

Research Article

Intelligent Management of Hydroponic Systems Based on IoT for Agrifood Processes

D. Hostalrich,¹ J. Pelegri-Sebastia ,² T. Sogorb,² and V. Pellicer¹

¹Research, Development and Innovation Department, Venalsol S.L., Palma Gandia, Spain

²Sensors and Magnetism Group, Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Campus de Gandia, Universitat Politècnica de Valencia, 46730 Grao de Gandia, Spain

Correspondence should be addressed to J. Pelegri-Sebastia; jpelegri@upv.es

Received 27 September 2022; Revised 17 November 2022; Accepted 29 November 2022; Published 12 December 2022

Academic Editor: Yuan Li

Copyright © 2022 D. Hostalrich et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There are a wide variety of new microprocessors that are easy to program and configure to perform complex tasks, and with the right features, sensors, and additional mechanisms, we can prepare them to monitor and take care of the crops with automated processes. Soil moisture, air temperature, humidity, CO₂, and water level are some of the most basic parameters to monitor with sensors, but any type of sensor can be added if the signal is adapted so that the microprocessor can read it. The data read from the sensors allow us to control and automate processes using relays connected to a variety of external components like illumination, refrigeration, and irrigation systems. We present a solution to the environmental monitoring hydroponic system based on IoT. The developed device is a low powered, and the data obtained is transmitted via Zigbee to a central system where we can configure and control all the devices paired, so it is relatively easy to implement and escalate.

1. Introduction

The innovation in control systems for agriculture using IoT devices comes from the need to improve the production and reduce the waste of energy and resources, making it more economically accessible [1–3]. With IoT devices, we can deploy a mesh of devices with various essential sensors that allows us to monitor and control the crops in a fast, easy, and scalable way, archiving a big amount of data about all the crops [4, 5]. With all the data obtained, we can create processes that will take care of the crops according to their necessities (variable photoperiod, irrigation schedule, temperature control, ventilation, etc.) [6, 7], archiving precision farming [8–10], reducing the waste of energy and resources. This also leads us to find the best photoperiod or automation to produce a better crop performance [11, 12], as we have all the data we need. Linking processes followed by the obtained

results, we can improve the system iteratively, fulfilling the need to grow more food for the population, which is growing more and faster [13, 14].

A wireless sensor network (WSN) is the perfect technology to implement with these emergent IoT technologies, as it satisfies the necessity to make the system simple, easy to deploy, scalable, and dynamic [15–17]. For a few years, we have seen how different studies developed WSN based on IoT for smart agriculture. Somov et al. and Miorandi et al. presented a deployment of a WSN based on the Libelium IoT products for smart agriculture for the control and monitoring of a tomato greenhouse where IA and cloud computing technologies were applied [18, 19]. Montoya-Munoz and Rendon also used the Libelium IoT products among others like Intel and Omicron to make an approach to fog computing for providing reliability in data collection using IoT devices in a Colombian coffee smart farm [20]. Other studies

TABLE 1: Related works.

Reference	Description	Technology	Sensors/systems
[22]	Mesh-like configuration system with nodes that can control sensors and act as actuators for external systems like valves	(i) Transmission protocol: NRF24L01 (between sink node and the rest of nodes) and Wi-Fi (between sink node and server) (ii) Sensor node devices: ATmega microcontroller (iii) Raspberry sink node (iv) Server for data storage and web application	(i) Soil moisture (ii) Temperature and humidity
[18]	IoT deployment for monitorization and control of a tomato greenhouse using a wireless sensor network, cloud computing, and artificial intelligence	(i) Transmission protocol: Zigbee (ii) Sensor node devices: ATmega microcontrollers (iii) Libelium gateway (iv) Web server for data storage, retrieval and visualization growth modeling, and optimal planning Monitoring and visualization	(i) EC (electric conductivity) (ii) Flow solution (iii) Temperature (iv) PAR (photosynthetically active radiation) (v) Humidity (vi) CO ₂ (vii) Heating and ventilation system (viii) High pressure sodium bulbs
[20]	Approach using IoT-fog-cloud architecture using machine learning to detect outliers and another	(i) Omicron system Intel system Libelium system	(i) Temperature (ii) Relative humidity (iii) Light intensity (iv) Soil moisture
Our proposal	Deployment of mesh-like nodes using IoT devices to read sensors, control external system, and a LabVIEW interface to configure and monitor the nodes	(i) Transmission protocol: custom protocol using Zigbee (ii) Sensor node devices: ATmega microcontrollers (iii) Web server for data storage and LabVIEW for control and visualization	(i) Ambient temperature and humidity (ii) Soil moisture (iii) CO ₂ (iv) Water level (v) Illumination system (vi) Refrigeration system (vii) Ventilation system

are starting to use the 5G network, which is fast and has a wide coverage, making it perfect for smart farming using AI-powered robots and cloud computing for data processing and real-time monitoring [21]. The Agrinex system [22] connects sensor nodes that will take the measurements with a sink node using a nrf24l01 module (that uses a free frequency band) connected to a server and web application, acting as a bridge.

As we can see in Table 1, the use of state-of-the-art technologies in this area is being adopted and implemented quickly. But there is a gap that needs to be solved: the security and ease to use by any end user. The system we are going to talk about is like the previous commented systems in the IoT section (hardware and transmission proto-

col). But unlike the other systems, we are using a custom communication protocol, which makes the device pairing easy and protects the communication between the devices with a custom encryption protocol. However, the most important thing is the interface, with which the user will interact to configure and control the controllers, photoperiods, and automations in an easy way.

As an experimental test, we have planted two different varieties of blueberries. The plants are located on a special reservoir with good isolation from the outside. This reservoir is prepared for setting up grow lights, ventilation, refrigeration, and irrigation systems. A connector interface has been placed on the reservoir, so we can place the controller outside (Figure 1). The parameters to be measured are ambient



FIGURE 1: Crop encapsulation.

temperature and humidity, soil moisture, CO₂ in the air, and the water level of the irrigation tanks.

