Programming

The flow chart shows in Figure 16 the greenhouse monitoring and control system sequence. First, the system initializes, connects to the cloud, and takes sensor readings. In automatic mode, the pump turns on if the water level exceeds zero, depending on the humidity. The user can turn the pump on or off manually using a button. Finally, the LCD screen displays updated system data.

Figure 17 shows the code section, where variables and pins necessary to operate an irrigation system controlled through Arduino IoT Cloud are configured, allowing remote updates of critical parameters such as humidity, water level, temperature, and UV exposure. The defined pins (TRIG_PIN and ECHO_PIN) are for the ultrasonic sensor that measures the distance to the water, which is crucial for determining the tank level. The relay pin controls a relay that activates or deactivates the water pump depending on the soil moisture and the operating mode. The switchPin and buttonPin pins manage the transition between automatic and manual modes. State variables such as manual mode, button state, and button pressed are essential for control logic that determines how and when actuators are activated based on sensor inputs and user interactions.

Figure 18 shows the source code corresponding to the initial configuration of an automated environmental monitoring and control system based on Arduino. This system integrates several key libraries, such as LiquidCrystal.h for interfacing with an LCD, thingProperties.h for property management on the IoT Cloud platform, and DHT.h for reading temperature and humidity data. The code specifies the pin and DHT sensor type used and then initializes the sensor and the LCD using the assigned pins. The setup() method configures the serial communication. It defines the pins corresponding to various devices, such as the relay, the ultrasonic sensor, and the buttons to operate manually or automatically. Finally, the system establishes the connection with the Arduino IoT Cloud, allowing the registration of properties that can be monitored and controlled remotely.

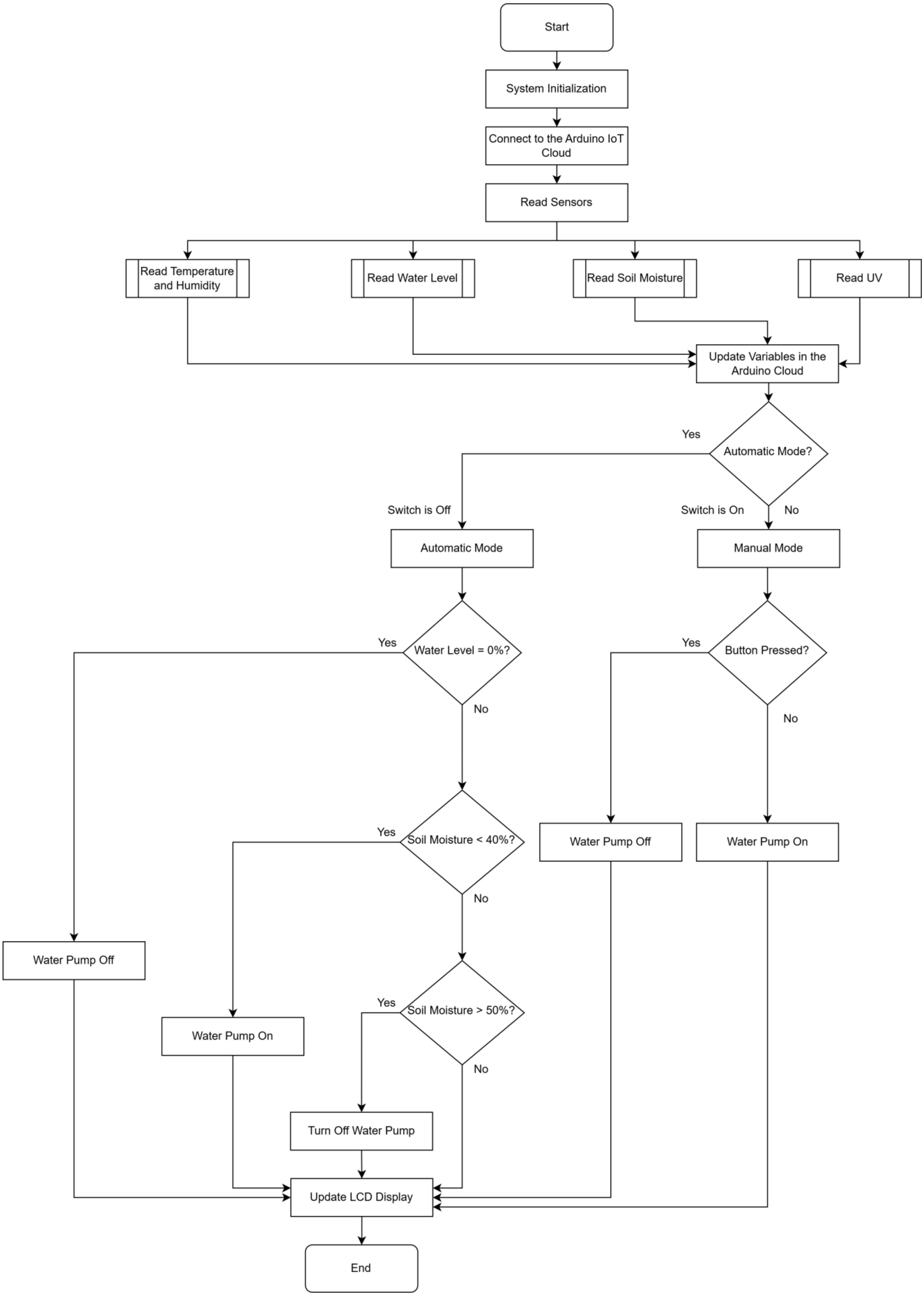


Figure 16. Flowchart of system operation

```
  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the Thing

  int humidity;
  int soilmoisture;
  int ultra;
  int waterlevel;
  CloudTemperature temperature;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch.
*/
#define TRIG_PIN  18
#define ECHO_PIN  4
#define relay 5
const int switchPin = 21;
const int buttonPin = 23;

int manualMode = LOW;
int ledState = LOW;
int buttonState = LOW;
int lastButtonState = LOW;
bool buttonPressed = false;
bool lastButtonPressed = false;

const int AirValue = 2000;
const int WaterValue = 1120;
int soilMoistureValue = 0;
int soilMoisturePercent = 0;

const int uvPin = 35;
float referenceVoltage = 3.3;
int maxAdcValue = 4095;
int uvIndex = 0;

#define DISTANCIA_LLENO 4.32
#define DISTANCIA_VACIO 12
```

Figure 17. Setting up variables and pins for the irrigation system controlled via Arduino IoT Cloud

```
#include <LiquidCrystal.h>
#include "thingProperties.h"
#include "DHT.h"
#define DHTpin 13
#define DHTTYPE DHT11

DHT dht(DHTpin, DHTTYPE);
LiquidCrystal lcd(26, 25, 33, 32, 14, 27);
unsigned long previousMillis = 0;
const long interval = 500;
unsigned long previousDHTMillis = 0;
const long dhtInterval = 2000;

void setup() {
  Serial.begin(9600);
  delay(1000);
  pinMode(relay, OUTPUT);
  digitalWrite(relay, HIGH);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  lcd.begin(16, 2);
  lcd.print("Initializing...");

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  ArduinoCloud.addProperty(soilmoisture, READ, ON_CHANGE, NULL);
  ArduinoCloud.addProperty(waterlevel, READ, ON_CHANGE, NULL);

  pinMode(switchPin, INPUT_PULLUP);
  pinMode(buttonPin, INPUT_PULLUP);


  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
  pinMode(36, INPUT);
}
```

Figure 18. Initial Setup of Environmental Monitoring System with Arduino

This section of the code corresponding to Figure 19 is the loop() function of an Arduino-based environmental control and monitoring system, which constantly updates sensor data and handles the control logic. First, the state of the switch in is checked to determine whether the system should operate automatically or manually, and the status is printed on the serial console. Pulses are sent to the TRIG_PIN pin to activate the ultrasonic sensor, which measures the water level in a tank, and the distance is calculated from the time it takes for the echo captured by ECHO_PIN to return. This distance is transformed into a percentage water level by mapping it between predefined values representing the empty and full tank. In addition, the value of a UV sensor connected to the vein is read, converting the analog reading into voltage and then into a UV index, which is also printed through the serial console. Finally, the DHT sensor functions are used to read the current humidity and temperature of the environment, as well as updated values displayed on the console.

```
void loop() {
  ArduinoCloud.update();

  manualMode = digitalRead(switchPin);
  Serial.print("Manual Mode: ");
  Serial.println(manualMode ? "Automatic" : "Manual");

  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH);
  int distance = duration * 0.034 / 2;
  waterlevel = map(distance, DISTANCIA_VACIO, DISTANCIA_LLENO, 0, 100);
  waterlevel = constrain(waterlevel, 0, 100);

  // Leer UV
  int uvReading = analogRead(uvPin);
  float uvVoltage = (uvReading * referenceVoltage) / maxAdcValue;
  uvIndex = map(uvVoltage * 10, 0, 33, 0, 11);
  uvIndex = constrain(uvIndex, 0, 11);

  Serial.print("Valor ADC: ");
  Serial.print(uvReading);
  Serial.print("  - Voltaje UV: ");
  Serial.print(uvVoltage);
  Serial.print("V  - Índice UV: ");
  Serial.println(uvIndex);

  int hm = dht.readHumidity();
  Serial.print("Humidity ");
  Serial.println(hm);

  int temp = dht.readTemperature();
  Serial.print("Temperature ");
  Serial.println(temp);
  humidity = hm;
  temperature = temp;
  ultra = uvIndex;
```

