IoT

MDPI

Article

A Smart IoT-Enabled 3D-Printed Hydroponic Tower with Real-Time pH Control

Eleftheria Maria Pechlivani [1,*], Sotirios Pemas [1] and Vasileios Zoidis [2,*]

[1] Centre for Research and Technology Hellas, Information Technologies Institute, 6th km Charilaou-Thermi Road, 57001 Thessaloniki, Greece; sopemas@iti.gr (S.P.)
[2] School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; vgzoidis@ece.auth.gr
* Correspondence: riapechl@iti.gr; Tel.: +30-231-125-7751

Abstract: Traditional hydroponic systems require significant manual effort to monitor and maintain optimal growing conditions for plants. This paper presents the design and implementation of a low-cost IoT-based Smart Hydroponic Tower that enables detailed remote monitoring and automates the most intensive parts of this process. The system is built around an ESP32 microcontroller that collects data from a suite of sensors monitoring water level, temperature, pH, electrical conductivity (EC), along with ambient temperature, humidity, light, and CO2 levels. The collected data are displayed on a local LCD screen, streamed to a custom mobile app (React-Native) and webpage (HTML) for remote monitoring, as well as saved to a cloud database (Supabase). Moreover, the ESP32 handles the scheduling of the pumps for the nutrient circulation and pH regulation according to the user's configuration. Consequently, the system successfully automates the monitoring and control of the hydroponic environment, providing real-time data visualization both locally and remotely. The automated control of nutrient circulation and pH levels ensures that plants are grown in optimal conditions, while the data logging feature allows for historical analysis and optimization of the growing process. To sum up, the Smart Hydroponic Tower provides an effective and affordable solution for automating hydroponic agriculture. The system's modular design and remote-control capabilities make it suitable for a wide range of applications, from hobbyist growers to small-scale commercial farms.

Keywords: Vertical Hydroponic Farming (VHF); Internet of Things (IoT); smart agriculture; remote monitoring; embedded systems; 3D printing; sustainable food production.

Academic Editor: Firstname Lastname

Received: date
Revised: date
Accepted: date
Published: date

Citation: To be added by editorial staff during production.

1. Introduction

Demographic projections indicate that the global population, currently at around 8 billion, is expected to rise to 9.7 billion by 2050 and may reach a peak of nearly 10.4 billion by the mid-2080s [1]. Such rapid population growth is expected to significantly increase the demand for food production, however United Nations reports that the world is not on course to achieve Sustainable Development Goal 2, Zero Hunger, by 2030, nor the established global nutrition targets [2]. Meeting global food needs requires more farmland, yet one-third of arable land has disappeared over the last 40 years due to erosion and pollution [3]. It is evident that current industrial agricultural methods are depleting soil quality at a rate much faster than natural processes can restore it, while consuming roughly 70% of available water through unsustainable irrigation practices [2].Vertical hydroponic farming (VHF) offers a promising solution to the global food security crisis, by cultivating crops in vertically stacked layers —using up to 90% less land than conventional agriculture— while conserving 70–95% more water through closed-loop recycling systems [4].

Generally, hydroponics is a method of growing plants without soil, by using mineral nutrient solutions in a water solvent, that has gained significant attention as a sustainable agricultural technique [5], [6]. It offers numerous advantages over traditional agriculture, including higher crop yields, faster growth rates, and reduced water consumption [7].

Specifically, vertical hydroponic farming holds distinct advantages over conventional hydroponic farming by optimizing space (enabling cultivation in urban environments), efficiency (increased yield per surface area), and thus sustainability [4]. Moreover, vertical hydroponic farms that are established in controlled environments near urban centers have shorter supply chains (reducing fossil fuel use) and ensure consistent year-round harvests regardless of climate variability [8]. However, maintaining the optimal conditions for plant growth in a hydroponic system can be a labor-intensive process, requiring frequent monitoring of various environmental and water parameters such as pH, electrical conductivity (EC), and temperature [9], [10].

The advent of autonomous control systems (ACS) and Internet of Things (IoT) has created new possibilities for automating and optimizing agricultural processes [2], [11]. By integrating sensors, microcontrollers, and wireless communication technologies, it is possible to create "smart" farming systems that can monitor and control the growing environment with minimal human intervention [12].

The primary objective of this research is to offer a practical, accessible solution for eliminating the manual labor associated with monitoring the environmental parameters of a hydroponic cultivation, maintaining automatically the optimal nutrition solution pH levels, in order to increase nutrient availability [13], while ultimately, storing all this information, with the aim of doing future research on data analytics with VHF systems to further enhance their potential. Key contributions of this project include the development of a cohesive hardware and software ecosystem, that encompasses the creation of a custom printed circuit board (PCB), an ESP32-based control unit, a webpage, a mobile application and a cloud database. By integrating these components, we aim to significantly improve the efficiency and effectiveness of VHF and general hydroponic farming practices.

2. Materials and Methods

The Smart Hydroponic Tower system is composed of three main parts: the physical tower and its integrated hardware, the ESP32-based firmware that facilitates control and data acquisition, and a mobile application / website for remote monitoring and management. This section details additive manufacturing techniques and the assembly process for the proposed prototype device. Additionally, it describes the hardware design of the proposed device, such as the microcontroller, sensors, LCD and PCB. Furthermore, it presents the whole system's firmware and software including the cloud database (Figure 1). (Finally, the experimental setup and the performance evaluation of the proposed device are presented.?)