The controller is designed to connect all the sensors and external systems needed in any connector intended for that use, adapting the device to our needs.

The data will be transmitted from the controller (one or more) to the coordinator. The coordinator will manage all the data and send it to the server, which will manage and store it in the database. The server will send commands to the coordinator to manage the photoperiods and automations that the user has configured (Figure 2).

2. Materials and Methods

2.1. Sensor Characterization. All the chosen sensors for the system are low powered and use the same operating and supply voltage, because we want to be able to connect almost every sensor to any connector. But there are specific connectors depending on the communication protocol the sensor uses (UART, i2c, simple analog/digital connection, etc.).

In this experiment, we are testing if the data obtained from our sensors is reliable. The CO₂ sensor is a MH-Z16 (using UART protocol); the temperature and humidity sensor is an AM2305 with digital output, a simple resistive analog soil moisture sensor, and a XKC-Y25-NPN touchless sensor for the water lever on the tanks.

The connectors on the controller are SP16 IP68, and the encapsulation of almost all the sensors has been modified to make all connections to be watertight.

2.2. Communication Protocol. The wireless communication protocol is one of the cores of this project. In this case, we are designing a scalable and easy to deploy system (Figure 3), so the communication protocol between the coordinator and the controllers must be wireless [15].

There are many options to choose for wireless communication (Wi-Fi, LoRa, Bluetooth, and Zigbee). However, the best choice for this project is the Zigbee protocol. Zigbee is a set of protocols based on the IEEE 802.15.4 standard [23, 24]. This protocol was made to meet the need for low power consumption, easy to implement, and scalable. In fact, the scalability is its best quality, because it uses a mesh topology which interconnects the devices dynamically, creating multiple paths. This is the reason why a Zigbee-based system is easy to deploy and scale.

This design is very common to see on monitoring and control systems based on IoT in agriculture. It is relatively easy to develop using the actual technologies [22]. In this case, the followed architecture is a four-layered one (application, processing, transport, and perception) [25, 26]:

- (i) Application: the IoT application of the system, like monitor and control the crops
- (ii) Processing: how the data is stored, processed, and visualized
- (iii) Transport: the communication protocol used
- (iv) Perception: the physical devices and the interaction between them

For now, we are using one controller for the experimental test, but we have made tests to check that the system works correctly by adding more controllers.

As you can see in Figure 4, before sending the data via Zigbee, the data is encrypted by our custom protocol and decrypted when received. In this case, the coordinator and the controller can be both, sender and receiver.

2.3. Controller. A device is based on an Atmel microcontroller, where all the sensors and external systems are connected. It has two inputs for UART sensors, four inputs for simple analog/digital sensors, one input for i2c sensor or devices, four outputs connected to one relay each one for external systems, and one more connector for the power supply.

The controller is connected also to a Zigbee module to communicate with its paired coordinator (Figure 5). A led will give basic information about the status of the device.

We can pair several controllers with the coordinator using a custom pairing method developed for this system (Figure 6).

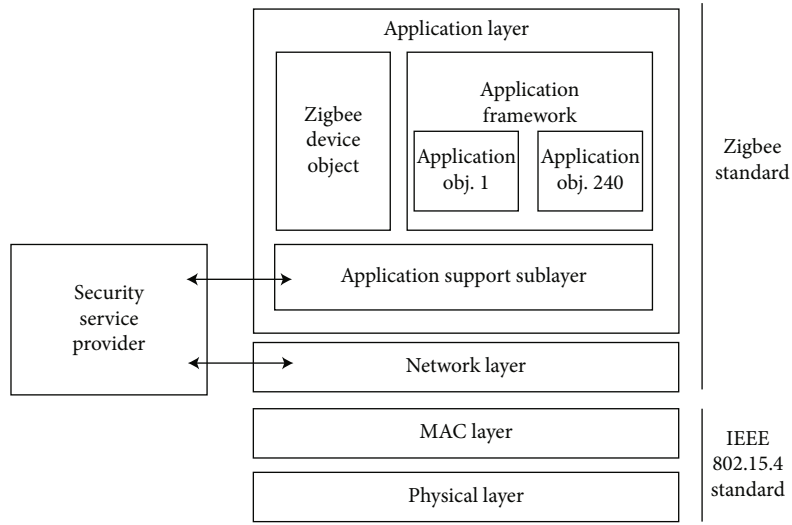


FIGURE 2: Zigbee protocol stack.

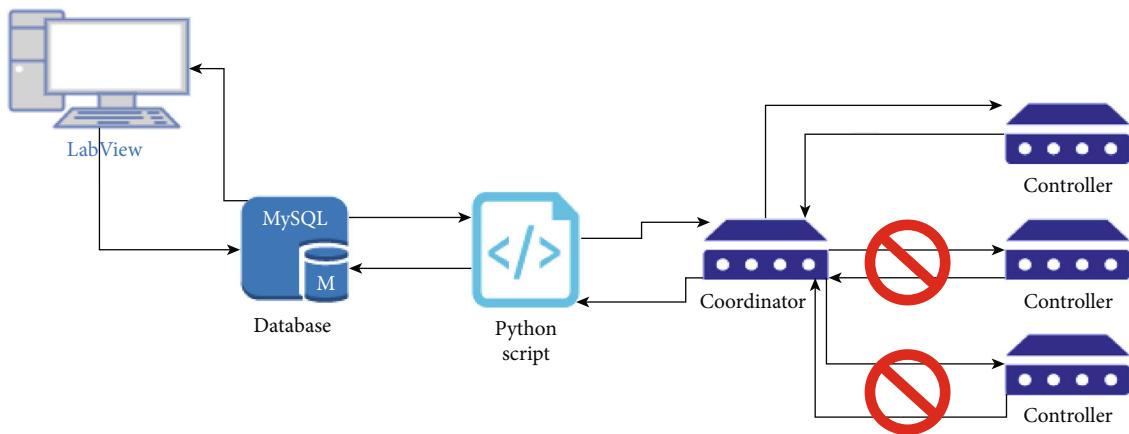


FIGURE 3: System communication.

