Alternatives of IoT Irrigation Systems for the Gardens of Arequipa

https://doi.org/10.3991/ijim.v15i22.22653

Jose Esquicha-Tejada(✉), Juan Copa-Pineda
Universidad Nacional de San Agustín de Arequipa, Arequipa, Peru
jesquichat@unsa.edu.pe

Abstract—To resolve the drinking water scarcity problem in cities around the world, innovative proposals have been proposed to automate garden irrigation at homes and reduce drinking water consumption. In this study, a survey was conducted on 68 inhabitants of the Region of Arequipa, Peru to determine common habits in the irrigation of gardens. From these data, two systems were implemented in two average gardens using the Arduino Uno board (integrated with Ethernet Shield) and NodeMCU. Each system integrates soil moisture sensors, water flow sensors, and actuators, such as solenoid valves and relays. In addition, the information was centralized through an Internet of things system (Home Assistant or Adafruit IO). This made it possible to compare the two systems used in the two gardens, generating a discussion on the advantages and disadvantages of each proposal and the capability of each system to save potable water in the irrigation of plants.

Keywords—IoT, Arduino, NodeMCU, garden, Raspberry Pi

1 Introduction

Drinking water is scarce worldwide. According to the Aquae foundation [1], only 0.007% of the water on earth is drinkable, and that amount is reducing every year due to increasing pollution and disproportionate consumption. Consequently, the United Nations states that 2.1 billion people lack clean water services [2]. Therefore, several research teams worldwide have proposed projects that take advantage of rainwater to generate drinking water [3],[4],[5].

Technological advances are currently improving the quality of people's lives. The Internet of things (IoT), with its systems of sensors, actuators, and interfaces, has been providing solutions to different tasks, facilitating the execution of various processes [6]. The message queue telemetry transport (MQTT) protocol is used to communicate between computers connected to the Internet, and it is frequently used for IoT projects [7]. Currently, this protocol is taught and used in various applications [8].

The aforementioned protocol can be integrated with various development boards to implement creative projects for different purposes. One example is the use of Arduino in automated watering of gardens [9],[10]. Another example is the use of NodeMCU for home automation [11] or in a drip irrigation system [12].

In this article, alternatives for IoT systems that integrate Home Assistant and Adafruit IO with Arduino, NodeMCU, and/or Raspberry Pi development boards are proposed with the objective of studying the irrigation of gardens in the city of Arequipa, because of its importance in preserving the Arequipa countryside [13], which has been declared a World Heritage Site by UNESCO. This research should benefit the control of drinking water, helping to protect the socio-economic future of Arequipa.

2 Problem statement

In the city of Arequipa, Peru, water scarcity is a frequent problem every year because the availability of this valuable resource depends on the rainfall from January to March. This rainfall is essential for filling reservoirs, which, after the water is treated, provide drinking water for consumption throughout the city [14]. Thus, it is necessary to make the population aware of the necessity of rational use of drinking water in their homes and the preservation of the Arequipa countryside through green areas in each home. Based on the results of this research, it is proposed that consumption of drinking water be reduced by being rational when using automated systems in the irrigation of plants in house gardens; this would help preserve the ecosystem of the city of Arequipa.

In this research, a survey was conducted whose sample size from a finite population was calculated using the following equation [15].

$$n = \frac{N * Z^2 * p * q}{d^2 * (N-1) + Z^2 * p * q} \tag{1}$$

where p is the percentage of individuals who have a characteristic, q is the percentage of individuals who do not have the characteristic, Z^2 is the number of units of deviation that indicates the confidence level, which is squared (90% is used, with a value of $Z = 1.645$), N is the population size, d is the sampling error, which is squared (90% is used with a value of $d = 0.1$), and n is the sample size of the population.

According to a previous report [16], the population (N) of the city of Arequipa in 2017 was 980,221 inhabitants. Taking all the data, the number of respondents required for the survey is calculated as

$$n = \frac{980221 * 1.645^2 * 0.5 * 0.5}{0.1^2 * (980221 - 1) + 1.645^2 * 0.5 * 0.5}$$

$$n = 67.65 \text{ respondents}$$

Thus, a survey was carried out with 68 inhabitants of the city of Arequipa to obtain data related to the management of gardens in homes. Figure 1 shows the results of the survey for frequency of garden irrigation, showing that 34% irrigate weekly, 32% irrigate every two days, and 18% irrigate every three days. This reflects the varied consumption of drinking water in different gardens.