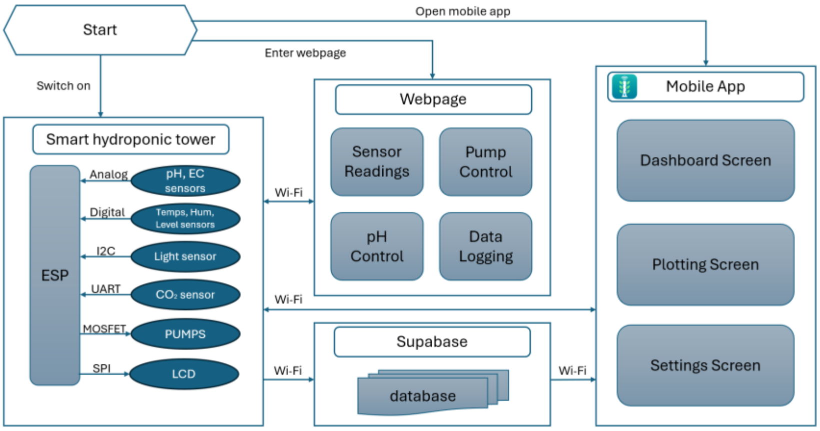
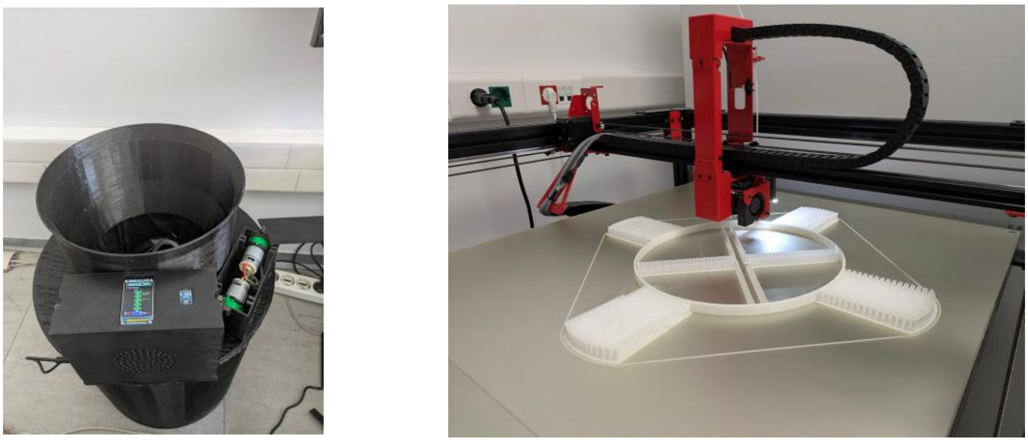


Figure 1. Architectural diagram of the Smart Hydroponic tower.

2.1. 3D-printed structure

2.1.1. Additive Manufacturing

The black tank is better suited to block sunlight, preventing algae build up and evaporation of the nutrient solution.



(**a**) (**b**)

Figure 1. Smart Hydroponic Tower Structure in the making: (**a**) Base of the tower assembled; (**b**) first stage of the main tower being printed.

2.1.2. Irrigation system

At the center of every main tower section a Palaplast 360° (max 40lt/h) 6mm Irrigation Nozzle is fixed in place. These nozzles were chosen because they can easily be daisy chained together -to match the number of tower sections- using 6mm tubes and T-shaped connectors (Figure 2), while sufficiently watering every plant position at the perimeter of each section.

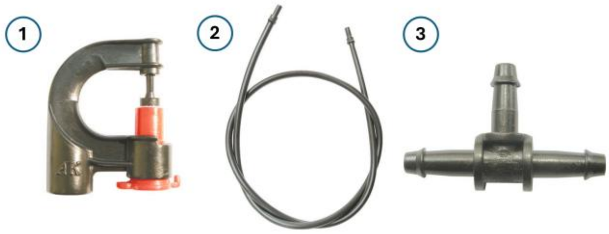


Figure 2. Expandable irrigation system: 1. 360° sprinkler; 2. 6mm tube; 3. T-shaped connector.

2.2. Hardware

The core of the control system is an ESP32-WROOM-32 development board, chosen for its low cost, Wi-Fi capabilities, and extensive community support [14]. More specifically the ESP32-DevKitC V4 from Espressif Systems was chosen because it is a compact, feature-rich development board with a lot of I/O pins broken out to pin headers. In detail, it features [15]:

1. A USB-to-UART Bridge with speeds up to 3 Mbps which was crucial for fast iteration during the development stage of this project;
2. Enough general-purpose input/output (GPIO) pins;
3. 12-bit ADC for reading the analog signals generated by the pH and EC sensors;
4. Motor PWM to control the speed of the main pump;
5. The necessary communication protocols such as:
    a. UART to collect data from the CO_2 sensor;
    b. I2C to collect data from the light sensor;
    c. SPI to send data to the LCD;
6. Integrated Wi-Fi antenna (2.4 GHz) with up to 150Mbps speeds.

2.2.1. Sensors

This microcontroller interfaces with a comprehensive suite of sensors (as presented in Figure 2 and Table 1) to monitor the hydroponic environment in real-time. Key parameters tracked include:

7. Temperature of the nutrient solution;
8. Acidity of the solution;
9. Nutrient concentration (corrected for the solution’s temperature);
10. Temperature and humidity around the tower;
11. Ambient light intensity;
12. Concentration of carbon dioxide in the air;
13. Water level in the reservoir (to prevent the pumps from running dry).

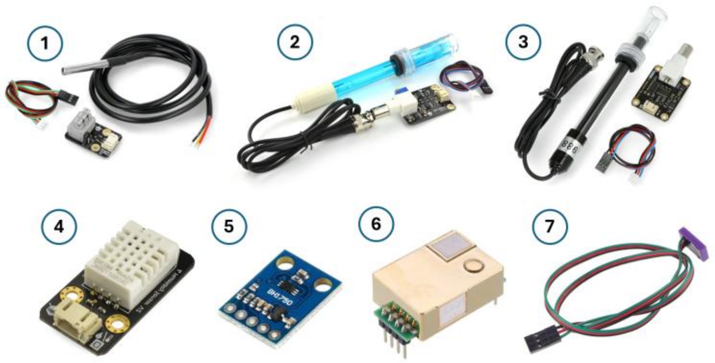


Figure 2. Sensors used to monitor the hydroponic environment.

Table 1. Sensor specifications.