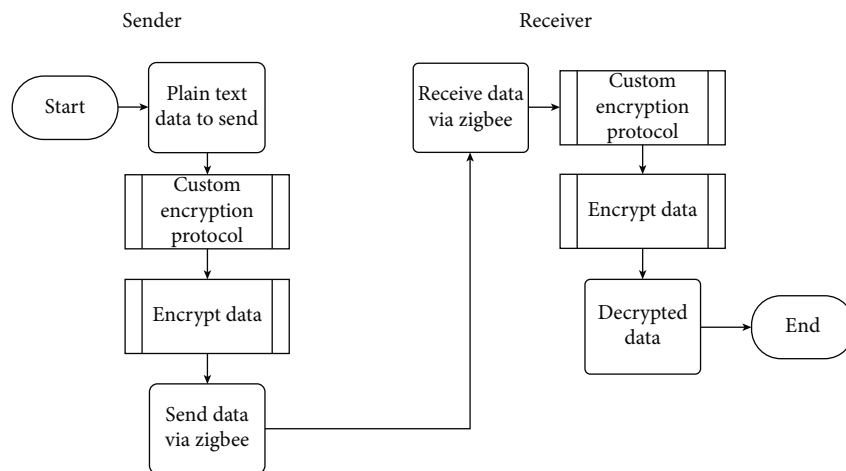


FIGURE 4: Communication between sender and receiver.

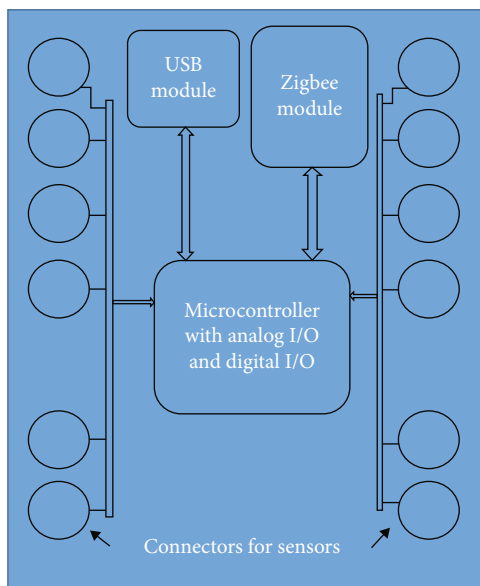


FIGURE 5: Scheme of the designed controller system. Where we have the sensor connectors, the microcontroller with analog and digital inputs and outputs, the Zigbee communication module, and a USB module for power and programing the controller.

2.4. Coordinator. Another device is based on an Atmel microcontroller. This one only has a Zigbee module to communicate with its paired controllers and a USB port to communicate with a PC (Figure 7). This device will be the bridge between all the controllers and the server.

The advantage of the custom protocol designed is that the pairing with the controllers is performed automatically when the coordinator is powered on, so the user only waits for the pairing process to end (the end of the process is notified when the LED of the coordinator stops flashing).

2.5. Server. Last, we have the PC that in this case will be on the server and control panel. It sends and receives data from the coordinator and stores the data obtained in the database. In addition, it executes a graphic interface made with LabVIEW, where we can monitor (Figure 8) all the data and configure all the controllers paired individually, as well as the schedules, photoperiods, and automations that we want to implement. In the background, a Python script will be running to manage all the data and communication between the coordinator and the PC (Figure 9). All the configurations and operations made with LabVIEW are stored in the database, so the Python script will notice and run them.

3. Results and Discussion

3.1. Experimental Measurements. For testing, we have used only one controller with a digital temperature and humidity sensor, an analog soil moisture sensor, one CO₂ UART sensor, and four external devices: a grow light, a water pump for the irrigation, and refrigeration and ventilation systems.

The controller has been configured using LabVIEW, as well as the photoperiod to control the light system and two automations to control the irrigation, ventilation, and refrigeration systems. Both the illumination and the automations are executed following a schedule configured by the user.

For monitoring all the data, we have prepared a window on LabVIEW (the viewer) where we have chosen the data we want to monitor. In this case, we are monitoring the data from all the sensors and external devices connected to the controller. By default, the controller executes only the operations to manage the photoperiods and the automations stored in it. To send the commands to the coordinator and read the sensors from a controller, we have to open the monitoring window on LabVIEW, where we can choose the controller to monitor and its sensors and photoperiods/automations configured. LabVIEW will create all the necessary commands and store them in the database. The Python script will send the commands stored in the database automatically to the coordinator.

With all the configurations done, the pipeline will be the next:

- (i) The controller will execute all the automations stored automatically, reading the necessary sensors. The photoperiod will be controlled by the Python script, which will send the commands to turn on and off the light (Figure 8)
- (ii) The viewer on LabVIEW will generate the commands every certain time and will be stored on the database. The Python script will send them to the coordinator
- (iii) The automations will also work with schedules created by the user, so the script will generate the commands to enable or disable them following the schedule associated

The controller has a little memory which stores the actual state of the photoperiod, the automations configured, and the sensor and the external systems that it has connected to it. Therefore, if there is an unexpected shutdown of the system, the controller will be able to reset itself and work again. In the case we want to change the control of any external system/device to manual mode, we can do it with the viewer on LabVIEW, being able to turn on or turn off them. We can turn on the automatic control again if we want.

4. Results

With the data stored in the database, we can visualize the values obtained from the sensors.