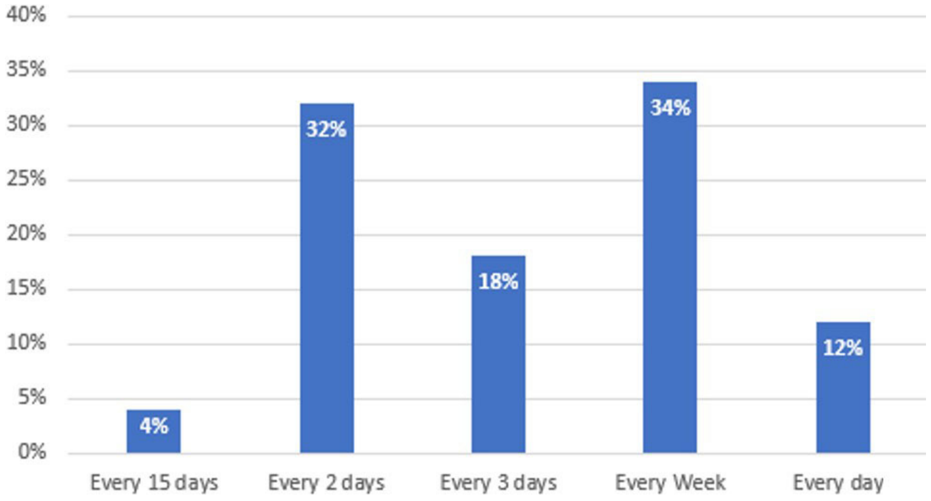

Fig. 1. Survey result (garden watering time)

Figure 2 shows results obtained for the number of automated systems for watering gardens among the Arequipa population, revealing that 90% do not have automated gardens; only 10% of the house gardens have automated systems. This shows that the majority of the Arequipa population water their gardens in the traditional way.

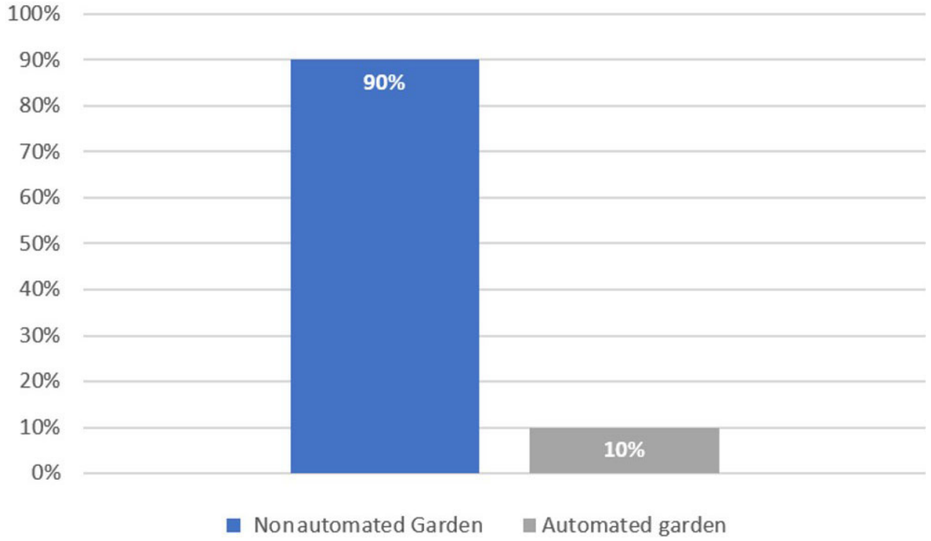

Fig. 2. Results of the survey (automated gardens)

Figure 3 shows the types of irrigation system used by the Arequipa population, with 36% irrigating by flooding or gravity irrigation. This type of irrigation system is inefficient owing to the high consumption of potable water.

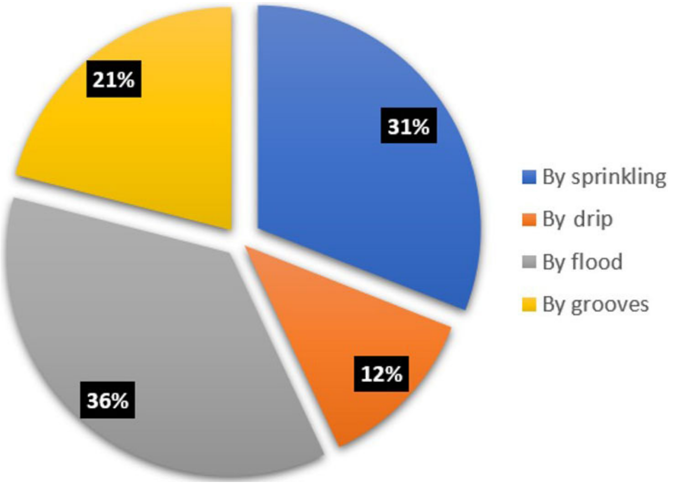

Fig. 3. Results of the survey (type of irrigation)

3 Automated garden irrigation system

Understanding the problems in the management of gardens in the Arequipeña population, four alternatives are proposed for the automated irrigation of gardens using either a Arduino or a NodeMCU board and two alternative IoT platforms (Home Assistant or Adafruit IO); the Home Assistant also requires a Raspberry Pi board. Both can save drinking water in Arequipeñas homes. The proposals work with the following components.