No	Sensor type	Part Number	Parameter	Value	Units
1	Water Temperature Digital	DFR0198 - DFRobot (Waterproof DS18B20)	Usable Range	- 55 to 125	°C
			Accuracy	± 0.5	°C
			Response time	750	ms
2	pH Level Analog	SEN0161 - DFRobot	Usable Range	0 to 14	pH
			Accuracy	± 0.1	pH
3	Electrical Conductivity Analog	DFR0300 - DFRobot	Usable Range	0 to 20	ms/cm
			Accuracy	± 5%	F.S.*
			Probe life	> 0.5	year
4	Temperature and Humidity Digital	SEN0137 - DFRobot (DHT22)	Tem Usable Range	- 40 to 80	°C
			Temp Accuracy	0.1	°C
			Temp Error	< ± 0.5	°C
			Hum Usable Range	0 to 100%	RH
			Hum Accuracy	0.1%	RH
			Hum Error	< ± 2%	RH
5	Light Level I2C	BH1750 - OEM	Usable Range	1 to 65535	lux
			Accuracy	± 20%	F.S.*
6	CO_2 Level UART	MH-Z19B - Winsen	Usable Range	0 to 5000	ppm
			Accuracy	± 50ppm + 3%	R.V.**
7	Water level Digital	SEN0485 - DFRobot	Output	High / Low	Level
			Working Temperature	5 to 80	°C

* Full scale.

** Reading value.

2.2.2. LCD display

A 2-inch LCD IPS display module is used to provide a local user interface, showing real-time sensor readings and system status (as shown in Figure 3). The display module has a resolution of 240 by 320 and it supports the SPI communication protocol.

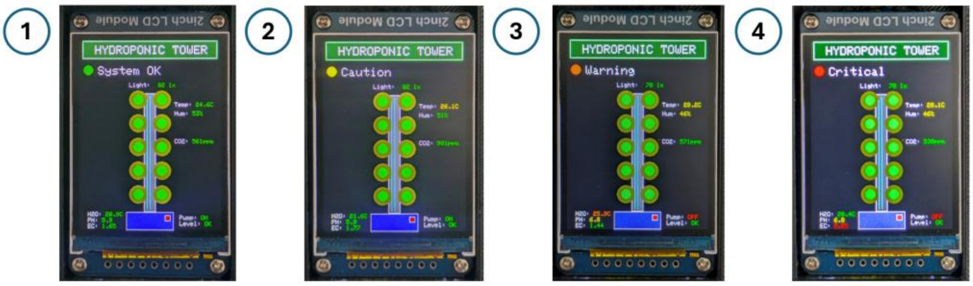


Figure 3. The LCD screen displaying: 1. The System OK; 2. Caution because the environmental temperature is a little high; 3. Warning because the nutrient solution temperature is high; 4. Critical due to the electrical conductivity of the solution being extremely low (meaning there are not enough nutrients for the plants).

2.2.3. Pumps

The ESP32 also controls 12V submersible (main) pump and two 12V peristaltic pumps for circulating the main solution and for dispensing pH up/down solutions accordingly (Figure 4).

Figure 4. 1. Submersible pump; 2. Peristaltic pump.

2.2.4. PCB

A Printed Circuit Board (PCB) was designed to provide a compact and reliable solution for connecting all the components. However, due to time constraints a perfboard with the same schematic was also created and used for the purposes of testing this prototype (Figure 5).

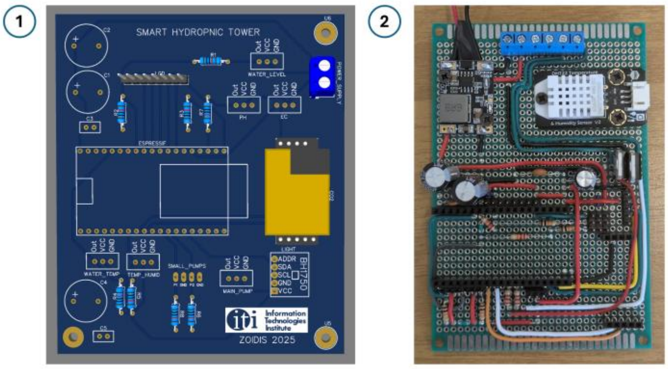


Figure 5. 1. Custom PCB; 2. Perfboard circuit.

2.2.5. System Integration and Enclosure

The complete system is powered by a single 12V DC main supply, which integrates all components seamlessly. This power supply directly drives the submersible and peristaltic pumps, to minimize noise interference with the ESP32 microcontroller and its measurements. To further enhance the system's resilience to disruptions, different size capacitors are utilized to filter out high and low frequency interference on both the 5V and 3.3V power rails [16].

A dedicated DC-DC converter steps down the voltage to 5V, safely powering the ESP32 and certain connected sensors. The ESP32 itself features an integrated voltage regulator that steps down the voltage from 5V to 3.3V. The regulator provides a maximum output current of around 600mA, which is sufficient for the ESP32 to power its operations, the LCD display, and the rest of the sensors [17].

The entire assembly is housed within a 3D-printed enclosure, which is perforated with strategic openings. This design protects the electronics from physical damage and moisture while ensuring sufficient airflow for the air-quality sensors (CO_2, temperature, humidity). On top, a clear window is incorporated to keep the LCD display visible to the user and to allow ambient light to reach the light sensor (Figure 6).

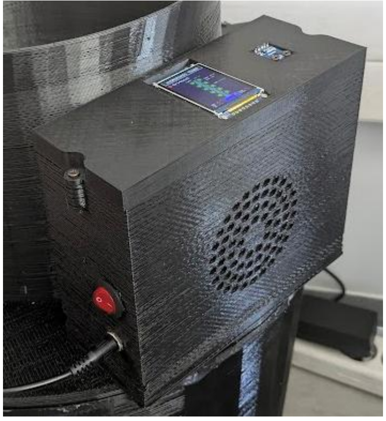


Figure 6. Change when window is implemented!!

2.3. ESP32 Firmware

The firmware for the ESP32 was developed in C++ using the Arduino framework of the PlatformIO plug-in for the Visual Studio Code IDE. The code is modular, with separate files for handling sensors, display, pump control, Wi-Fi connectivity, and data logging, offering easier maintainability and scalability.

Furthermore, to ensure operational integrity, a Watchdog Timer is employed to automatically reset the system in case of prolonged unresponsiveness. At the same time comprehensive error handling mechanisms provide checks and recovery protocols for critical operations such as network resilience. This way the system can maintain functionality even during Wi-Fi disruptions, automatically attempting reconnection in the background while ensuring continuous operation.