With the readings obtained from the ambient humidity/temperature sensor placed within the reservoir, we can see that before the configured photoperiod turns on the lights, the ambient temperature is decreasing, reaching a minimum of 16.2°C while the ambient humidity is stable at ≈62%. This happens because with

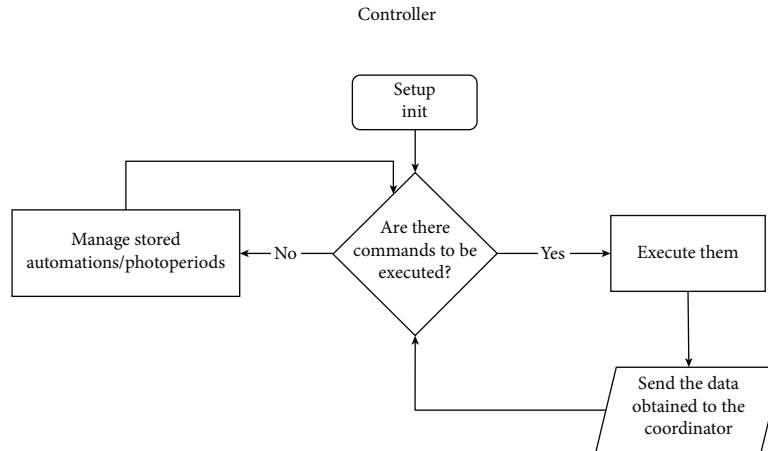


FIGURE 6: Controller flow chart.

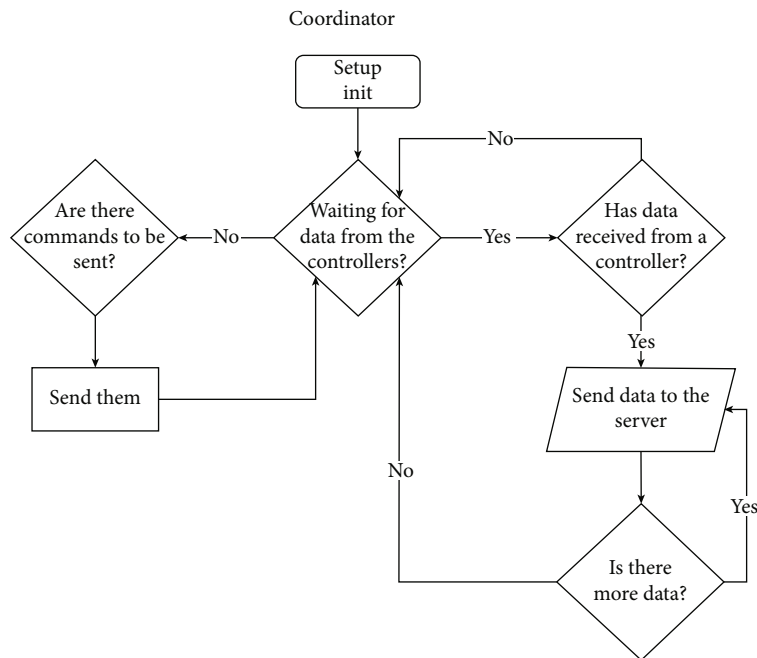


FIGURE 7: Coordinator flow chart.

nightfall, the temperature decreases, but the humidity of the reservoir is maintained since it remains isolated from the outside. When the photoperiod turns on the light, the temperature increases dramatically, reaching a maximum of 25.5°C while the humidity drops considerably to a minimum of 38.8% (Figures 10 and 11). The heat emitted by the light is causing this variation. When the photoperiod turns off the light again, we see how the temperature starts decreasing and the humidity starts increasing again.

We can also see a variation of the temperature and humidity when the temperature reaches the limit of 25°C (Figure 10 and Table 2). This happens because we have con-

figured an automation that turns on the ventilation system and the refrigeration system when the temperature sensor detects that the temperature inside the reservoir is equal or higher than 25°C. When this happens, the temperature suddenly drops below 25°C again and the ventilation and refrigeration systems stop (because the temperature is now below the threshold that triggers them). This series of events causes us to see this saw shape on the chart.

When the ventilation and refrigeration systems are turned on, they achieve their goal, but we see that the temperature drops too much (reaching a drop of more than 1°C), so a drop in the power of the refrigeration system should be considered.