Figure 19. **Operation of the main loop in an environmental monitoring system with Arduino**

This code segment in Figure 20 is responsible for controlling logic in an irrigation system's automatic and manual modes. In automatic mode, it first assesses the water level; if empty, the pump is deactivated to prevent dry running. Then, it checks the soil moisture: it activates the pump if the humidity is below 40% to irrigate dry soil and turns it off if it is above 50% to prevent over-watering. In manual mode, the pump is controlled directly by a button: a change of state on the button toggles the pump operation, allowing manual control.

```
soilMoistureValue = analogRead(36);
Serial.println(soilMoistureValue);
soilMoisturePercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);
soilMoisturePercent = constrain(soilMoisturePercent, 0, 100);
soilmoisture = soilMoisturePercent;

if (manualMode == HIGH) {
  Serial.println("Modo Automático Activo");

  if (waterlevel <= 0) {
    Serial.println("Nivel de agua 0%, bomba apagada");
    digitalWrite(relay, HIGH);
  } else {
    if (soilmoisture < 40) {
      Serial.println("Seco, bomba encendida");
      digitalWrite(relay, LOW);
    } else if (soilmoisture > 50) {
      Serial.println("Mojado, bomba apagada");
      digitalWrite(relay, HIGH);
    }
  }

} else {

  buttonState = digitalRead(buttonPin);

  if (buttonState == LOW && lastButtonState == HIGH) {
    buttonPressed = !buttonPressed;
    Serial.println("Botón presionado, cambiando estado del LED");
    Serial.println(buttonPressed ? "Botón Presionado - relay Encendido" : "Botón Soltado - relay Apagado");
    delay(50);
  }
  lastButtonState = buttonState;

  if (buttonPressed){
    digitalWrite(relay, HIGH);
  } else{
    digitalWrite(relay,LOW);
  }
}
```

Figure 20. **Code segment for automatic and manual control of the system**

Figure 21 shows the code fragment that prints the current soil moisture and tank water level values to the serial console, providing a simple textual user interface for monitoring the status of the irrigation system. In the case of soil moisture, the code checks whether it is at its maximum (100%) or minimum (0%) capacity, respectively printing "100%" or "0%, pump on". If the value is between these extremes, it simply displays the current percentage. It prints the distance measured by the ultrasonic sensor and the calculated tank fill percentage for water level, providing clear and continuous feedback on the tank conditions.

```
if (soilmoisture >= 100) {
  Serial.println("100 %");
} else if (soilmoisture <= 0) {
  Serial.println("0 %, bomba encendida");
} else if (soilmoisture > 0 && soilmoisture < 100) {
  Serial.print(soilmoisture);
  Serial.println("%");
}
Serial.print("Distancia: ");
Serial.print(distance);
Serial.print(" cm - Porcentaje de llenado: ");
Serial.print(waterlevel);
Serial.println(" %");
```

Figure 21. Code segment for serial monitoring of soil moisture and water level in irrigation system

Figure 22 shows the code snippet that checks whether enough time has passed since the last update (based on the predefined interval) to minimize LCD flickering or overload. If so, it clears the display and sets the cursor to display data for temperature ("T:"), humidity ("H:"), UV index ("UV:"), soil moisture ("S:"), water level ("L:"), and the current operating mode (auto "A" or manual "M"). In manual mode, it also displays whether the button is pressed to turn the pump on or off.

```
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
  previousMillis = currentMillis;

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("T:");
  lcd.print(temp);

  lcd.setCursor(0, 1);
  lcd.print("H:");
  lcd.print(hm);

  lcd.setCursor(5, 0);
  lcd.print("UV:");
  lcd.print(ultra);

  lcd.setCursor(11, 0);
  lcd.print("S:");
  lcd.print(soilmoisture);

  lcd.setCursor(5, 1);
  lcd.print("L:");
  lcd.print(waterlevel);

  lcd.setCursor(11,1);
  lcd.print(manualMode ? "A" : "M");

  if (manualMode == LOW) {
    lcd.setCursor(13, 1);
    lcd.print(buttonPressed ? "OFF" : "ON");
  } else {
    lcd.setCursor(13, 1);
    lcd.print("   ");

}
  delay(500);
}
}
```

Figure 22. Code segment to update and display data on the system LCD screen

3-5- Equations

This formula converts the analog signal from the GUVA-S12SD sensor into a voltage value proportional to the intensity of the detected UV radiation. The sensor generates a voltage that varies from 0 V to 3.3 V, depending on the incident UV radiation. Multiplying the ADC reading, which represents the digitized value of the sensor's analog signal, by the system reference voltage, which in this case is 3.3 V, and dividing it by the ADC resolution, which in the ESP32 is 4096 due to its 12-bit conversion, yields the UV radiation value in volts. Since the relationship between the sensor output voltage and UV radiation is linear, this process allows an accurate voltage value to be obtained without the need for calibration with predefined thresholds since the only transformation required is the conversion of the analog signal into its volt equivalent using the equation, without requiring additional adjustments or compensation.

$$UV\ Voltage = \left(\frac{ADC\ reading \times\ Reference\ voltage}{ADC\ resolution}\right) \tag{1}$$

This formula converts UV voltage into a UV index, a standard measure of UV radiation intensity. A low index (0-2) indicates minimal radiation, while a high index (8-11) represents an increased risk of damage from sun exposure. To obtain this index, a mapping is applied based on the sensor's measurement capability and reference voltage. In the equation, the UV voltage corresponds to the voltage previously calculated from the ADC reading, which varies between 0V and 3.3V depending on the UV radiation detected. It is multiplied by 10 to adjust the scale and divided by 3.3, the system's reference voltage, ensuring that the mapping is within the sensor's measurement values. The result is normalized to a range of 0 to 11, where 0 represents no radiation, intermediate values reflect moderate radiation levels, and 11 indicates extreme levels with a high risk of damage from prolonged exposure.

$$UV\ Index = map\left(\frac{UV\ Voltage \times 10}{3.3, 0, 11}\right) \tag{2}$$

The formula converts the sensor's humidity value into a percentage using the AirValue (2000) and WaterValue (1120) values as references. AirValue represents 0% humidity when the sensor is in dry air, and WaterValue corresponds to 100% humidity when the sensor is completely submerged in water. The sensor reading is mapped between these two points to obtain a humidity percentage. This facilitates irrigation automation, activating the pump if the humidity is low (below 40%) and turning it off if it is high (above 50%). Its resistance was first measured in arid conditions to calibrate the sensor, exposing it to air and recording a 2000 value corresponding to AirValue. It was then completely immersed in water, obtaining a value of 1120, defined as WaterValue.

$$Soil\ Moisture\ = \left(\frac{Sensor\ Value - Air\ Value}{Water\ Value\ -\ \ Air\ Value}\right) \times 100 \tag{3}$$

The equation for calculating distance using an ultrasonic sensor is based on the physical principle of sound propagation. The sensor emits a signal that travels to the water surface and returns to the sensor. The total travel time is multiplied by the speed of sound in air (343 m/s or 0.034 cm/µs), but since the signal travels both ways, it is necessary to divide by 2 to obtain only the distance between the sensor and the water surface. This formula is essential for accurately measuring the distance from the sensor to the surface of the liquid in a tank, which serves as a basis for estimating the fill level.

$$Distance = \frac{duration \times 0.034}{2} \tag{4}$$

The tank fill percentage equation converts the distance measured by the sensor into a percentage value by comparing it to predefined distances for empty and full tanks. The distance d, obtained using Formula (4), represents the separation between the ultrasonic sensor and the water surface. To calibrate the HC-SR04 sensor, the reference values d_empty = 12 cm for the empty tank and d_full = 4.32 cm for the full tank were set. These measurements were made by recording the distance from the sensor to the bottom of the tank when empty and to the water surface when full. These values transform any intermediate distance into a fill percentage, where d_empty equals 0%, and d_full equals 100%, facilitating water level monitoring in control and automation applications.

$$\%Fill = 100 \times \frac{d -\ d_{empty}}{d_{full} -\ d_{empty}} \tag{5}$$

3-6- Model Design

Figure 23 shows the prototype of the automated irrigation system, highlighting the main box that houses the circuits and electronic components essential for its operation. The hose that runs from the tank and returns to it is part of the drip irrigation system, designed to simulate irrigation in a controlled environment. A female jack for the power supply can also be seen on the side of the box. Although an additional compartment with a flower pot is used in the simulation to demonstrate the operation of the system, the main structure of the prototype is this box, which integrates all the elements necessary for the control and automation of irrigation.