2.3.1. Sensor Reading

The `sensors.cpp` module is responsible for initializing and reading data from all the sensors, as well as making the appropriate ADC and offset calculations when needed. Of course, all necessary sensor calibrations have been performed beforehand and custom scripts are available to calibrate each sensor again whenever its manufacturer recommends. Additionally, to ensure accurate but responsive readings, a tuned moving average filter has been implemented for the analog pH sensor, in order to have most reliable data possible to control for in the `pump_control.cpp` module.

2.3.2. LCD Display

The `display.cpp` module manages the LCD display, rendering real-time data and status indicators with color codes (green for optimal, yellow for caution, orange for warning, red for critical). To maximize the perceived refresh rate, the system only renders the whole screen at the start, afterwards, it only erases and rewrites values as needed.

2.3.3. Pump control

The `pump_control.cpp` module implements the logic for automated pump cycling and pH control. The main circulation pump operates on a configurable timer (defaulting to 15 minutes on, 45 minutes off for an hour-long duty cycle). The pH control system employs a feedback loop to maintain the pH within a specified target range. It activates the pH adjustment pumps, either increasing or decreasing the pH as necessary, while ensuring a cooldown period is observed to prevent overshooting the desired pH level.

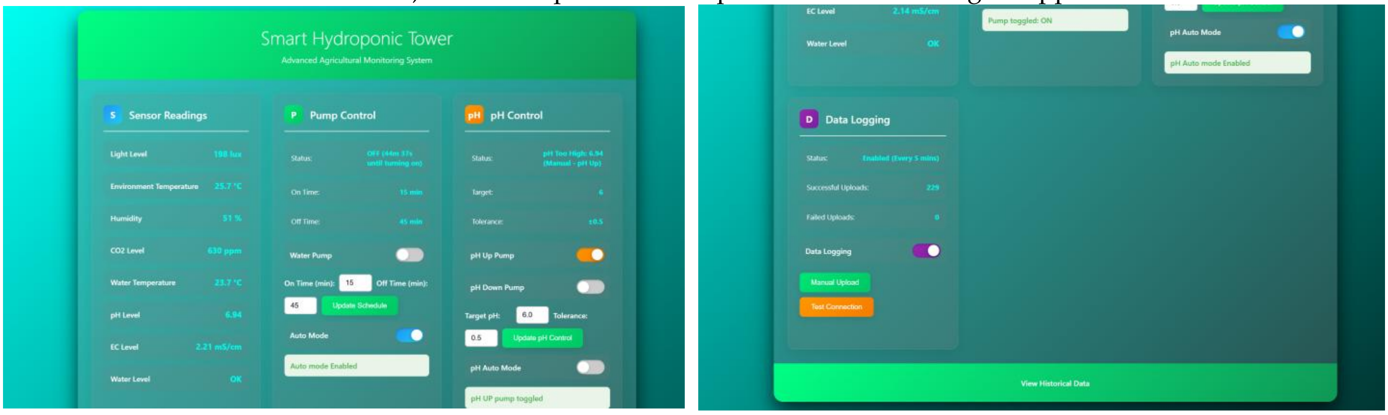
Additionally, a manual override feature is available for both systems, providing users with the flexibility to manually control pump operation. This functionality is particularly beneficial when preparing a new nutrient solution or flushing the reservoirs.

2.3.4. Wi-Fi Server

The `wifi_server.cpp` module is responsible for connecting the device to a Wi-Fi network using a static IP address. This is useful because it provides a stable address for the system, ensuring that a user can reliably connect to the web interface without having to find a new IP address if the device or network resets.

Once connected, it initializes an Async Web Server, which can handle multiple web requests simultaneously without blocking other critical tasks like reading sensors or controlling pumps. The server provides several key functions such exposing RESTful APIs that allow a client to fetch real-time sensor data in JSON format, control the main water pump, manage the pH dosing pumps, and interact with the data logging module.

Ultimately, in its home address, the server serves a webpage that delivers a user interface directly to a web browser (Figure 7). This allows any user on the network to easily monitor and control the hydroponic tower (and interact with all the APIs we mentioned before) from their phone or computer without needing an application.



Smart Hydroponic Tower
Advanced Agricultural Monitoring System

S Sensor Readings

Light Level	198 lux
Environment Temperature	25.7 °C
Humidity	51 %
CO2 Level	630 ppm
Water Temperature	23.7 °C
pH Level	6.94
EC Level	2.21 mS/cm
Water Level	OK

P Pump Control

Status: OFF (44m 37s until turning on)
On Time: 15 min
Off Time: 45 min
Water Pump ○
On Time (min): 15 Off Time (min): 45
Auto Mode ●
Auto mode Enabled

pH pH Control

Status: pH Too High: 6.94 (Manual - pH Up)
Target: 6
Tolerance: ±0.5
pH Up Pump ●
pH Down Pump ○
Target pH: 6.0 Tolerance: 0.5
pH Auto Mode ○
pH UP pump toggled

(a)

EC Level 2.14 mS/cm
Water Level OK
Pump toggled: ON
pH Auto Mode ●
pH Auto mode Enabled

D Data Logging

Status: Enabled (Every 5 mins)
Successful Uploads: 229
Failed Uploads: 0
Data Logging ●

(b)

Figure 7. Screenshots of the webpage: (**a**) Top of the page showing the sensor, pump and pH control panels; (**b**) bottom of the page showing the data logging panel.

2.3.5. Data logging

Finally, the `data_logger.cpp` module is responsible for the continuous logging of sensor data to a Supabase cloud database at regular five-minute intervals. This allows for long-term data storage and analysis, empowering users to make informed decisions based on historical trends. An advantage of this approach is that the data is accessible from anywhere via the internet, rather than being limited to the local network. To mitigate data loss, the ESP32 automatically retries uploading to the database with exponential backoff. This, along with the background reconnection strategy, ensures continuity and backfill behavior, avoiding data gaps for short Wi-Fi outages.

Note: The source code for both the ESP32 firmware and the mobile application is available in the project's GitHub repository [18].

2.4. Mobile Application

For the purposes of this project, a mobile application for Android was developed, using the React Native framework and TypeScript. The app provides a user-friendly interface for interacting with the Smart Hydroponic Tower and following the data that it has gathered. It features three main screens that are easy to navigate from the bottom of the screen, all the while displaying the system's status and connectivity.