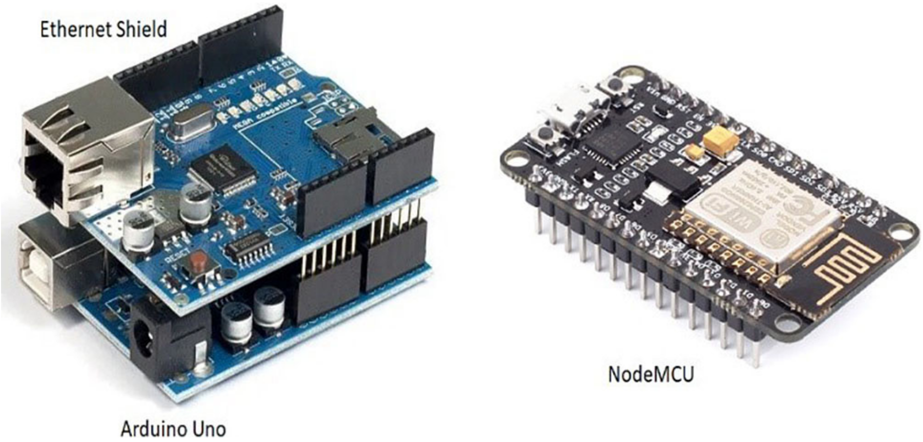

Fig. 4. Plate alternatives used for irrigation automation

- Adafruit IO is an IoT platform that easily integrates boards from Arduino, Raspberry Pi, ESP8266, and others [17].
- Home Assistant home automation system runs on Linux distributions [18].
- Hass.io is software that integrates with Home Assistant.
- Raspberry Pi is a low-cost, low-consumption computer that enables operating systems to run with an ARM architecture [18].
- Arduino Uno (see Figure 4) is a board that incorporates the ATmega326; it has six analog pins and 14 digital pins [19].
- Ethernet Shield (see Figure 4) is a board that acts as a shield for the Arduino board and makes Internet access possible through its RJ45 port [20].
- NodeMCU v3 Lolin (see Figure 4) is a board based on the ESP8266 Wi-Fi SoC, with Wi-Fi connectivity, one analog pin, and 17 digital pins [11].
- Relay, an electromechanical device used to switch circuits, works as a switch and is controlled by an electromagnet [21].
- Water flow sensor is a water flow sensor in pipes [22].
- Solenoid valve is an electrical mechanical device that controls fluid flow [23].

3.1 Option 1: using Arduino with Adafruit IO

As an alternative to an automated irrigation system, it is necessary to use the Arduino Uno board with its Ethernet Shield, which provides a wired Internet connection, as well as sensors (soil moisture sensor and water flow sensor) and actuators (relay and solenoid valve). They are configured in the Arduino integrated development environment (IDE), in addition to being integrated through the Adafruit IO IoT platform, which is hosted on the Internet (see Figure 5).

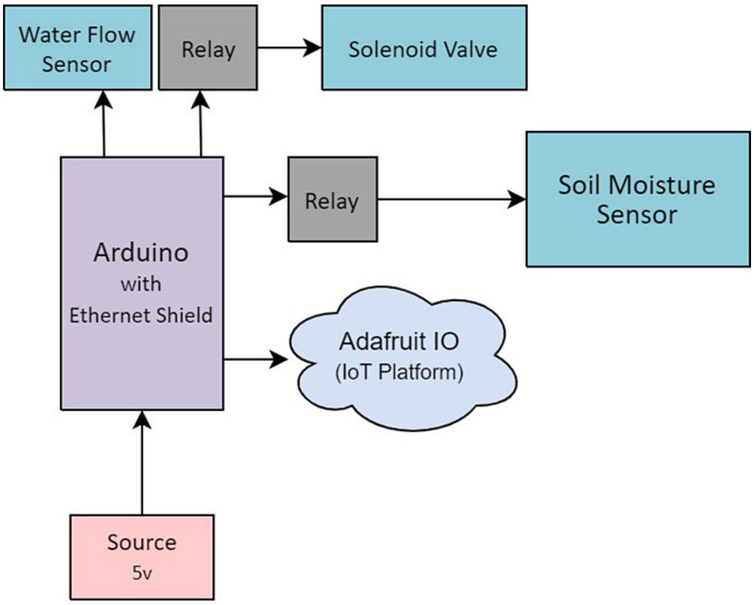

Fig. 5. Design of the option with Arduino and Adafruit IO

3.2 Option 2: using NodeMCU with Adafruit IO

As a second option, the NodeMCU board is integrated with a soil moisture sensor, water flow sensor, solenoid valve, and relay. These are integrated through the IoT Adafruit IO platform through the wireless connection of NodeMCU (see Figure 6).