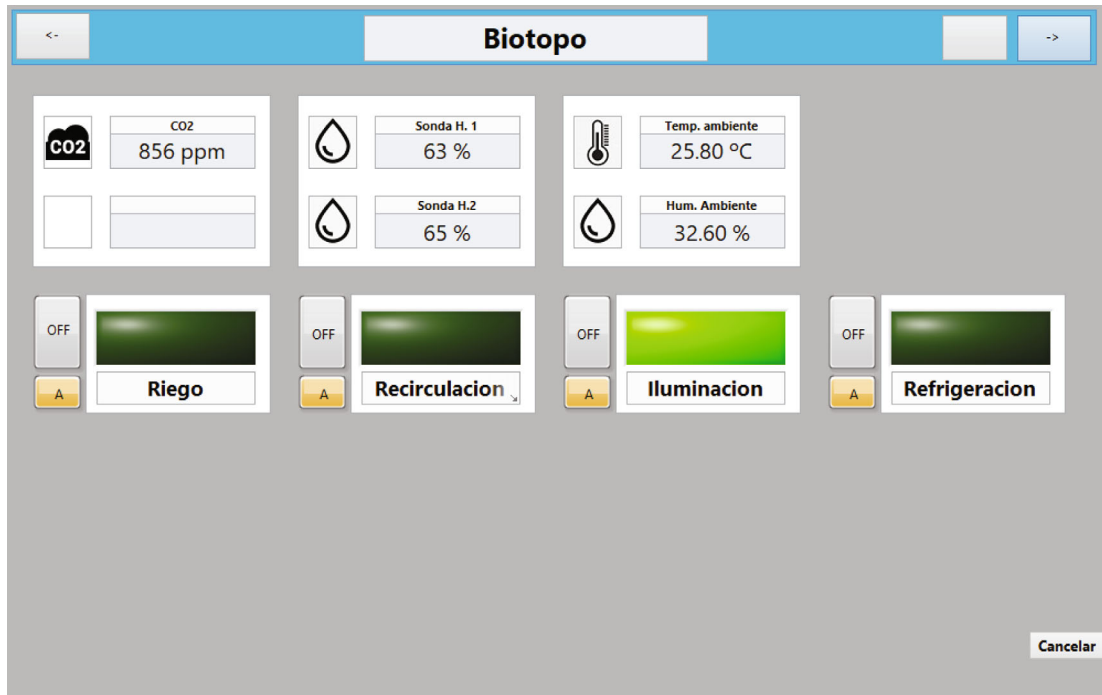


FIGURE 8: LabVIEW monitoring interface.

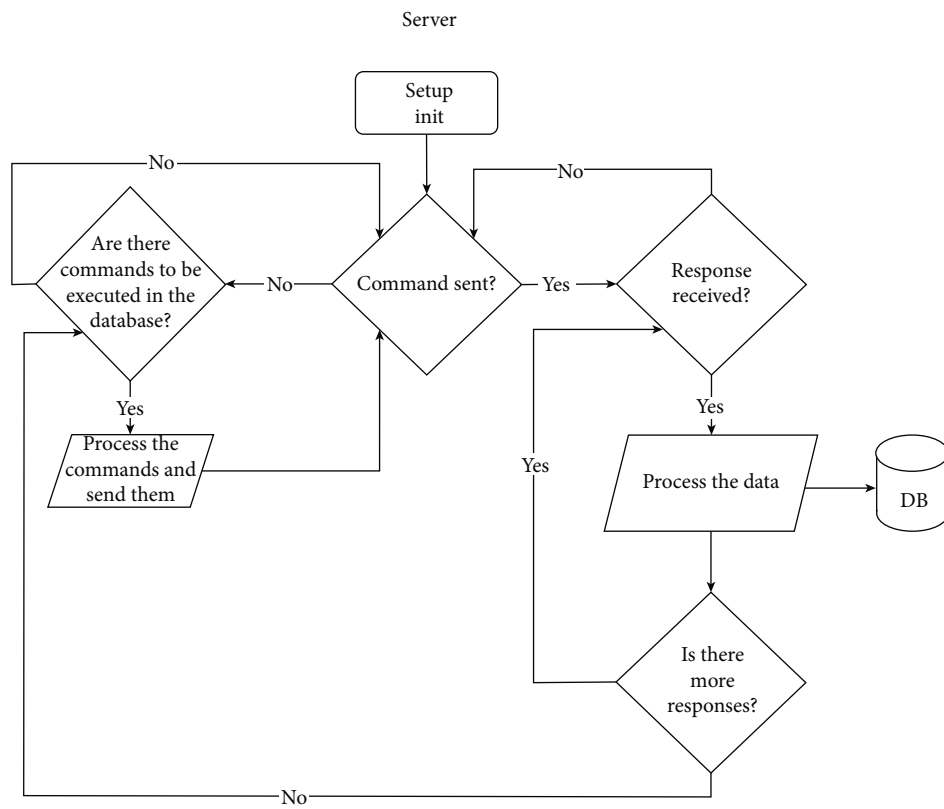


FIGURE 9: Server flowchart.

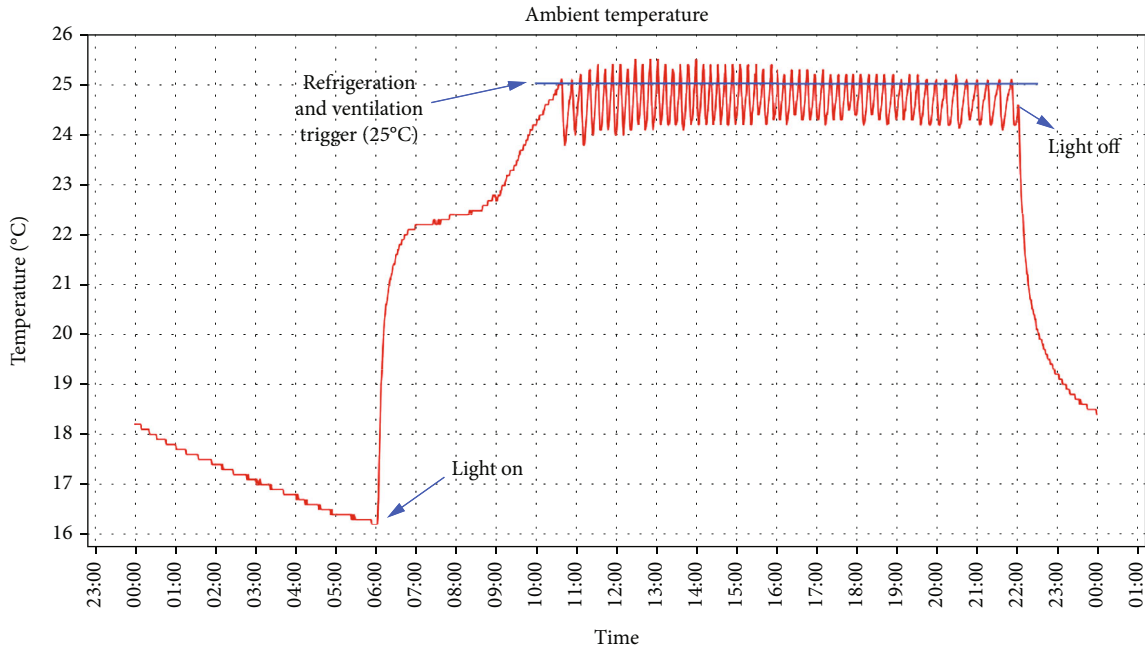


FIGURE 10: Ambient temperature readings.