Furthermore, to ensure a reliable and responsive user experience the following have been implemented. To handle potential network instability, request timeouts with appropriate error messaging, prevent the app from freezing if the hardware is offline or slow to respond. Moreover, the app employs optimistic UI updates, making controls like the pump toggle feel instantaneous. Performance is further enhanced through intelligent state management; the app avoids unnecessary re-renders by comparing incoming data with the current state and only updating when changes are detected. This efficiency is bolstered by the use of memoization, which minimizes redundant processing and ensures the interface remains smooth, even while polling for new data every second. These combined strategies result in an application that is not only feature-rich but also stable and efficient in its operation.

2.4.1. Dashboard Screen

The dashboard provides an immediate, live overview of the entire system. It displays real-time color-coded data from all the sensors splitting them into two categories for quick identification. The nutrient solution panel features details for the pH, EC, Temperature and Level of the solution, meanwhile around the tower the environmental temperature, humidity, light and CO_2 are displayed. Beside those, a button presents the main pump's status, while allowing the user to toggle the pump's state. To further enhance the intuitive nature of this layout, a dynamic graphical representation of the physical tower, showing the current water level and animating the water flow when the pump is active has been implemented (Figure 9).

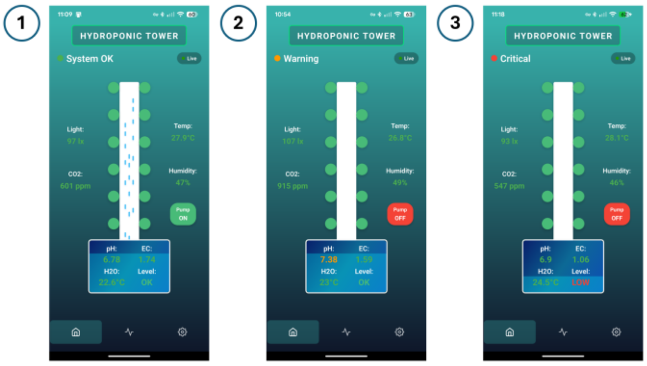
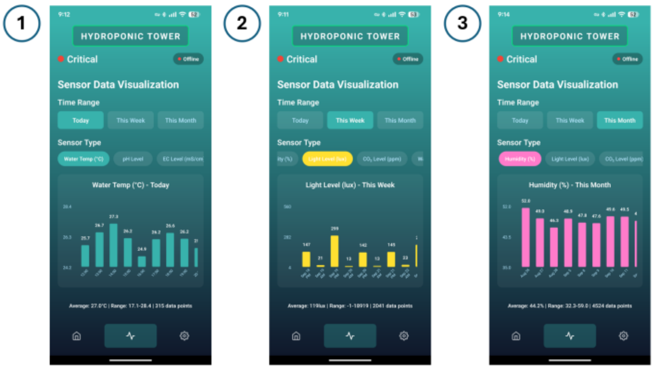


Figure 9. Dashboard screen displaying: 1. The main pump on and its flow animation; 2. The main pump off and a warning for the solution's high pH level; 3. A critical condition due to the tank level being too low (risk of running the pump dry).

2.4.2. Plotting Screen

When the user navigates to the plotting screen, the app connects to a Supabase cloud database to fetch and display historical sensor data in interactive charts. Afterwards, the user can select different sensors or view trends over various time ranges (as shown in Figure 10). At the bottom of the screen an average and the range of values for that time frame are presented, as well as the sum of the data points collected in the selected time span.

This feature is crucial for moving beyond simple monitoring, to data-driven cultivation. By analyzing historical trends, users can understand how different environmental factors interact, correlate changes with plant health, and make informed decisions to optimize future growth cycles.



Average: 27.0°C | Range: 17.1-28.4 | 315 data points

Average: 119lux | Range: -1-18919 | 2041 data points

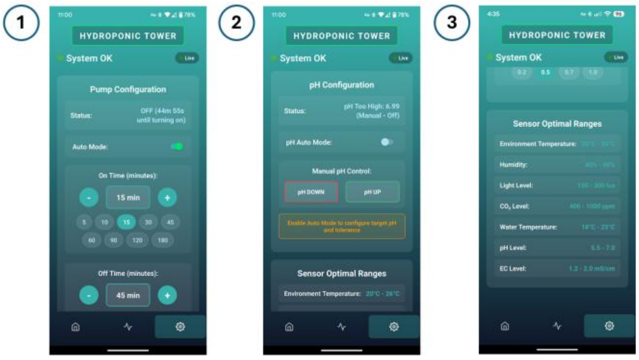
Average: 44.2% | Range: 32.3-59.0 | 4524 data points

Figure 10. Plotting screen displaying: 1. The nutrient solution temperature for that day; 2. The average ambient light every 12 hours (AM/PM) for that week; 3. The daily average humidity for that month.

Note: Since this feature uses a cloud database it is naturally accessible from the entire internet. However, this functionality can easily be extended for all features of the Smart Hydroponic Tower, by exposing the IP address of the ESP32. This can be achieved by either configuring port forwarding with the assistance of your local network administrator, or by setting up an IP tunneling service (such as ngrok [19]), on a computer that remains consistently connected to the local network.

2.4.3. Settings Screen

The settings screen empowers the user to fine-tune the system's automated behaviors. This includes configuring the on/off cycle times for the main circulation pump and setting the parameters for the automated pH control system (target pH and tolerance). Letting the user set not just a target value but also a tolerance helps prevent the system from over-correcting and oscillating, ensuring a more stable environment for the plants. Similarly with the webpage, the user can easily switch between automated and manual control, offering a complete override capability when needed. Below all those options, by default, the optimal ranges for growing leafy greens (Appendix A.1) are displayed (Figure 11). All the above provide flexibility to adapt the system to different plant types, growth stages, or nutrient solutions.