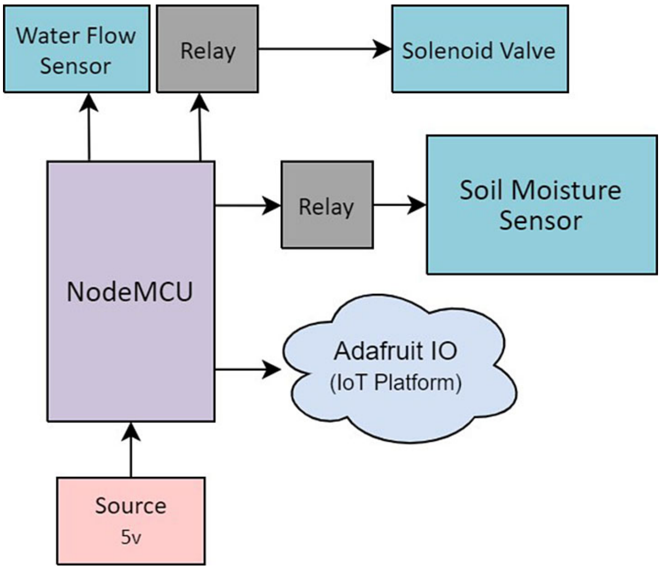

Fig. 6. Design of the option with NodeMCU and Adafruit IO

Figures 5 and 6 show that the relays are placed both in the solenoid valve and in the soil moisture sensor, because they control the passage of water or the operation of the soil moisture sensor; thus, there are energy savings and a longer life of the electronic components.

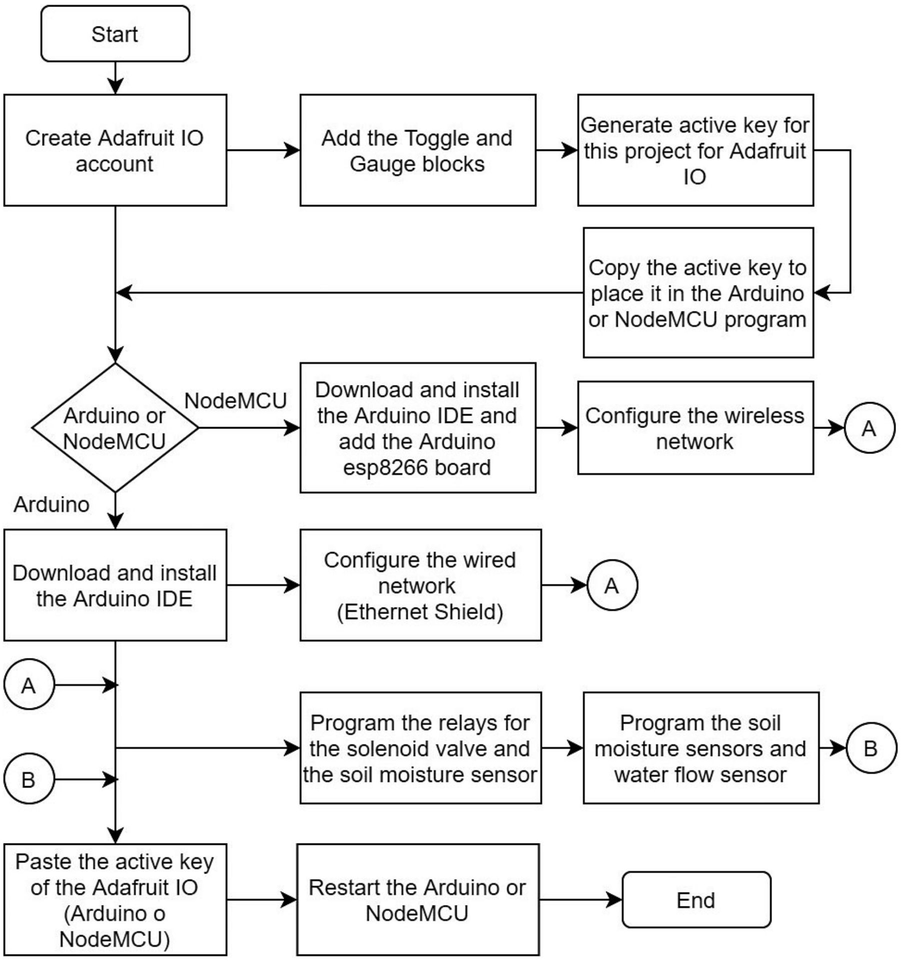
Start
Create Adafruit IO account
Add the Toggle and Gauge blocks
Generate active key for this project for Adafruit IO
Copy the active key to place it in the Arduino or NodeMCU program
Arduino or NodeMCU
NodeMCU
Download and install the Arduino IDE and add the Arduino esp8266 board
Configure the wireless network
A
Arduino
Download and install the Arduino IDE
Configure the wired network (Ethernet Shield)
A
A
B
Program the relays for the solenoid valve and the soil moisture sensor
Program the soil moisture sensors and water flow sensor
B
Paste the active key of the Adafruit IO (Arduino o NodeMCU)
Restart the Arduino or NodeMCU
End