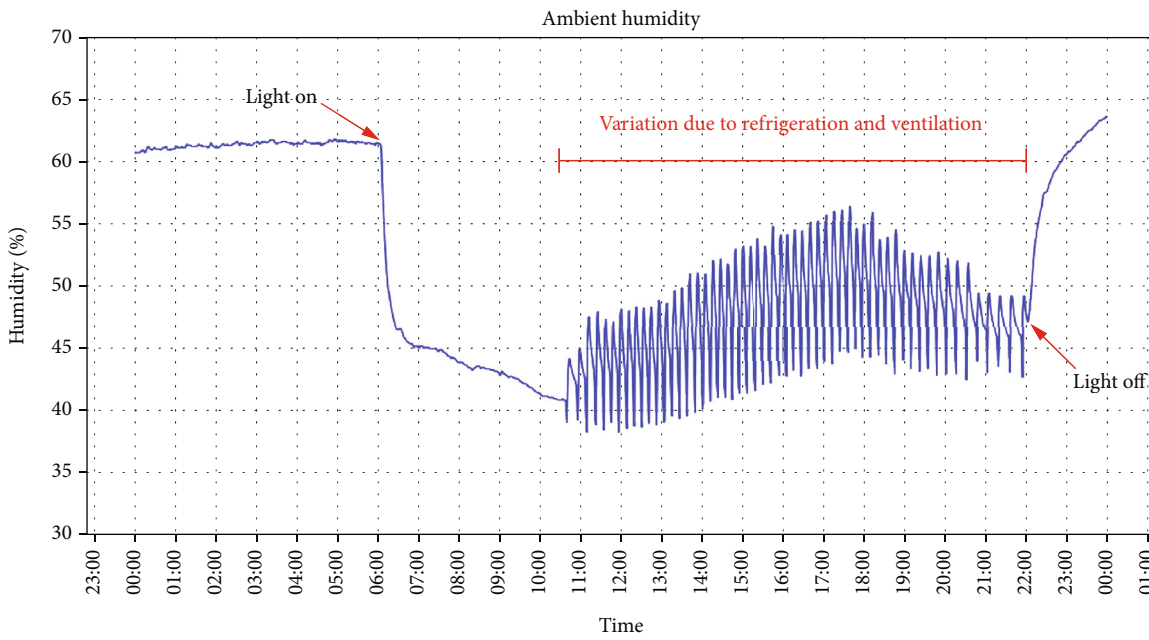


FIGURE 11: Ambient humidity readings.

TABLE 2: Ambient temperature and humidity results.

	Max. temp. (°C)	Min. temp. (°C)	Max. humidity (%)	Min. humidity (%)
Light off (00:00 to 06:00)	18.2	16.2	61.8	60.7
Light on (06:00 to 22:00)	25.5	23.8	56.4	38.2
Light off (22:00 to 00:00)	24.6	18.4	63.6	47.1

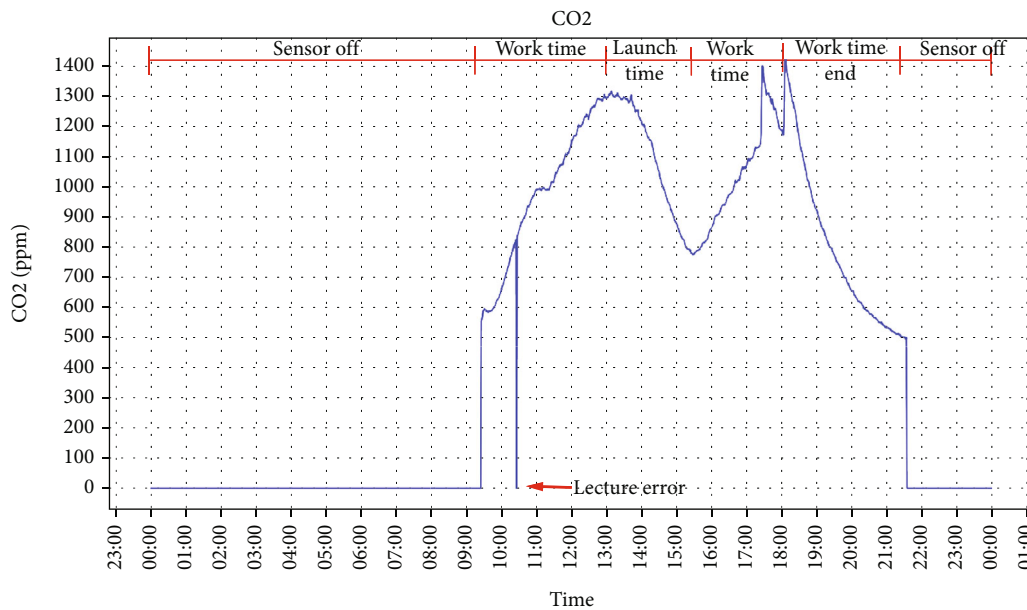


FIGURE 12: CO₂ level readings.

TABLE 3: Maximum and minimum values of CO₂.