1

HYDROPONIC TOWER

System OK — Live

Pump Configuration

Status: OFF (44m 55s until turning on)

Auto Mode:

On Time (minutes): - 15 min +

5 10 15 30 45 60 90 120 180

Off Time (minutes): - 45 min +

2

HYDROPONIC TOWER

System OK — Live

pH Configuration

Status: pH Too High: 6.99 (Manual - Off)

pH Auto Mode:

Manual pH Control: pH DOWN pH UP

Enable Auto Mode to configure target pH and tolerance

Sensor Optimal Ranges

Environment Temperature: 20°C - 26°C

3

HYDROPONIC TOWER

System OK — Live

0.2 0.5 0.7 1.0

Sensor Optimal Ranges

Environment Temperature:	20°C - 26°C
Humidity:	40% - 60%
Light Level:	150 - 300 lux
CO_2 Level:	400 - 1000 ppm
Water Temperature:	18°C - 23°C
pH Level:	5.5 - 7.0
EC Level:	1.2 - 2.0 mS/cm

Figure 11. Setting screen displaying: 1. The configuration menu for the main pump scheduling; 2. The pH control menu with the manual override enabled; 3. The optimal value ranges for a hydroponic cultivation of leafy greens.

3. Results

The Smart Hydroponic Tower prototype was successfully assembled, and performance tests were executed on the hydraulic and electronic parts that were described in the second section of this article. The prototype was evaluated over a continuous 30-day period under typical indoor conditions to validate the functionality of its hardware and software. All control parameters (pump on/off schedule, pH target and tolerance) were left at their default values unless otherwise stated (main pump: 15 min ON / 45 min OFF cycle; pH target 5.5 – 6.5).

3.1. Hardware validation

3.1.1. Irrigation system

The irrigation system based on the 360° sprinkler nozzles provided uniform nutrient solution distribution horizontally, across all plant positions at the perimeter of each section, and vertically, across all levels. Additionally, the expandable arrangement (described in section 2.1.2) allowed straightforward build-up of the tower by simply extending the daisy chain to additional sections.

Collection tests carried out on day 15 and day 30 showed a coefficient of variation (CV) of outlet distribution between 7–9%, which is within acceptable limits for small-scale hydroponics. Furthermore, the main pump maintained adequate pressure to drive the entire chain of nozzles, with flow measurements averaging 38.6 ± 1.7 L/h per nozzle (close to the manufacturer’s rating of 40 L/h). Pressure losses along the 6 mm tubing runs were negligible over the tested lengths (up to six levels), confirming the suitability of this configuration for modular expansion. Ultimately, no significant dry spots were observed, even at the highest level of the tower.

3.1.2. Sensor accuracy and stability

Sensor performance was evaluated by comparing the system’s sensors to calibrated handheld/reference instruments where applicable. For each sensor we report the mean error, standard deviation (SD) of the error, and the root-mean-square error (RMSE). Notably, the pH and electrical conductivity (EC) sensors exhibited high accuracy for hydroponic control, with mean errors of less than 0.05 pH units and less than 0.05 mS/cm, respectively (see Table 2). Slight drift for the pH sensor was observed over multi-day runs (≈0.05 pH/day). However, periodic manual recalibration as the manufacturer suggests mitigates this [20].

Table 2. Sensor performance (prototype vs reference instruments).

Sensors	Reference Method	Range Tested	Mean Error	SD	RMSE
Water temperature (DS18B20)	Reference thermometer	15 – 30 °C	– °C	– °C	– °C
pH (DFR SEN0161)	Laboratory benchtop pH meter	4.0 – 7.5 pH	– pH	– pH	– pH
EC (DFR0300)	Commercial EC meter (with temp. compensation)	0.2 – 2.5 mS/cm	– mS/cm	– mS/cm	– mS/cm
Air temperature / humidity (DHT22)	Refence thermometer / specialized hygrometer	15 – 30 °C 30 – 80 %RH	– °C – %RH	– °C – %RH	– °C – %RH
Ambient Light (BH1750)	Lux meter	10 – 10,000 lux	– lux	– lux	– lux
CO$_2$ (MH-Z19B)	NDIR reference instrument	200 – 2,000 ppm	– ppm	– ppm	– ppm

3.1.3. Pump scheduling and mechanical reliability

The states of the main circulation pump and peristaltic pH pumps were regularly assessed for operational reliability throughout the 30-day test period through random checks. For both pump systems, the physical state matched the one scheduled by the ESP32 across 100% of logged cycles. The peristaltic pumps, after 30 days of intermittent dosing, showed no visible wear or clogging; Repeatably dosing the same volume per burst ±4% (measured gravimetrically). At first, air bubbles did reduce dosing precision but adjusting the tube orientation and fixing them in place eliminated this problem. In short, the pump control and scheduling logic proved to be robust for continuous operation, however, longer endurance tests are suggested for commercial deployment.

3.1.4. Power Consumption

Power draw was measured with a handheld multimeter/amp meter at different points during the system's operation (Table 3).

Table 3. Power consumption during different operational states.

State	Power draw
Idle	0.95 W
Main pump active	14.4 W
Peristaltic pump active	1.8 W
Main + peristaltic active	15.25 W
Average over a duty cycle	4,33 W

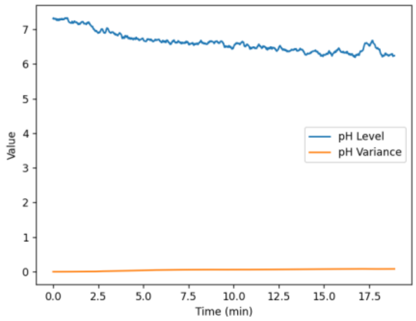
The system's power footprint is modest and compatible with common 12 V DC supplies. For off-grid installations, an inverter/solar + battery should be sized to accommodate pump peak loads.

3.2. Software validation

3.2.1 Automated pH control performance

The pH control loop was tested by artificially raising and lowering the reservoir pH in controlled increments and observing the system response (activation of dosing pumps, convergence time, and overshoot).

Writers note: Figure with a representative 30-minute pH correction trace should be included showing (a) initial induced deviation, (b) dosing events (peristaltic pump on), and (c) settling into the target window. Plot the pH signal with vertical markers where dosing occurred and annotate time-to-target.



Target window 5.8 – 6.2 pH
Mean time-to-target after dosing event – mins
Overshoot (max) observed after correction – pH

Interpretation. The closed-loop controller rapidly returned pH to the target window with minimal overshoot. Short dosing bursts with enforced cooldown periods prevented oscillation. The response time and small steady-state deviation demonstrate the system's suitability for maintaining nutrient availability for leafy greens.