Fig. 7. Adafruit IO, Arduino, and NodeMCU process diagram

3.3 Configuring the Adafruit IO of options 1 and 2

For the operation of the IoT platform in the first two options, it is necessary to initially create an account on the Adafruit IO main page, which makes it possible to use the dashboard and place the blocks (Toggle and Gauge) necessary to control the relays of the soil moisture sensors and the water flow sensor. Then, the active key of Adafruit IO is generated for later use in its integration with the security measures provided by the platform. Next, from the Arduino IDE, the necessary code instructions for the operation of the wired or wireless network must be placed. Then, the code instructions corresponding to the sensors and actuators must be placed so that, later, the active key generated previously is placed. Finally, the Arduino or NodeMCU is restarted for the integration of the automated irrigation system (see Figure 7).

3.4 Option 3: using Arduino with Home Assistant

As a third option, one can use Arduino Uno, with its Ethernet Shield, sensors (soil moisture sensor and flowmeter), and actuators (relay and solenoid valve). This alternative provides a local system that integrates Home Assistant installed on a Raspberry Pi board, which controls the relay to activate the solenoid valve. This option does not require an Internet connection for its operation (see Figure 8).

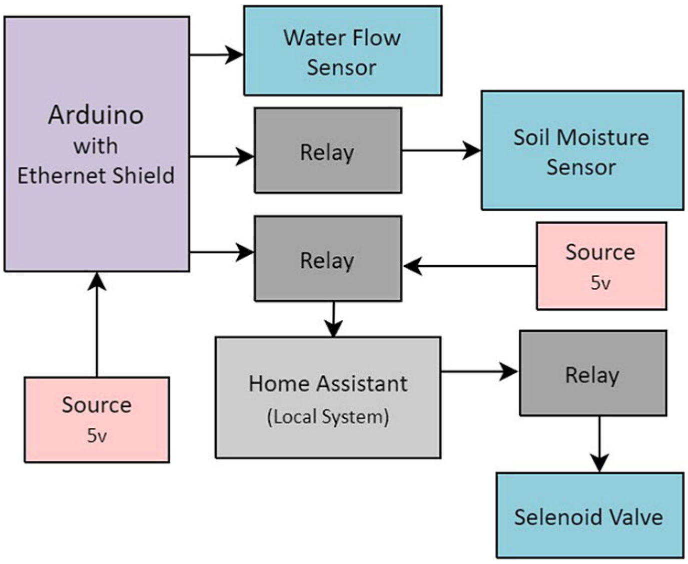

Fig. 8. Design of the Arduino and Home Assistant option

3.5 Option 4: using NodeMCU with Home Assistant

For the fourth option, as in the second option, NodeMCU is required, but with the difference that the Home Assistant system is used. It is mounted on a Raspberry Pi board to generate an automated plant irrigation system. This option does not require an Internet connection, only a local network connection, as in the previous option (see Figure 9).

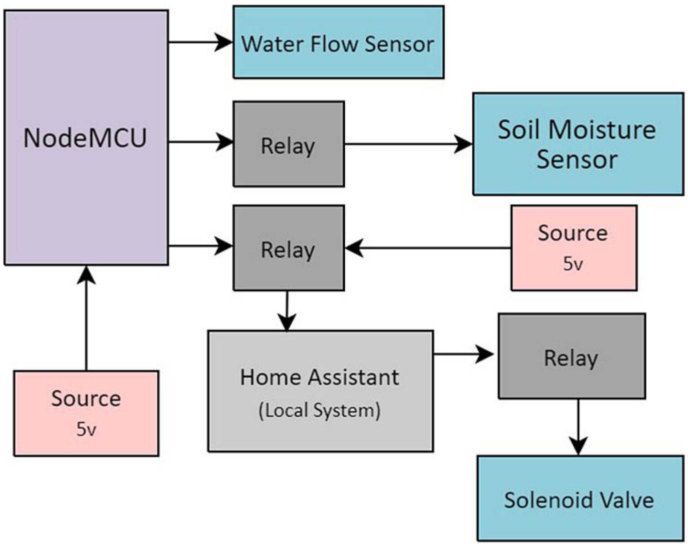

Fig. 9. Design of the NodeMCU and Home Assistant option

Figures 8 and 9 show options that require more energy because they have a local server installed. However, a personalized configuration can be made so that the local server is turned on when it is time to water the garden, making consumption moderate compared with the other options. Finally, similar to the first two options described earlier, it is necessary to use relays that control the solenoid valve and the soil moisture sensor.