Figure 24 shows the design of the device, which includes an LCD screen for real-time display of metrics such as temperature, air humidity, UV index, water level in the tank, and soil moisture. Next to the screen, a switch and a push-button allow management of the different operating modes and manual water pump control. The black access panel facilitates water tank maintenance, ensuring quick and easy intervention. Figure 25 shows the integration of a hose with small holes for drip irrigation. This hose is designed to exit the tank, distribute water in the irrigation area, and then return to the tank. This closed-loop configuration has been implemented to simulate irrigation operation in a prototype, preventing the pump from generating overpressure by keeping the water in circulation. While this arrangement is suitable for demonstrating the system, the actual configuration in a greenhouse could vary depending on the characteristics of the environment and irrigation needs. The pot compartment and the additional space have been added exclusively for demonstration purposes to visualize the system's functioning in the prototype. In a practical implementation, the structure and layout of the irrigation could be adjusted according to the specific requirements of the crop. In Figure 26, a female jack for the power supply can be seen, as well as the depth of the device and how the layout of the components is organized. Finally, in Figure 27, the vents that guarantee adequate ventilation of the internal electronic components are highlighted, preventing overheating and ensuring optimal system performance. In addition, below the right vent is a hole specifically designed for the passage of the data cable.

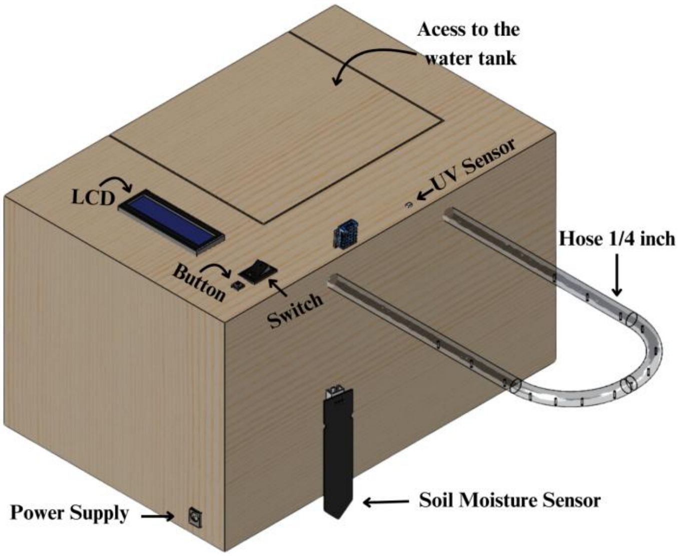


Figure 23. Isometric view of the main container

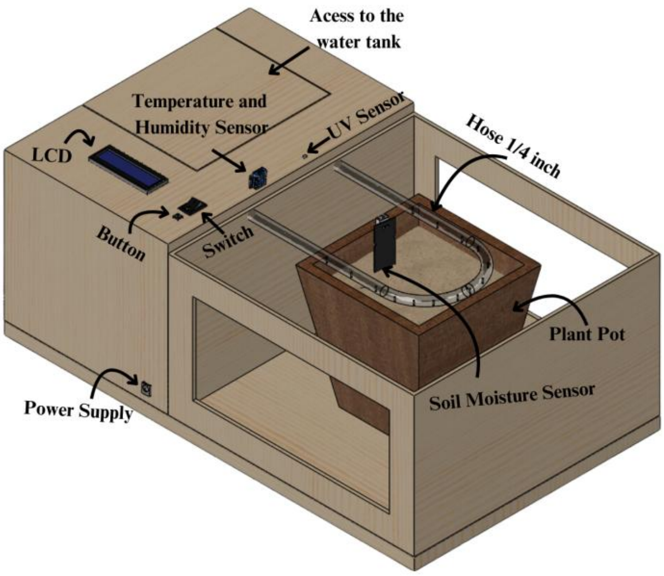


Figure 24. Isometric view of the project design

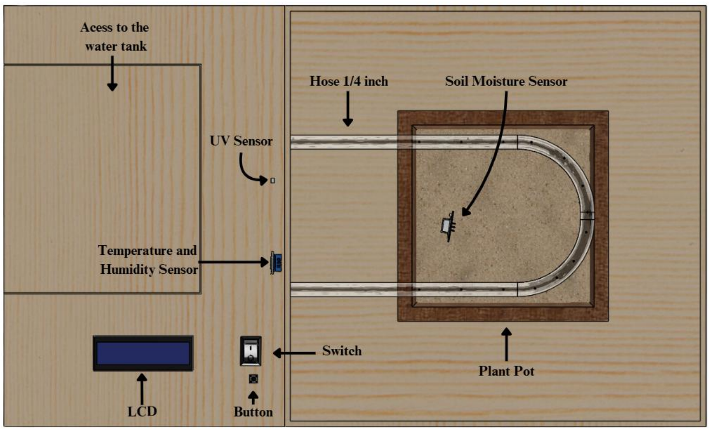


Figure 25. Top view of project design

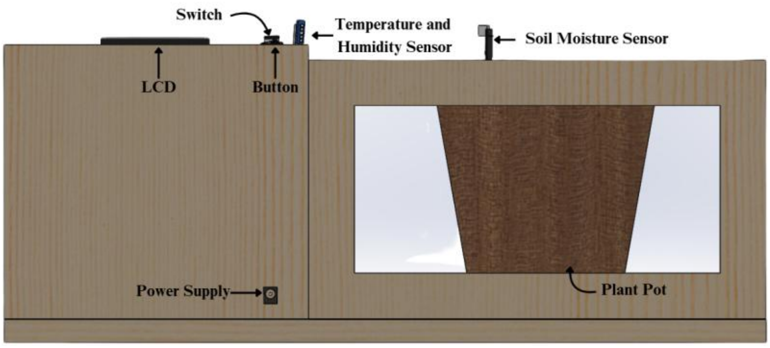


Figure 26. Side view of the project design

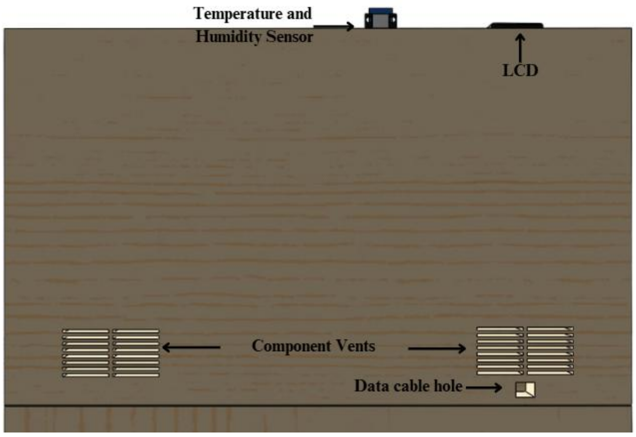
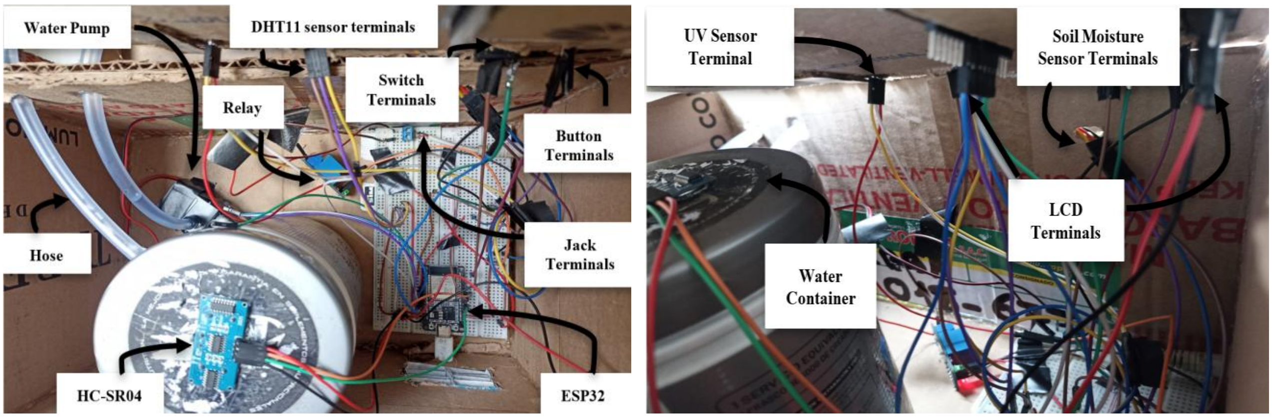


Figure 27. Front view of the project design

4- Results and Discussion

4-1- Physical Implementation

Figure 28(a) shows the physical implementation of the system, where a water tank with an ultrasonic sensor on the lid, used to measure the water level, is shown. In addition, a pump controlled by a relay and the terminals of a jack connected to this system are also shown. An ESP32 is mounted on a breadboard to which various sensors are connected. On the top are the connections of the DHT11 sensor, a switch, and a push button. On the other hand, in Figure 28(b), the side wall of the tank is shown, where the terminals of the soil moisture sensor, responsible for controlling the irrigation system, can be seen. At the top of the figure, the LCD and the UV sensor terminals are also seen.