3.2.2. Data logging, latency, and availability

During the trial period, the data logging functionality was unfortunately not operational for the full 30 days of this trial period. Nevertheless, in the last 7 days, all the 2016[1] intended logged samples were received, achieving a 100% success rate, despite temporary Wi-Fi dropouts. Short 1or 2 minutes outages did occur due to local network instability, however, the retry feature for failed uploads managed to successfully backfill the data upon reconnection.

[1] By default the ESP32 is configured to upload a sample of all sensor data to the cloud database every 5 mins. This consistent logging translates to 288 samples per day, 2,016 samples per week and a total of 8,640 samples per month.

3.2.3. Usability, UI and user feedback

A small informal usability test (n = 5 participants with gardening experience) was performed to evaluate the webpage and mobile app UI (Table 4 & Table 5). Participants rated the system on a 5-point Likert scale (1 = very poor, 5 = excellent).

Table 4. Webpage UI evaluation.

Feature	Average Grade
Live monitoring	4.4
Main pump controls	4.6
pH pump controls	4.6
Data logging controls	4.2
Responsiveness	4.0
Overall	4,36

Table 5. Mobile app UI evaluation.

Feature	Average Grade
Dashboard Clarity	5.0
Main pump controls	4.8
pH pump controls	4.8

Historical Plots	4.6
Responsiveness	4.6
Overall4	4.76

Overall, these findings suggest both the system’s digital interfaces provide an accessible and user-friendly experience. However, the mobile app achieved consistently higher ratings, with a 15% increase in perceived responsiveness and the dashboard clarity receiving a perfect score of 5.0. Participants highlighted the intuitive layout and real-time visualization of system status as well as the presets on the pump configuration menus as particularly helpful. While the mobile app and web interface were positively evaluated, the informal usability study involved a small participant sample; larger, more diverse testing would provide stronger evidence of user acceptance.

4. Discussion

This study demonstrated that a low-cost IoT-enabled hydroponic tower can effectively automate nutrient circulation and pH control, while providing real-time monitoring through both web and mobile platforms. The proposed design proved to have uniform irrigation, accurate sensor performance, and reliable pH stabilization, confirming its capabilities to support sustainable plant growth. Moreover, the prototype successfully achieved timely problem detection and long-term monitoring, enabling users to make proactive adjustments and optimize plant care. These findings extend previous hydroponics research, which often focused on nutrient delivery but left historical data analysis and remote decision-making limited to high-end commercial systems.

In addition to enhancing operational efficiency, the system's digital interfaces were positively evaluated, with participants favoring the mobile app for its clarity and responsiveness. This reflects trends in smart farming where ease of use is a decisive factor for adoption, particularly by small-scale or hobbyist growers who may lack technical expertise. The system's small footprint makes it suitable for urban centers yet, importantly, the modest energy footprint (≈4.3 W average) makes the design compatible with renewable off-grid setups too. This aligns with the sustainability goals of vertical hydroponic farming, which already offers significant savings in land and water use compared to conventional agriculture.

Nonetheless, limitations remain. The trial was short-term and restricted to leafy greens under indoor conditions. The lifespan of low-cost pH and EC probes may affect long-term reliability, and usability testing also involved only a small sample. Future work should extend testing to multiple crop types and growth cycles, explore predictive analytics for nutrient management, and enhancements in sensor durability. Additionally, expanding the system’s IoT capabilities through integration with smart greenhouse infrastructures or city-scale IoT networks could unlock opportunities for large-scale deployment in urban agriculture.

Overall, the Smart Hydroponic Tower exemplifies how affordable, open-source, and modular technologies can bridge the gap between hobbyist hydroponics and commercial smart farming. By lowering technical and financial barriers to entry, such systems could accelerate the democratization of vertical farming, especially in urban contexts where space and resources are limited. Ultimately, this work contributes toward the broader objective of ensuring sustainable food production for a rapidly growing global population.

5. Conclusions

This section is not mandatory but can be added to the manuscript if the discussion is unusually long or complex.

This work presents the design, development, and evaluation of a low-cost, IoT-enabled Smart Hydroponic Tower that integrates real-time sensing, automated nutrient circulation, and pH control within a modular, 3D-printed structure. The system successfully demonstrated uniform irrigation, reliable pump scheduling, accurate sensor performance, and effective closed-loop pH regulation. Its digital ecosystem—comprising a cloud-connected database, web interface, and mobile application—provides users with both immediate control and long-term data insights, thereby lowering the technical barriers of entry for hydroponic farming. Experimental validation confirmed that the tower can operate continuously under typical indoor conditions, with modest power requirements compatible with renewable off-grid setups. Informal usability testing further highlighted the clarity and responsiveness of the mobile application, underscoring the importance of intuitive interfaces for adoption among non-expert users.

While limitations remain —such as short-term testing, reliance on low-cost sensors with potential drift, and evaluation restricted to leafy greens— the findings establish a strong foundation for future work. Longer-term trials, testing with diverse crops, predictive analytics for nutrient management, and integration with larger-scale smart agriculture infrastructures could further enhance system robustness and scalability.

Overall, the Smart Hydroponic Tower exemplifies how affordable, open-source technologies can democratize vertical farming practices, making them accessible to hobbyists, urban growers, and small-scale producers. By addressing both environmental sustainability and food security challenges, this system contributes to the advancement of smart agriculture and the broader goal of sustainable food production in a rapidly urbanizing world.

6. Patents

This section is not mandatory but may be added if there are patents resulting from the work reported in this manuscript.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/doi/s1, Figure S1: title; Table S1: title; Video S1: title.

Author Contributions: Conceptualization, E.P.; methodology, S.P. and V.Z.; software, V.Z.; validation, V.Z.; formal analysis, V.Z.; investigation, V.Z.; resources, E.P.; data curation, V.Z.; writing—original draft preparation, V.Z.; writing—review and editing, S.P. and V.Z.; visualization, V.Z.; supervision, S.P.; project administration, E.P, funding acquisition, E.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The source code for the ESP32 firmware and the React Native application is available in the project's GitHub repository.