3.6 Configuring Home Assistant, Arduino, and NodeMCU in options 3 and 4

The last two alternatives for automating plant irrigation utilize a Raspberry Pi board as a central server to host Home Assistant integrated with Hass.io [24]. After entering the web platform through a browser, the following add-on store is installed (see Figure 10).

- Configurator is a web application that provides a file system browser with a text editor [25].
- Mosquitto broker is an open-source message broker (EPL/EDL license) [26].
- Samba Share uses the SMB protocol to exchange files on different operating systems [27].

In Home Assistant, it is necessary to configure the soil moisture sensors and the water flow sensor [22],[28]; moreover, the Ethernet library that MQTT uses [29] must be added. The important libraries that must be used are Ethernet.h and PubSubClient.h. Then, to integrate Home Assistant with Arduino, the configurator.yaml file must be edited. Finally, the Home Assistant service must be restarted for Home Assistant to function properly (see Figure 10).

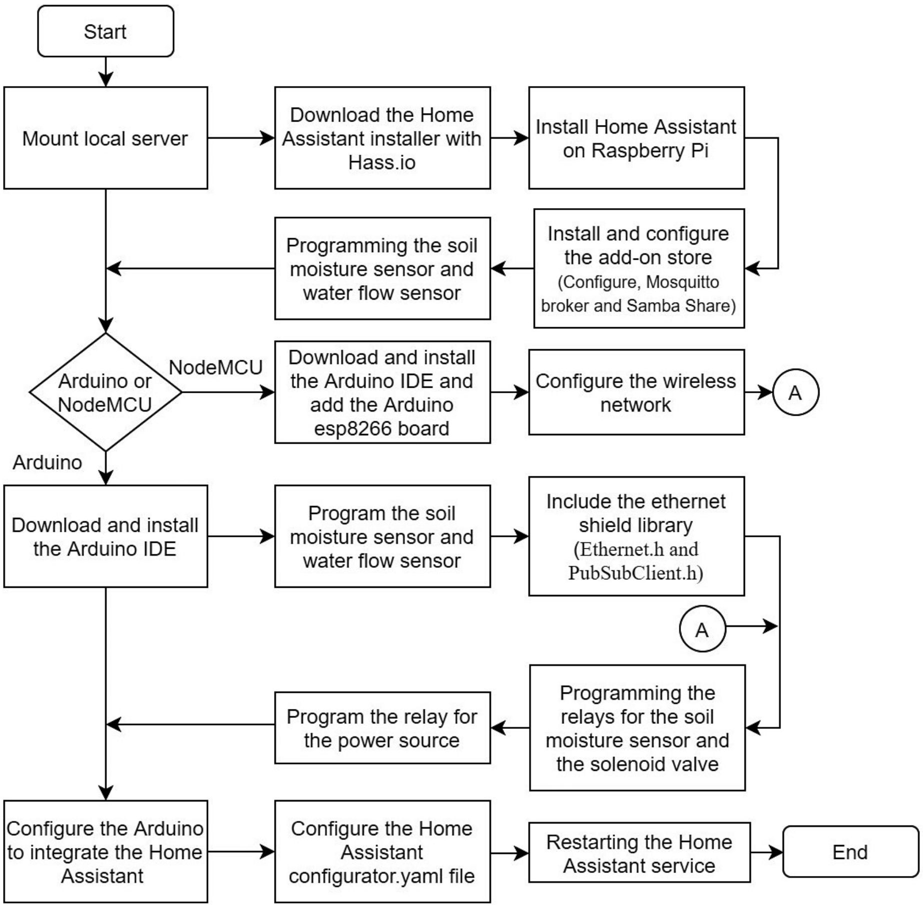
Start
Mount local server
Download the Home Assistant installer with Hass.io
Install Home Assistant on Raspberry Pi
Install and configure the add-on store (Configure, Mosquitto broker and Samba Share)
Programming the soil moisture sensor and water flow sensor
Arduino or NodeMCU
NodeMCU
Download and install the Arduino IDE and add the Arduino esp8266 board
Configure the wireless network
A
Arduino
Download and install the Arduino IDE
Program the soil moisture sensor and water flow sensor
Include the ethernet shield library (Ethernet.h and PubSubClient.h)
A
Programming the relays for the soil moisture sensor and the solenoid valve
Program the relay for the power source
Configure the Arduino to integrate the Home Assistant
Configure the Home Assistant configurator.yaml file
Restarting the Home Assistant service
End

Fig. 10. Home Assistant, Arduino, and NodeMCU process diagram

4 Results and discussion

After analyzing the results of the survey, two average gardens were chosen to validate the proposals for the automation of garden irrigation in homes in the city of Arequipa, one of 8 and the other of 18 square meters. In the first garden, Arduino was used with its necessary components; for the second garden, the NodeMCU board was used. The Adafruit IO system was integrated for three months and then Home Assistant for another three months. Both alternatives had the same conditions required for the investigation.