(a) (b)

Figure 28. Physical assembly of the system.

Figure 29 shows an "Initializing…" message on the LCD screen, indicating the start of the system. Figure 30 shows the physical implementation of the system from a top view, where the sensors and components are clearly visible. In addition, a flowerpot connected to a drip irrigation system by means of a hose is included. Figure 31 shows the LCD displaying the metrics of each sensor: T (Temperature) with a value of 23°C, H (Humidity) with a value of 57%, UV

(Ultraviolet) with a value of 1 indicating a mild level, L (Water level) with a value of 100% indicating that it has reached its maximum capacity, and S (Soil moisture) with a value of 60%, meaning that the pump is off. Additionally, it can be seen that the system is operating in automatic mode, indicated by the letter "A". Near the switch are the letters A (Automatic) and M (Manual), indicating the system modes, as well as a button with the letters ON/OFF next to it.

Figure 29. Initialization message when powering on the device

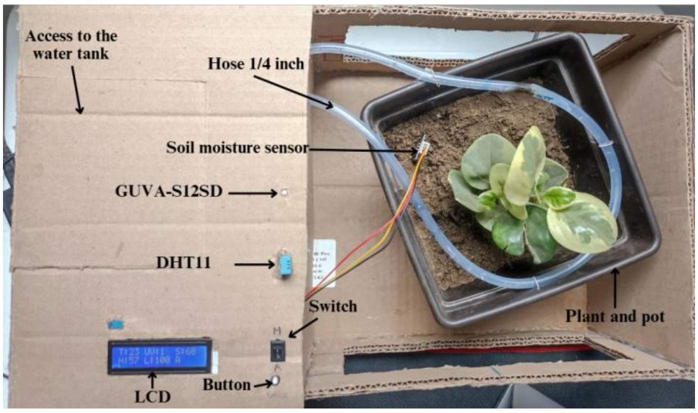


Figure 30. Top view of the physical implementation of the system in automatic mode

Figure 31. LCD screen displaying sensor metrics and automatic mode

Figure 32 shows the arrangement of the sensors and components of the system, including the pot with the plant connected to a drip irrigation system. Water droplets are observed in the irrigation area, and a color change in the soil indicates significant water absorption, leading to a humidity level of 100%. Figure 33 shows the LCD display that the system is operating in Manual mode with the ON status, allowing direct intervention in the irrigation process. The water level in the tank, indicated by the letter "L," has decreased to 88%, signaling that the system is in the middle of the irrigation process. As a result, the soil humidity, represented by the letter "S," has reached 100%, indicating over-hydration of the soil. For this reason, in automatic mode, the system was set to turn off the pump if the humidity exceeded 50% to avoid over-watering.

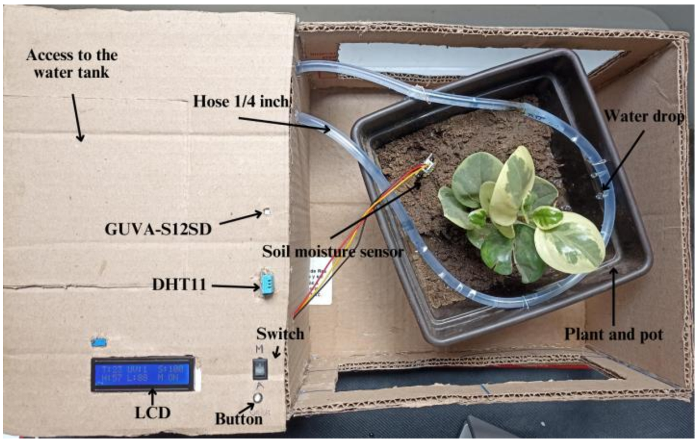


Figure 32. Top view of the physical implementation of the system in manual mode

Figure 33. **LCD screen displaying sensor metrics in manual mode with system turned ON**

Figure 34 shows an LCD display showing the metrics from different sensors in real-time. A temperature (T) of 25°C, an ambient humidity (H) of 49%, an ultraviolet (UV) index with a value of 5, indicating a moderate level, and a water level (L) of 62%, suggesting a medium level, can be observed. Soil moisture (S) is reported at 61%. The system is in manual mode with the status "OFF"; it should be noted that the "ON" or "OFF" indication only appears when the system is set to manual mode.

Figure 34. **LCD screen displaying sensor metrics in manual mode with system turned OFF**

Figure 35 presents the side view of the system implementation, which shows a green plant in a pot and its depth. At the bottom, a black cable connected to a jack can be seen, which corresponds to the power supply needed for the water pump. Figure 36 presents the front view of the system implementation with ventilation grids that ensure proper air circulation inside, which is essential to keep the circuit ventilated. In addition, a data cable is visible and connected at the bottom, which allows code updating and communication with the ESP32 and powers the system. At the top is an open access intended for the system tank, facilitating its filling or maintenance.

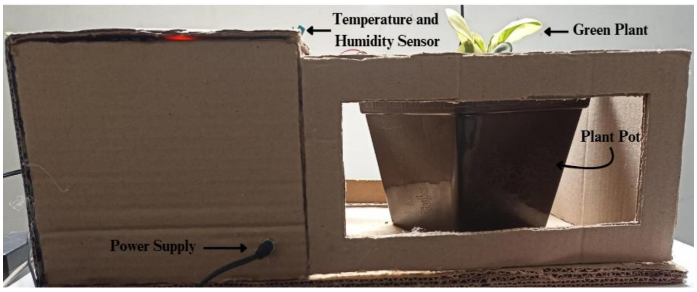


Figure 35. **Side view of the physical implementation of the system**

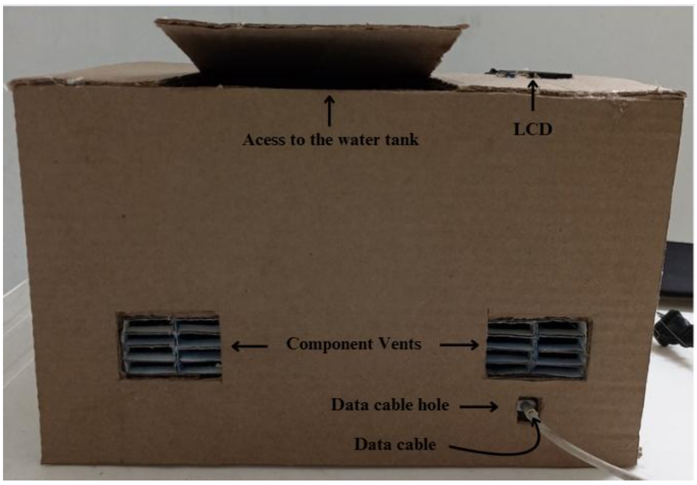


Figure 36. **Front view of the physical implementation of the system**

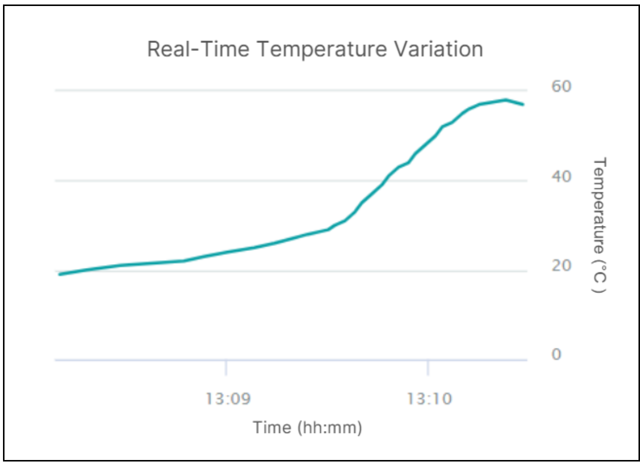
4-2- Cloud-Based Digital Aspect

In Figure 37(a), the graph shows a progressive increase in temperature inside the greenhouse, reaching a peak of approximately 58°C. Trujillo is an arid area where temperatures rarely drop below 0°C. This study established a range of 19°C to 58°C to analyze the thermal behavior inside the greenhouse and the sensor's detection limits. Although extremely low values are not relevant in this context, high temperatures can occur due to heat accumulation from solar radiation and lack of ventilation, a common problem in closed structures. The graph shows a sustained increase in temperature, indicating that thermal energy is accumulating without adequate dissipation. The value of 58°C is not a standard temperature in Trujillo, but it was included to demonstrate the effect of poor ventilation and its impact on the greenhouse microclimate. Under real-life conditions, if ventilation is limited, trapped heat can cause the temperature to exceed 40°C, affecting photosynthesis and plant metabolism. This phenomenon accelerates evapotranspiration, which can lead to water deficits in crops, increasing the need for irrigation and affecting system efficiency. To avoid these adverse effects, it is recommended to improve air circulation through natural or mechanical ventilation, the use of shading screens, or even evaporative cooling systems to maintain an adequate thermal balance.