Acknowledgments: We would like to thank Sofia Faliagka from the Laboratory of Agricultural Constructions and Environmental Control of the University of Thessaly for her contributions to our study. During the preparation of this article, the author(s) used Chat-GPT 5 for the purposes of generating text. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

VHF	Vertical Hydroponic Farming
ACS	Autonomous Control Systems
IoT	Internet of Things
LCD	Liquid Crystal Display
PCB	Printed Circuit Board
IPS	In-Plane Switching
IDE	Integrated Development Environment
ADC	Analog to Digital Converter
IP	Internet Protocol
REST	Representational State Transfer
API	Application Programming Interface

Appendix A

Appendix A.1

The appendix is an **not yet written** optional section that can contain details and data supplemental to the main text—for example, explanations of experimental details that would disrupt the flow of the main text but nonetheless remain crucial to understanding and reproducing the research shown; figures of replicates for experiments of which representative data is shown in the main text can be added here if brief, or as Supplementary data. Mathematical proofs of results not central to the paper can be added as an appendix.

Table A1. This is a table caption.

Title 1	**Title 2**	**Title 3**
entry 1	data	data
entry 2	data	data

Appendix B

All appendix sections must be cited in the main text. In the appendices, Figures, Tables, etc. should be labeled starting with "A"—e.g., Figure A1, Figure A2, etc.

References

[1] P. A. Valavanidis, "Feeding a Growing World Population.".

[2] K. H. Mohd Azmi, N. A. Mohamed Radzi, and A. Ahmad, "THE FUTURE OF SUSTAINABLE AGRICULTURE: A REVIEW OF IOT AND AUTONOMOUS CONTROL IN VERTICAL HYDROPONIC FARMING," *Adv. Electr. Electron. Eng.*, vol. 22, no. 2, pp. 146–162, June 2024, doi: 10.15598/aeee.v22i2.5321.

[3] A. Hossain *et al.*, "Agricultural Land Degradation: Processes and Problems Undermining Future Food Security," in *Environment, Climate, Plant and Vegetation Growth*, S. Fahad, M. Hasanuzzaman, M. Alam, H. Ullah, M. Saeed, I. Ali Khan, and M. Adnan, Eds., Cham: Springer International Publishing, 2020, pp. 17–61. doi: 10.1007/978-3-030-49732-3_2.

[4] A. K. Jha, A. Kumar, K. Kumari, and A. Kumar, "VERTICAL FARMING: THE FUTURE OF FOODS," vol. 3, pp. 1–4, Apr. 2025.

[5] J. B. J. Jr, *Hydroponics: A Practical Guide for the Soilless Grower*, 2nd ed. Boca Raton: CRC Press, 2016. doi: 10.1201/9780849331671.

[6] H. Singh, N. Puri, and N. Mishra, *Sustainability Reporting and Blockchain Technology*. Taylor & Francis, 2024.

[7] D. I. Pomoni, M. K. Koukou, M. G. Vrachopoulos, and L. Vasiliadis, "A Review of Hydroponics and Conventional Agriculture Based on Energy and Water Consumption, Environmental Impact, and Land Use," *Energies*, vol. 16, no. 4, p. 1690, Jan. 2023, doi: 10.3390/en16041690.

[8] C. Vatistas, D. D. Avgoustaki, and T. Bartzanas, "A Systematic Literature Review on Controlled-Environment Agriculture: How Vertical Farms and Greenhouses Can Influence the Sustainability and Footprint of Urban Microclimate with Local Food Production," *Atmosphere*, vol. 13, no. 8, p. 1258, Aug. 2022, doi: 10.3390/atmos13081258.

[9] H. M. Resh, *Hydroponic Food Production: A Definitive Guidebook for the Advanced Home Gardener and the Commercial Hydroponic Grower*, 8th ed. Boca Raton: CRC Press, 2022. doi: 10.1201/9781003133254.

[10] T. Sangeetha and E. Periyathambi, "Automatic nutrient estimator: distributing nutrient solution in hydroponic plants based on plant growth," *PeerJ Comput. Sci.*, vol. 10, p. e1871, Feb. 2024, doi: 10.7717/peerj-cs.1871.

[11] K. Bouarroudj, F. Babaa, and A. Touil, "IoT-based monitoring and control for optimized plant growth in smart greenhouses using soil and hydroponic systems," *Internet Things*, vol. 33, p. 101710, Sept. 2025, doi: 10.1016/j.iot.2025.101710.

[12] P. P. Ray, "Internet of things for smart agriculture: Technologies, practices and future direction," *J. Ambient Intell. Smart Environ.*, vol. 9, no. 4, pp. 395–420, June 2017, doi: 10.3233/AIS-170440.

[13] G. Rajaseger *et al.*, "Hydroponics: current trends in sustainable crop production," *Bioinformation*, vol. 19, no. 9, pp. 925–938, Sept. 2023, doi: 10.6026/97320630019925.

[14] "ESP32 Wi-Fi & Bluetooth SoC | Espressif Systems." Accessed: Sept. 22, 2025. [Online]. Available: https://www.espressif.com/en/products/socs/esp32

[15] "ESP32-DevKitC V4 - ESP32 - — esp-dev-kits latest documentation." Accessed: Sept. 24, 2025. [Online]. Available: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/user_guide.html

[16] E. Bogatin, *Bogatin's Practical Guide to Prototype Breadboard and PCB Design*. Artech House, 2021.

[17] R. Jain, "Getting Started with the ESP32 Development Board," Play with Circuit. Accessed: Sept. 24, 2025. [Online]. Available: https://playwithcircuit.com/getting-started-with-esp32/

[18] "vgzoidis/Smart_Hydroponic_Tower." Accessed: Sept. 23, 2025. [Online]. Available: https://github.com/vgzoidis/Smart_Hydroponic_Tower

[19] "Putting the ESP32 microcontroller on the Internet - ngrok." Accessed: Sept. 24, 2025. [Online]. Available: https://ngrok.com/blog-post/putting-the-esp32-microcontroller-on-the-internet

[20] "wiki.dfrobot.com/PH_meter_SKU__SEN0161_#More Documents." Accessed: Sept. 27, 2025. [Online]. Available: https://wiki.dfrobot.com/PH_meter_SKU__SEN0161_#More%20Documents

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.