4.1 Option 1: watered using an automated system incorporating Arduino and Adafruit IO

Using the Arduino board and the Adafruit IO platform, it was possible to automate garden irrigation (see Figure 11, Garden A); however there was a limitation in that several network cabling footages were needed to connect with the home network because this alternative requires an Internet connection.

4.2 Option 2: irrigated using an automated system incorporating NodeMCU and Adafruit IO

The second option uses the NodeMCU board, which is smaller than Arduino (see Figure 11, Garden B). Contrary to the previous alternative, this has the advantage of using a home wireless network without the need to include a network cable to integrate with the IoT platform (Adafruit IO).

These first two options have the same result. Owing to the Adafruit IO platform (see Figure 11), it is left to the discretion of the user to wire the house or use a wireless connection so that the garden can be monitored at any time.

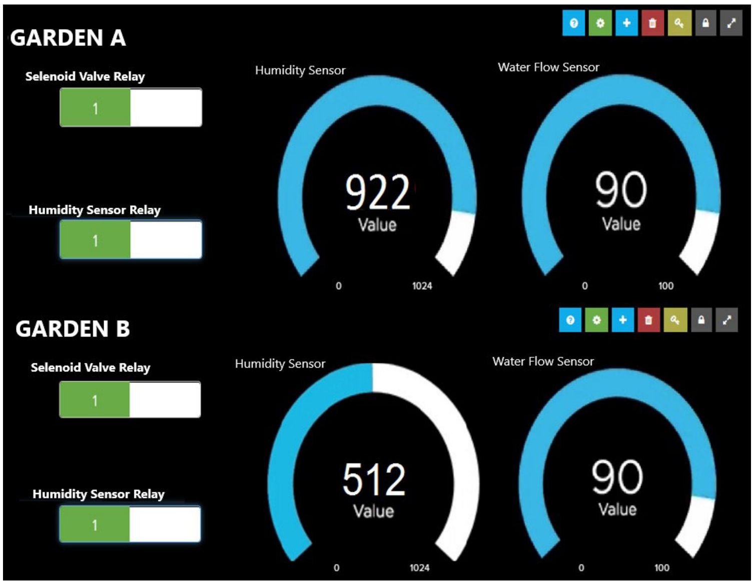

Fig. 11. Adafruit IO with Arduino and NodeMCU

4.3 Option 3: watered using an automated system incorporating Arduino and Home Assistant

Unlike the first two options, the third irrigation automation alternative uses a local server that is installed on a low-power computer (Raspberry Pi). The home automation system installed is Home Assistant, which makes possible the integration of IoT equipment with different manufacturers for the automation of the rest of the home. This option uses free code that makes customization according to the needs of the user possible; however, it has the same disadvantage as the first option of requiring wiring of several footages from the garden to the electronic components (see Figure 12, Garden C).

4.4 Option 4: watered using an automated system incorporating NodeMCU and Home Assistant

The last automation option is similar to the penultimate option because it uses Home Assistant, with the difference that it integrates the NodeMCU board instead of the Arduino one, and it has the advantages of being wireless and having a local server (see Figure 12, Garden D).

These last two options have the same display format (see Figure 12), in addition to providing facilities for home automation. This final option is a good option for beginners who are familiarizing themselves with different IoT equipment, because it makes it possible to scale the system with different IoT manufacturers.