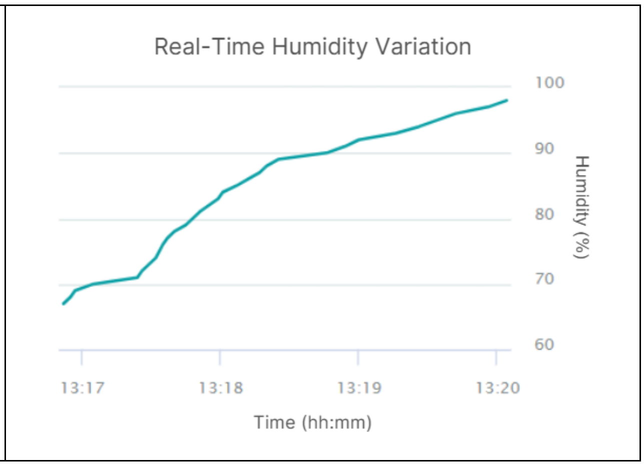
Figure 37(b) shows a progressive increase in air humidity, from 68% to almost 100% in a short period of time, reflecting a transition from moderate to extreme humidity conditions. This increase can be associated with phenomena such as rainfall, excessive condensation, or deficient ventilation in closed spaces. In indoor environments, high humidity can favor the growth of mold, bacteria, and mites, affecting air quality and human health and causing structural damage such as corrosion and deterioration of materials. However, excessively low humidity levels can also be problematic, causing dry skin, respiratory irritation, static electricity, and dehydration of materials such as wood, promoting cracking. Ideally, humidity should be maintained within an optimal range (40%–60%) to ensure comfort and prevent health problems and environmental deterioration. Figure 37(c) shows a progressive increase in soil moisture, from a dehydrated state (0%) to complete saturation (100%), passing through intermediate levels such as 50% and 54%. This behavior represents a controlled irrigation cycle, where the initial humidity is low, and as water infiltrates the soil, the sensor detects an increase in humidity. Interpreting these values is key for agricultural management and crop maintenance. A level below 35% could indicate a state of water stress, affecting plant development by reducing nutrient absorption and causing root dehydration. To avoid this condition, the automatic irrigation system is programmed to activate when humidity drops below 40%, ensuring that plants receive water before they enter a critical state of drought. When humidity reaches values between 50% and 70%, it is considered optimal for most crops, ensuring adequate water availability without the risk of waterlogging.

Finally, when the level reaches 100%, the soil is completely saturated, which can cause a lack of oxygenation in the roots and increase the likelihood of root diseases. This behavior is associated with an irrigation system in automatic mode, where water begins to be pumped when humidity drops below 40% and stops when it exceeds 50%. Much higher levels, up to 100%, would only be achieved in manual mode since, in automatic mode, the soil moisture sensor would send a signal to the ESP32, which, through the relay, would disconnect the pump once humidity exceeds 50%. In manual mode, however, the user can activate the pump without restrictions, which could lead to soil saturation. Figure 37(d) shows an increase in UV radiation from a safe level (0) to a dangerous level of 11, passing through a moderate level of 5. UV radiation is an important environmental factor influencing human health and plant development. On the ultraviolet radiation measurement scale, an index of 0 to 2 is considered low. At the same time, a value between 3 and 5 is moderate, with a moderate risk of damage to the skin and eyes during prolonged exposure. At a UV index of 6 to 7, the risk level becomes high, and when it reaches 8 to 10, it is considered very high, requiring additional protection. Finally, an index of 11 or more is classified as extreme, meaning that unprotected sun exposure can cause sunburn within minutes and negatively affect plant physiology. In a greenhouse, prolonged exposure to high UV values can cause damage to plant leaves, affect their metabolism, and decrease photosynthetic efficiency.

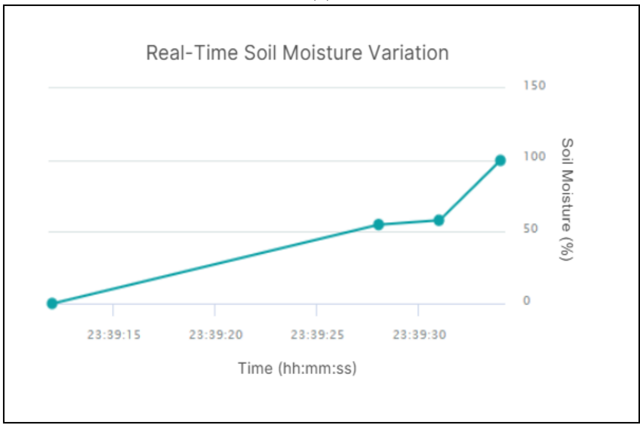
Furthermore, it can cause premature aging of materials used in the greenhouse structure, such as plastics or covering nets. To mitigate these effects, the use of UV-filtering shading nets, protective films, and monitoring systems that allow mitigation strategies to be activated when radiation reaches critical levels is recommended. Finally, Figure 37(e) shows an increase in the water level in the irrigation tank, from 0% (empty) to 100% (full), passing through an intermediate level of 50%. This behavior represents the dynamics of tank filling and its regulation based on the water supply. The water level in a tank is crucial for controlling automated irrigation systems. A level of 0% indicates that the tank is empty, which could mean that water is not available for irrigation, affecting crop supply. An intermediate level of 50% represents a state with sufficient water, but it must be monitored to prevent the level from dropping too low. Finally, 100% indicates that the tank has reached its maximum capacity, which can be a reference point to stop filling and prevent overflows.



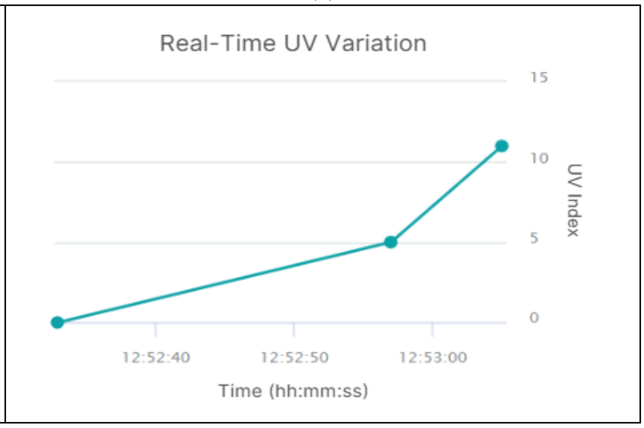
(a)



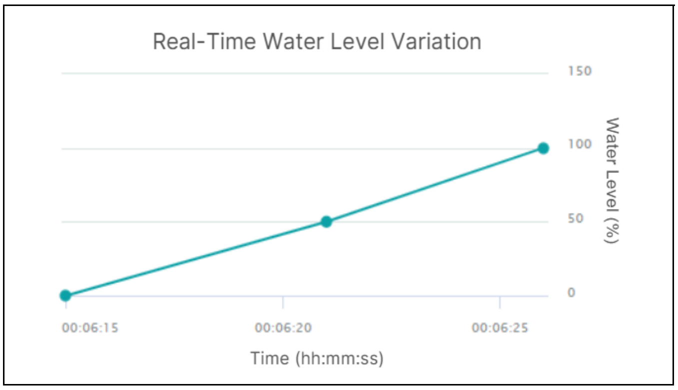
(b)



(a)



(b)



(e)

Figure 37. Real-Time Trends: a) Temperature, b) Humidity, c) Soil moisture level, d) UV level, e) water level in the tank.

Figures 38 and 39 present the real-time data monitoring interfaces provided by Arduino Cloud, designed for desktop and mobile devices, respectively. Figure 39, with a light background and adapted for large screens, organizes key metrics such as temperature, humidity, water level, soil moisture, and UV radiation in a neat and detailed format. Its design allows users to observe and compare several variables simultaneously, which is ideal for work environments or fixed monitoring stations where deep and continuous data analysis is needed. In contrast, Figure 39 shows the mobile version with a dark background optimized to provide quick and convenient access from any location. Figure 39(a) provides an overview of the monitored parameters. Subsections 39(b) and 39(c) highlight specific graphs for temperature and humidity, while subsections 39(d) and 39(e) present data on UV radiation, soil moisture, and water level. This design allows users to monitor data while performing other activities, such as moving between locations or managing tasks outside of a fixed environment, thanks to the ease of access via mobile devices. Both interfaces are synchronized through the Arduino Cloud, ensuring that information is always up to date and available in real-time, allowing users to respond immediately to changes in the monitored variables, no matter where they are.