	CO ₂ (ppm)
Max. value	1420
Min. value	502

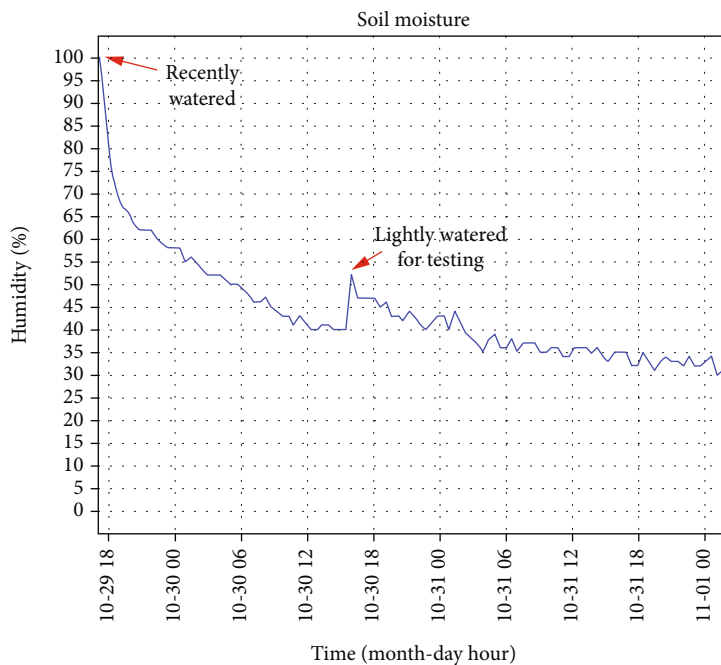


FIGURE 13: Soil moisture readings.

We also see how the humidity varies too much when these systems are activated, something that could affect the health of plants, so it would be interesting to try not to activate the ventilation system together with the refrigeration system.

These results also help us to see how the configured automations have successfully activated the ventilation and refrigeration systems.

The data from the CO₂ sensor (Figure 12) is taken from the outside of the reservoir. The sensor was placed outside because the controller was not able to read the data from the sensor when it was placed inside. We are currently investigating this issue, but the most probable reason is that the extra connectors to interconnect the sensor inside the reservoir with the controller placed outside and the length of the cable are corrupting the signal. For this reason, we have decided to test the sensor taking readings from the office where the reservoir is placed to test the sensor itself.

As shown in the graph, we can see how the ppm (parts per million) of CO₂ in the air increases when there are people working in the office. The breath of the workers and the computers are causing this increase. When it is lunchtime and the office empties, the level of CO₂ decreases to a minimum of 774 ppm and increases again when the lunchtime is over, reaching 1420 ppm. At 18:00, all people leave the office, and the computers are off, causing the CO₂ level to decrease again, reaching 502 ppm before the sensor turns off (Table 3). Considering the situation in which the sensor has been tested, the data obtained is what you would expect, so there should be no problems implementing it correctly once the reading error problem has been solved.

The data from the soil moisture (Figure 13) were taken at a time interval of approximately two and a half days. Recently watered, it starts with 100% of soil moisture. The fast initial decrease of the soil moisture is due to the heat caused by the light. When the photoperiod turns off the light, the soil moisture decreases more slowly. When the soil moisture is below 50%, approximately, the decrease slows down and the heat of the light does not affect as much as it did at the beginning because the surface of the soil is already dry, while the soil moisture is kept below it.

5. Conclusions

This system design allows us to approach a wide variety of situations, making it a great solution to monitor and control the crops. It can be adapted to the most common necessities, being easy to configure and deploy, relatively cheap, and low powered. In addition, based on the data obtained, we can improve the system by modifying all the automations and photoperiods, achieving better efficiency.

Despite the system is a good solution to automate and monitor the crops, it has its limitations. The number of connectors is limited, and even if the user can configure where to connect each sensor, depending on its communication protocol, it must be connected to a compatible connector. Also, the power supply of all the sensors must be 5V, because it is the voltage that the connectors provide.

Another factor to consider is that if there are a lot of controllers, the number of coordinators should be increased too.

If a coordinator has too many controllers under its control, the system will slow down since the communication with its controllers is carried sequentially.

But on the other hand, the user can adapt the system to its necessities. Since the device is low powered, it can be attached to small batteries and portable solar panels and placed easily. Even with the described limitations, the customization that can be applied to the system is higher than the other similar systems mentioned. For example, you can configure the controllers to connect only one type of sensor, like soil moisture sensors, or configure it to work with several different sensors, like we did in our tests. The user only has to wait for the coordinator to perform the automatic pairing process to establish communication with the controllers nearby and then configure the system as he wishes using the graphic user interface designed with LabVIEW.

In addition, the ability to configure custom photoperiods and automations with custom schedules in an easy way for the user is the core of the system to automate the crop and achieve a better performance compared to the other systems on the market.

For now, the results of the test with the blueberries are satisfactory. The growth stage is finished, and now, the photoperiod has changed to the next step, the flowering stage, where the plant should start growing the fruit. Nevertheless, we have noticed some things. First, the soil moisture sensors have lost precision since their first implementation. After checking the sensor, it seems that the cause of the precision decrease is due to the electrolysis corrosion. This happens because the sensor is always powered on, so the exposition to a moist environment causes this effect [27]. To solve this, the best solution is to power off the sensor when it is not taking any measurement.

Another effect, we have noticed is the redness on the leaves of the blueberries. This is due to the proximity of the luminaire and its infrared LED. The plant changes the color of its leaves to adapt itself to the environment conditions. Turning the leaves red causes the infrared component of the light to be rejected. To solve this, we will move the luminaire further away from the plant and its spectral range will be managed with the controller (the luminaire has this function).

The length of the cable that connects the sensor to the controller is also something to take care of, because it can affect the reliability of the data obtained, especially with analog sensors like the soil moisture sensor. One sensor that has been particularly affected is the CO₂ sensor, which seems that, due to the length of the cable and the extra connector of the reservoir, the data is not reaching the controller properly, causing reading errors.

For future improvements, apart from solving the problems found, we are adding more sensors to have a better precision, like a light micromole sensor, or sensors for the maintenance of the system like a flood sensor, with the objective to make this project a solution for a wide variety of situations with different needs.