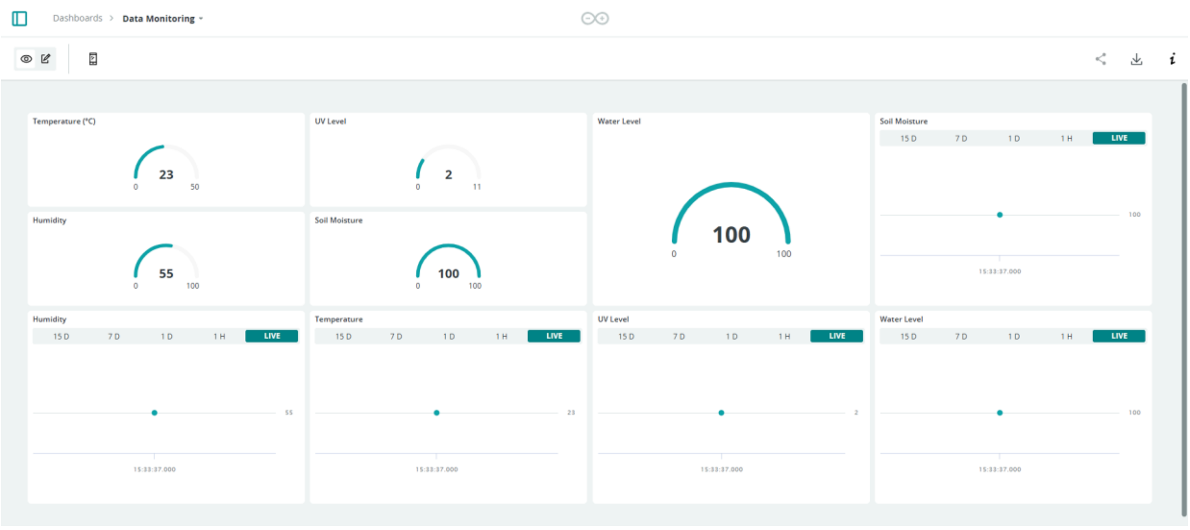


Figure 38. Desktop User Interface for Real-time Variable Monitoring on Arduino Cloud

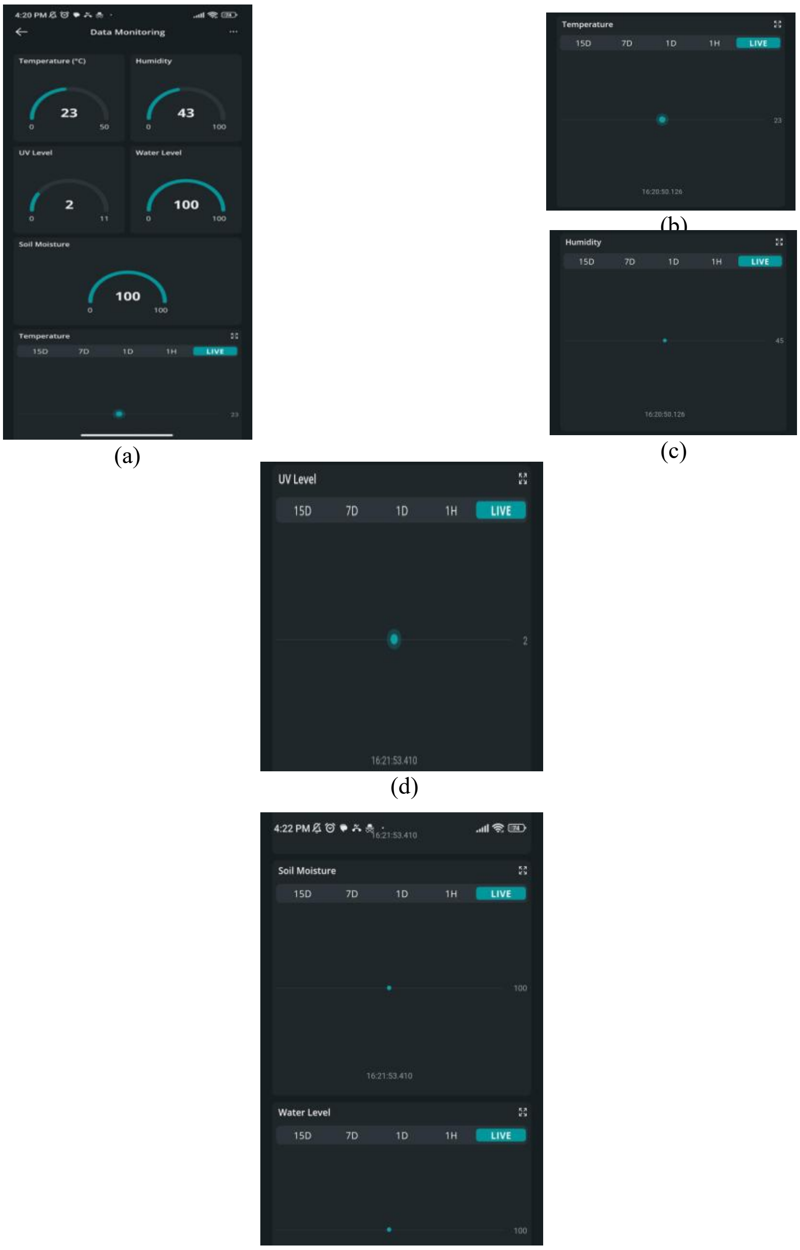


Figure 39. Mobile Interface for Real-time Data Monitoring on Arduino Cloud

4-3- ESP32 Integration and Comparison with Other Systems

The ESP32-powered monitoring and automated irrigation system proved to be an effective solution for water usage optimization, which is crucial in regions like Trujillo due to water scarcity. Unlike other Arduino-based systems like [11, 14] that require additional modules for connectivity, the ESP32 integrates Wi-Fi and Bluetooth, allowing remote monitoring through the Arduino Cloud, simplifying the design and reducing costs.

4-4- Sensors and Drip Irrigation

Unlike the Pérez-Baca et al. [11] study, which uses only one humidity sensor, this system includes additional sensors such as the GUVA-S12SD (UV radiation), DHT11 (temperature and humidity), and HC-SR04 (water level). This integration improves the accuracy of microclimate management, allowing irrigation to be automatically adjusted based on multiple variables, not just soil moisture. Furthermore, using the drip irrigation system optimizes water distribution and reduces waste, improving the system's overall efficiency.

4-5- Efficiency of Automated Drip Irrigation

Drip irrigation has proven to be a highly efficient technique for reducing water waste in agriculture. In Rahim et al. [13] study, a NodeMCU-based irrigation system and the Blynk platform were implemented, achieving a 32.5% reduction in water consumption by monitoring soil moisture. In comparison, the present study further optimizes this percentage, achieving a 35% reduction thanks to integrating multiple environmental sensors and a more robust control algorithm that dynamically adjusts irrigation based on real-time weather conditions. On the other hand, Nũnez-Tapia [14] reported a 75% reduction in water consumption using an Arduino Uno-based automatic irrigation system. However, the present study surpasses this system's connectivity and adaptability, allowing cloud integration and remote configuration of irrigation parameters. Furthermore, the ESP32 eliminates the need for additional modules, simplifying the system architecture and reducing costs.

4-6- Sensor Integration and Monitoring Accuracy

The integration of sensors in the proposed system significantly improves the accuracy of climate monitoring in greenhouses compared to previous studies. While Pérez-Baca et al. [11] used only a soil moisture sensor to activate irrigation, the present system incorporates a combination of advanced sensors, such as the DHT11 to measure ambient temperature and humidity, the GUVA-S12SD to assess UV radiation, and the HC-SR04 to monitor the water level. This integration allows for more precise microclimate control, avoiding unnecessary irrigation in high humidity conditions and optimizing water use. Furthermore, unlike Canlas et al. [15] study, which used a machine learning-based system with historical data, the proposed system offers a key advantage by allowing real-time adaptation, ensuring more efficient irrigation without relying exclusively on predefined patterns, thereby improving the response to unexpected environmental variations.

4-7- Connectivity and Remote Accessibility

The ESP32 used in this study offers significant connectivity advantages compared to the Arduino Uno or NodeMCU-based systems used in [12, 14]. While these studies required additional modules for Wi-Fi communication, the ESP32 has built-in Wi-Fi and Bluetooth connectivity, facilitating data transmission to the Arduino cloud and allowing farmers to monitor and adjust irrigation from anywhere via mobile devices. Furthermore, unlike Pérez-Baca et al. [11], where remote monitoring relied exclusively on internet connectivity, this study incorporates an LCD screen for local data display, ensuring that users can monitor the system even in rural areas with unstable connectivity.

4-8- Operating Modes: Automatic vs. Manual

This system differs from those studied in Canlas et al. [15] and Satra et al. [16] by offering a dual operating mode (automatic and manual). In automatic mode, irrigation is adjusted based on real-time sensor measurements, optimizing water use without user intervention. The farmer can manually activate or deactivate irrigation according to their specific needs. This flexibility is a key advantage over previous systems that operated exclusively in automatic mode, limiting the user's ability to intervene in the event of unforeseen conditions.

4-9- Limitations and Future Improvements

During experimental testing, some limitations in the system were identified. Fluctuations in sensor readings were observed, likely due to electromagnetic interference caused by the proximity of the cables. This situation was also reported in Nũnez-Tapia [14], suggesting the need to implement signal filters and improve cable routing to minimize electrical noise. Furthermore, following the recommendation of Abouelmehdi et al. [17], the future integration of solar panels is proposed to ensure the system's energy autonomy in areas with limited access to electricity. Combining IoT with renewable energy has proven to be an effective strategy for improving sustainability in agriculture, reducing dependence on the power grid, and optimizing system performance.

5- Conclusions

This study developed and evaluated an ESP32-based IoT system for intelligent drip irrigation optimization and climate monitoring in greenhouses. The combination of advanced sensors, cloud connectivity, and automatic control algorithms significantly improved water use efficiency and agricultural microclimate management. Experimental results confirmed that dynamic adjustments to the irrigation system, based on data collected by soil moisture, temperature, and UV radiation sensors, resulted in up to a 35% reduction in water consumption compared to traditional systems. Furthermore, integrating a cloud-based architecture facilitated real-time remote monitoring, offering users greater flexibility in irrigation management. Compared to previous studies, the proposed system stands out for its ability to operate in dual modes (automatic and manual), allowing it to adapt to different needs and improve operational efficiency.

Furthermore, using ESP32 with integrated Wi-Fi/Bluetooth connectivity simplified the design and reduced implementation costs, eliminating the need for additional modules for data transmission. Its key contributions include the optimization of drip irrigation by dynamically adjusting irrigation based on soil moisture, temperature, and UV radiation; the reduction of water waste through real-time monitoring and precise irrigation control; the integration of a local and cloud interface for data visualization via an LCD display and remote access with Arduino Cloud; operational flexibility thanks to its dual-mode, which allows the user to adjust the system according to specific needs; and its potential for scalability and sustainability through the integration of renewable energies such as solar panels to increase autonomy and reduce operating costs. In particular, UV radiation monitoring provided valuable information on potential risks to plant growth, allowing adjustments to minimize damage and optimize growing conditions. However, the system presented some limitations, such as fluctuations in sensor measurements due to electromagnetic interference, suggesting the need to implement signal filters and improve the wiring layout. Furthermore, the dependence on a stable internet connection could be a constraint in rural areas with low coverage. Future research recommends optimizing sensor accuracy through calibration and data filtering algorithms, integrating artificial intelligence and machine learning to improve crop water requirement prediction, evaluating system performance under different climatic conditions to extend its applicability to various agricultural regions, and developing an autonomous renewable-powered system incorporating solar panels to ensure its operation in areas without access to electricity. In conclusion, the proposed IoT system represents a significant advance in agricultural irrigation automation, providing a scalable, efficient, and adaptable solution for different farm environments. Its integration with renewable energy and emerging technologies can consolidate it as a key tool for the sustainability and resilience of the agricultural sector in the face of climate change.

6- Declarations

6-1- Author Contributions

Conceptualization, J.J.C.Q., M.A.T.B., and C.C.V.; methodology, J.J.C.Q., M.A.T.B., and C.C.V.; software J.J.C.Q.; validation, M.A.T.B. and C.C.V.; formal analysis, J.J.C.Q. and M.A.T.B.; investigation, J.J.C.Q., M.A.T.B., and C.C.V.; data curation, C.C.V.; writing—original draft preparation, J.J.C.Q., M.A.T.B. and C.C.V.; writing—review and editing, J.J.C.Q., M.A.T.B., and C.C.V.; visualization, J.J.C.Q., M.A.T.B., and C.C.V.; project administration, J.J.C.Q., M.A.T.B., and C.C.V.; funding acquisition, J.J.C.Q., M.A.T.B., and C.C.V. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data presented in this study are available in the article.

6-3- Funding

The authors received financial support from the Universidad Privada del Norte for the publication of this article.

6-4- Acknowledgements

The authors would like to thank Universidad Privada del Norte for supporting this work by providing a research grant for this study.

6-5- Institutional Review Board Statement

Not applicable.

6-6- Informed Consent Statement

Not applicable.

6-7- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Saleem, A., Anwar, S., Nawaz, T., Fahad, S., Saud, S., Ur Rahman, T., Khan, M. N. R., & Nawaz, T. (2024). Securing a sustainable future: the climate change threat to agriculture, food security, and sustainable development goals. Journal of Umm Al-Qura University for Applied Sciences, 1–17. doi:10.1007/s43994-024-00177-3.

[2] Batista, C., Knipper, M., Sedas, A. C., Farante, S. V., Wainstock, D., Borjas-Cavero, D. B., Araya, K. R., Arteaga España, J. C., & Yglesias-González, M. (2024). Climate change, migration, and health: perspectives from Latin America and the Caribbean. The Lancet Regional Health - Americas, 40, 100926. doi:10.1016/j.lana.2024.100926.

[3] Garreaud, R., Boisier, J. P., Álvarez-Garreton, C., Christie, D., Carrasco-Escaff, T., Vergara, I., ... & Godoy, L. (2025). Hyperdroughts in central Chile: Drivers, Impacts and Projections. EGUsphere, 2025, 1-31. doi:10.5194/EGUSPHERE-2025-517.

[4] Bolan, S., Padhye, L. P., Jasemizad, T., Govarthanan, M., Karmegam, N., Wijesekara, H., Amarasiri, D., Hou, D., Zhou, P., Biswal, B. K., Balasubramanian, R., Wang, H., Siddique, K. H. M., Rinklebe, J., Kirkham, M. B., & Bolan, N. (2024). Impacts of climate change on the fate of contaminants through extreme weather events. Science of the Total Environment, 909, 168388. doi:10.1016/j.scitotenv.2023.168388.

[5] Kumar, L., Chhogyel, N., Gopalakrishnan, T., Hasan, M. K., Jayasinghe, S. L., Kariyawasam, C. S., Kogo, B. K., & Ratnayake, S. (2021). Climate change and future of agri-food production. Future Foods: Global Trends, Opportunities, and Sustainability Challenges, 49–79. doi:10.1016/B978-0-323-91001-9.00009-8.

[6] Lozano-Povis, A. A. (2023). Agriculture and climate change: Main findings and proposals for decision-making in two natural regions of Peru. South Sustainability, 4(1), 1-4. doi:10.21142/ss-0401-2023-e068.

[7] Choudhary, V., Guha, P., Pau, G., & Mishra, S. (2025). An overview of smart agriculture using internet of things (IoT) and web services. Environmental and Sustainability Indicators, 26, 100607. doi:10.1016/j.indic.2025.100607.

[8] Mane, S. S., Narawade, V., & Ranshur, N. J. (2024). Revolutionizing Agriculture Soil Testing with Agriculture 4.0 and IoT Integration. Current Agriculture Research Journal, 12(3), 1333–1344. doi:10.12944/CARJ.12.3.26.

[9] Saha, G., Shahrin, F., Khan, F. H., Meshkat, M. M., & Azad, A. A. M. (2025). Smart IoT-driven precision agriculture: Land mapping, crop prediction, and irrigation system. PLoS ONE, 20(3 March), 319268. doi:10.1371/journal.pone.0319268.

[10] Hashemi, S. Z., Darzi-Naftchali, A., Karandish, F., Ritzema, H., & Solaimani, K. (2024). Enhancing agricultural sustainability with water and crop management strategies in modern irrigation and drainage networks. Agricultural Water Management, 305, 109110. doi:10.1016/j.agwat.2024.109110.

[11] Pérez-Baca, M. S., Sambrano-Luna, K. L., Sánchez-Ramírez, J. M., Cabana-Cáceres, M., & Castro-Vargas, C. (2024). Design and implementation of an automated irrigation control for home plantations. Indonesian Journal of Electrical Engineering and Computer Science, 35(3), 1437–1446. doi:10.11591/ijeecs.v35.i3.pp1437-1446.

[12] Shahar, S. H., Ismail, S. I., Dzulkefli, N. N. S. N., Abdullah, R., & Zain, M. F. M. (2023). Arduino based irrigation monitoring system using Node microcontroller unit and Blynk application. Indonesian Journal of Electrical Engineering and Computer Science, 31(3), 1334–1341. doi:10.11591/ijeecs.v31.i3.pp1334-1341.