Data Availability

The reviewer can obtain more information in the web page <https://venalsol.com/es/content/ledcultivo-innovacion>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the I+D+i Program of the Generalitat Valenciana, Spain (AEST/2018/010).

References

- [1] E. Navarro, N. Costa, and A. Pereira, "A systematic review of IoT solutions for smart farming," *Sensors*, vol. 20, no. 15, p. 4231, 2020.
- [2] A. Walter, R. Finger, R. Huber, and N. Buchmann, "Opinion: smart farming is key to developing sustainable agriculture," *Proceedings of the National Academy of Sciences*, vol. 114, no. 24, pp. 6148–6150, 2017.
- [3] P. P. Ray, "Internet of Things for smart agriculture: technologies, practices and future direction," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 4, pp. 395–420, 2017.
- [4] D. Pivoto, P. Waquil, E. Talamini, C. Spanhol, V. Corte, and G. Mores, "Scientific development of smart farming technologies and their application in Brazil," *Information Processing in Agriculture*, vol. 5, 2018.
- [5] H. R. Zhang, Z. L. Li, T. F. Zou, X. Y. Wei, and G. C. Yang, "Overview of agriculture big data research," *Computer Science*, vol. 41, no. 11A, pp. 387–392, 2014.
- [6] A. Stefanos, D. K. Nikolidakis, and D. Dimitrios, "Energy efficient automated control of irrigation in agriculture by using wireless sensor networks," *Computers and Electronics in Agriculture*, vol. 113, pp. 154–163, 2015.
- [7] J. Lee, S. Kim, S. Lee, H. Choi, and J. Jung, "A study on the necessity and construction plan of the Internet of Things platform for smart agriculture," *Journal of Korea Multimedia Society*, vol. 17, no. 11, pp. 1313–1324, 2014.
- [8] Y. Kim, R. Evans, and W. Iversen, "Remote sensing and control of an irrigation system using a distributed wireless sensor network," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 7, pp. 1379–1387, 2008.
- [9] E. C. Leonard, "Precision agriculture," in *Encyclopedia of Food Grains*, vol. 4, pp. 162–167, Elsevier, Amsterdam, The Netherlands, 2016, ISBN 9780123947864.
- [10] J. M. Talavera, L. E. Tobón, J. A. Gómez et al., "Review of IoT applications in agro-industrial and environmental fields," *Computers and Electronics in Agriculture*, vol. 142, pp. 283–297, 2017.
- [11] H. Zhang, X. Wei, T. Zou, Z. Li, and G. Yang, "Agriculture big data: research status, challenges and countermeasures," in *Computer and Computing Technologies in Agriculture VIII. CCTA 2014*, D. Li and Y. Chen, Eds., vol. 452 of IFIP Advances in Information and Communication Technology, Springer, Cham, 2015.
- [12] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming - a review," *Agricultural Systems*, vol. 153, pp. 69–80, 2017.
- [13] R. Gorli and A. Professor, "Future of smart farming with Internet of Things," *Journal of Information Technology and its Applications*, vol. 2, no. 1, 2017.
- [14] United Nations, *Department of Economic and Social Affairs; Population Division. World Population Prospects 2019*, Highlights; United Nations Department for Economic and Social Affairs, New York, NY, USA, 2019.
- [15] L. Hamami and B. Nassereddine, "Application of wireless sensor networks in the field of irrigation: a review," *Computers and Electronics in Agriculture*, vol. 179, p. 105782, 2020.
- [16] R. Pahuja, H. K. Verma, and M. Uddin, "A wireless sensor network for greenhouse climate control," *IEEE Pervasive Computing*, vol. 12, no. 2, pp. 49–58, 2013.
- [17] S. Sadowski and P. Spachos, "Wireless technologies for smart agricultural monitoring using internet of things devices with energy harvesting capabilities," *Computers and Electronics in Agriculture*, vol. 172, p. 105338, 2020.
- [18] A. Somov, D. Shadrin, I. Fastovets et al., "Pervasive agriculture: IoT-enabled greenhouse for plant growth control," *IEEE Pervasive Computing*, vol. 17, no. 4, pp. 65–75, 2018.
- [19] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.
- [20] A. I. Montoya-Munoz and O. M. C. Rendon, "An approach based on fog computing for providing reliability in IoT data collection: a case study in a Colombian coffee smart farm," *Applied Sciences*, vol. 10, no. 24, p. 8904, 2020.
- [21] T. Yu, S. Dananjayan, C. Hou, Q. Guo, S. Luo, and Y. He, "A survey on the 5G network and its impact on agriculture: challenges and opportunities," *Computers and Electronics in Agriculture*, vol. 180, p. 105895, 2021.
- [22] N. M. Tiglaou, M. Alipio, J. V. Balanay, E. Saldivar, and J. L. Tiston, "Agrinex: A low-cost wireless mesh-based smart irrigation system," *Measurement*, vol. 161, p. 107874, 2020.
- [23] S. Safaric and K. Malaric, "ZigBee wireless standard," in *Proceedings ELMAR*, pp. 259–262, Zadar, Croatia, June 2006.
- [24] C. Muthu, M. Shanmugaraj, and R. Prabakaran, "Study on Zig-Bee technology," in *2011 3rd international conference on electronics computer technology*, pp. 297–301, Kanyakumari, India, April 2011.
- [25] J. Chen and A. Yang, "Intelligent agriculture and its key technologies based on Internet of Things architecture," *IEEE Access*, vol. 7, pp. 77134–77141, 2019.
- [26] Telecommunication Standardization Sector of ITU Recommendation ITU-T Y, *Overview of the Internet of Things*, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, 2060.
- [27] B. Mccollum and K. H. Logan, "Electrolytic corrosion of iron in soils," *Proceedings of the American Institute of Electrical Engineers*, vol. 32, no. 7, pp. 1515–1575, 1913.