[13] Rahim, A. A., Mohamad, R., Shuhaimi, N. I., & Buclatin, W. C. (2023). Real-time soil monitoring and irrigation system for taro yam cultivation. Indonesian Journal of Electrical Engineering and Computer Science, 32(2), 1042–1049. doi:10.11591/ijeecs.v32.i2.pp1042-1049.

[14] Ñũnez-Tapia, L. (2020). A prototype of an automatic irrigation system for peruvian crop fields. International Journal of Advanced Computer Science and Applications, 11(8), 731–734. doi:10.14569/IJACSA.2020.0110888.

[15] Canlas, F. Q., Al Falahi, M., & Nair, S. (2022). IoT based Date Palm Water Management System Using Case-Based Reasoning and Linear Regression for Trend Analysis. International Journal of Advanced Computer Science and Applications, 13(2), 549–556. doi:10.14569/IJACSA.2022.0130264.

[16] Satra, S., Agrawal, A., Gogate, S., Daryapurkar, R., & Mehendale, N. (2023). Design and Implementation of Arduino-based Automatic Irrigation with Moisture Sensor. SSRN Electronic Journal, 1-5. doi:10.2139/ssrn.4513859.

[17] Abouelmehdi, K., Elhattab, K., & El Moutaouakkil, A. (2022). Smart Agriculture Monitoring System using Clean Energy. International Journal of Advanced Computer Science and Applications, 13(5), 370–377. doi:10.14569/IJACSA.2022.0130544.

[18] Padhiary, M., Hoque, A., Prasad, G., Kumar, K., & Sahu, B. (2025). Precision agriculture and AI-driven resource optimization for sustainable land and resource management. Smart Water Technology for Sustainable Management in Modern Cities, 197–231. doi:10.4018/979-8-3693-8074-1.ch009.

[19] Lyu, L., Matheson, S., Fleck, R., Torpy, F. R., & Irga, P. J. (2024). Modulating phytoremediation: How drip irrigation system affect performance of active green wall and microbial community changes. Journal of Environmental Management, 370, 122646. doi:10.1016/j.jenvman.2024.122646.

[20] Nsoh, B., Katimbo, A., Guo, H., Heeren, D. M., Nakabuye, H. N., Qiao, X., Ge, Y., Rudnick, D. R., Wanyama, J., & Bwambale, E. (2024). Internet of Things-Based Automated Solutions Utilizing Machine Learning for Smart and Real-Time Irrigation Management : A Review. Sensors (Switzerland), 24(7480), 1–37. doi:https://doi.org/10.3390/s24237480.

[21] Morchid, A., Et-taibi, B., Oughannou, Z., Alami, R. El, Qjidaa, H., Jamil, M. O., Boufounas, E. M., & Abid, M. R. (2025). IoT-enabled smart agriculture for improving water management: A smart irrigation control using embedded systems and Server-Sent Events. Scientific African, 27, 2527. doi:10.1016/j.sciaf.2024.e02527.

[22] Al-Qudah, R., Almuhajri, M., & Suen, C. Y. (2025). Unveiling the potential of sustainable agriculture: A comprehensive survey on the advancement of AI and sensory data for smart greenhouses. Computers and Electronics in Agriculture, 229, 109721. doi:10.1016/j.compag.2024.109721.

[23] Gaitan, N. C., Batinas, B. I., Ursu, C., & Crainiciuc, F. N. (2025). Integrating Artificial Intelligence into an Automated Irrigation System. Sensors, 25(4), 1199. doi:10.3390/s25041199.

[24] Mohsin Tahir, D., & Omran Al-Sulttani, A. (2024). Smart Irrigation Technique in the Fixed Irrigation System Based on Soil Moisture Content. IOP Conference Series: Earth and Environmental Science, 1374(1), 12061. doi:10.1088/1755-1315/1374/1/012061.

[25] Duguma, A. L., & Bai, X. (2025). How the internet of things technology improves agricultural efficiency. Artificial Intelligence Review, 58(2), 1–26. doi:10.1007/s10462-024-11046-0.

[26] Preite, L., & Vignali, G. (2024). Artificial intelligence to optimize water consumption in agriculture: A predictive algorithm-based irrigation management system. Computers and Electronics in Agriculture, 223, 109126. doi:10.1016/j.compag.2024.109126.

[27] Parra-López, C., Ben Abdallah, S., Garcia-Garcia, G., Hassoun, A., Trollman, H., Jagtap, S., Gupta, S., Aït-Kaddour, A., Makmuang, S., & Carmona-Torres, C. (2025). Digital technologies for water use and management in agriculture: Recent applications and future outlook. Agricultural Water Management, 309, 109347. doi:10.1016/j.agwat.2025.109347.

[28] Mohan, R. N. V. J., Rayanoothala, P. S., & Sree, R. P. (2024). Next-gen agriculture: integrating AI and XAI for precision crop yield predictions. Frontiers in Plant Science, 15, 1451607. doi:10.3389/fpls.2024.1451607.

[29] Balamurali, D., Chakankar, S., Sharma, G., Pagey, A. P., Natarajan, M., Shaik, S., Gnanavendan, S., & Arıcı, M. (2025). A solar-powered, internet of things (IoT)-controlled water irrigation system supported by rainfall forecasts utilizing aerosols: a review. Environment, Development and Sustainability, 1–40. doi:10.1007/s10668-024-05953-z.

[30] Mousavi, R., Mousavi, A., Mousavi, Y., Tavasoli, M., Arab, A., Kucukdemiral, I. B., Alfi, A., & Fekih, A. (2025). Revolutionizing solar energy resources: The central role of generative AI in elevating system sustainability and efficiency. Applied Energy, 382, 125296. doi:10.1016/j.apenergy.2025.125296.

[31] Tychola, K. A., & Rantos, K. (2025). Cyberthreats and Security Measures in Drone-Assisted Agriculture. Electronics (Switzerland), 14(1), 149. doi:10.3390/electronics14010149.

[32] Zhukabayeva, T., Zholshiyeva, L., Karabayev, N., Khan, S., & Alnazzawi, N. (2025). Cybersecurity Solutions for Industrial Internet of Things–Edge Computing Integration: Challenges, Threats, and Future Directions. Sensors, 25(1), 213. doi:10.3390/s25010213.

[33] Chang, Y. H., Wu, F. C., & Lin, H. W. (2025). Design and Implementation of ESP32-Based Edge Computing for Object Detection. Sensors, 25(6), 1656. doi:10.3390/s25061656.

[34] Frumkin, H., Geller, R. J., Rubin, I. L., & Nodvin, J. (2009). Safe and Healthy School Environments. Safe and Healthy School Environments, 1–480. doi:10.1093/acprof:oso/9780195179477.001.0001.

[35] Chowdhury, M., Ahsan, T. M. A., & Ahamed, M. S. (2023). Assessment of health hazards of greenhouse workers considering UV exposure and thermal comfort. Smart Agricultural Technology, 5, 100319. doi:10.1016/j.atech.2023.100319.

[36] Abdelmoneim, A. A., Al Kalaany, C. M., Khadra, R., Derardja, B., & Dragonetti, G. (2025). Calibration of Low-Cost Capacitive Soil Moisture Sensors for Irrigation Management Applications. Sensors, 25(2), 343. doi:10.3390/s25020343.

[37] Sulistyawan, V. N., Salim, N. A., Abas, F. G., & Aulia, N. (2023). Parking Tracking System Using Ultrasonic Sensor HC-SR04 and NODEMCU ESP8266 Based IoT. IOP Conference Series: Earth and Environmental Science, 1203(1), 12028. doi:10.1088/1755-1315/1203/1/012028.

[38] Prakash, S. S., Usha, R., Karuppiah, N., Saravanan, S., Kalaiyarasi, M., & Karunanithi, K. (2025). Smart Home and Security Systems: An IoTBased Approach Utilizing ESP 32 and Multisensor Integration. E3S Web of Conferences, 616, 2004. doi:10.1051/e3sconf/202561602004.

[39] Ngoma, D. H., Nkongo, D., Abdul-Rahman, H. M., Muna, B. H., Buberwa, A. P., Ngaiza, E., & Rhee, H. (2025). Design and Development of IoT Smart Drip Irrigation and Fertigation Prototype for Small and Medium Scale Farmers. Acase Study of Tomato Farmers in Tanzania. Journal of The Institution of Engineers (India): Series A, 1–23. doi:10.1007/s40030-024-00857-7.

[40] Sallem, A., Souissi, Z., Nasri, M. A., Benhala, B., & Masmoudi, N. (2025). ESP32 charging system prototype for EV: Design and implementation using wireless energy transmission. E3S Web of Conferences, 601, 112. doi:10.1051/e3sconf/202560100112.