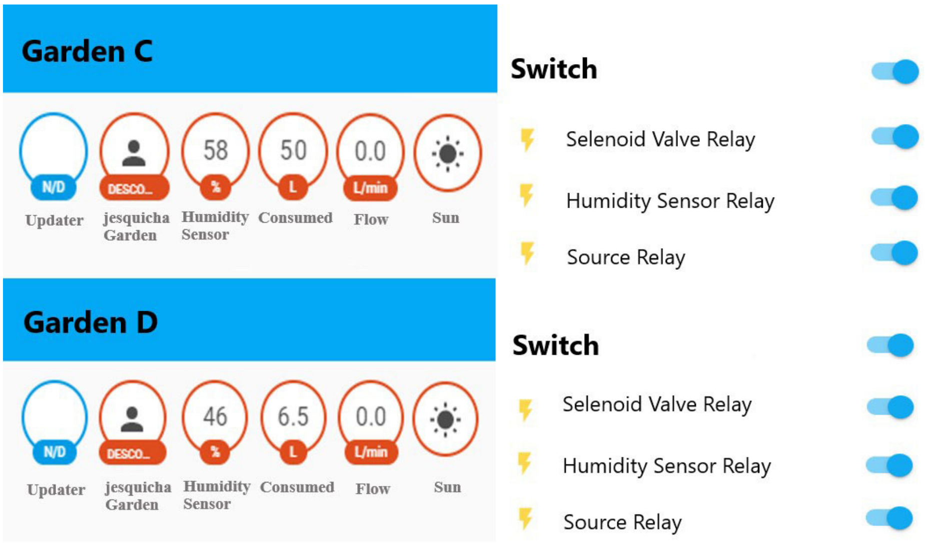

Fig. 12. Home Assistant with Arduino and NodeMCU

4.5 Optimizing the efficient use of water

Table 1 shows the water consumption results for the two gardens. The traditional consumption of gardens using gravity irrigation was analyzed in the first instance. Then, it was automated as Garden A, first with the option using NodeMCU with Adafruit IO. Then, after three months, the automated system was changed to the Home Assistant platform. The same experiment was conducted with Garden B, but using Arduino with Adafruit IO. Then, after three months, the system was changed to Home Assistant, as in the first garden. In addition, Table 1 shows savings of an average of 42%.

This is important because it makes it possible to rationalize consumption, resulting in a decrease in the drinking water bill.

Table 1. Garden watering (traditional and optimized)

Gardens	Normal Consumption (L)	Optimized Consumption (L)	Savings (%)
A	60.5	35	42.1
B	455.2	265.7	41.6

4.6 Optimizing efficient use of energy

The four options presented in this research were tested by keeping them on for 24 h. The electricity consumption, amperage, and power of each were obtained (see Table 2).

- Option 1: Arduino with Adafruit IO required 0.27 A, with a power of 1.35 W, resulting in an electrical consumption of 972 Wh/month.
- Option 2: NodeMCU and Adafruit IO required 0.09 A with a power of 0.45 W, and its electrical consumption was 324 Wh/month.
- Option 3: Arduino integrated with Home Assistant required 0.64 A, with a power of 3.2 W, generating an electrical consumption of 2304 Wh/month.
- Option 4: NodeMCU integrated with Home Assistant required 0.49 A, with a power of 2.45 W, consuming 1764 Wh/month.

Table 2. Comparison of the four options

Options	Description	Amperage (A)	Voltage (V)	Power (W)	Wh/Month
1	Arduino and Adafruit IO	0.27	5	1.35	972
2	NodeMCU and Adafruit IO	0.09	5	0.45	324
3	Arduino and Home Assistant	0.64	5	3.2	2304
4	NodeMCU and Home Assistant	0.49	5	2.45	1764

During the first month of experimentation, it was observed that the four options worked correctly; however, the last two proposals had a high consumption and deterioration of the soil moisture sensors. Thus, it was decided to add a switch on the NodeMCU and Arduino development boards to allow the local server (Raspberry Pi) to run for only 4 h and automatically turn on the next day. Because the Home Assistant system stores the history of everything that happens in the 4 h it is turned on daily, the system has sufficient time to capture the data through the soil moisture sensor and watering during that time if needed.

Table 3 shows that the lowest power consumption is 564 Wh/month with the NodeMCU option with Home Assistant; the power consumption of the Arduino alternative with Home Assistant was reduced to 1194 Wh/month by only having it on for 4 h daily.

Table 3. Optimizing the irrigation of the final two options

Options	Description	Time on (h/day)	Wh/month
3	Arduino and Home Assistant	24	2304
4	NodeMCU and Home Assistant	24	1764
3	Arduino, Home Assistant, and NodeMCU	4	1194
4	NodeMCU and Home Assistant	4	564

4.7 Arduino and NodeMCU hardware features

Table 4 reflects the differences between Arduino and NodeMCU.

Table 4. Hardware comparisons

Hardware	Arduino UNO	NodeMCU
Microcontroller	ATmega326	Tensilica 32-bit RISC CPU Xtensa LX106
Limit input voltage	3–12 V	3.3–5 V
Memory RAM	2 kB	64 kB
Memory flash	32 kB	4 MB
Energy consumption	0.27 A	0.09 A
Scalability	SI	SI
Network connection	100 Mb/s wired (requires Ethernet Shield)	Wireless Wi-Fi 802.11 b/g/n
Temperature (working)	40°C	24°C
Analog Pins	6	1
Input/Output Digital Pins	14	17
Price (USD)	$16	$8
Implementation price (Home Assistant)	$146	$134
Implementation price (Adafruit IO)	$59	$46

- The Arduino Uno board has more analog pins, making it possible to use more sensors or actuators, but this leads to an increase in power consumption.
- The NodeMCU has more RAM (64 kB), which is important because it has more active processes in memory, as well as more flash memory, providing more space to add source code in case it is required to create a more complex system (4 MB).
- To have a wired network connection in Arduino Uno, Ethernet Shield is required. Its transmission speed is 100 Mb/s with a cable, whereas NodeMCU uses a wireless network (802.11 b/g/n).
- The operating temperature of Arduino reaches 40 ºC, requiring a fan (cooler) in the summer, on the other hand, the NodeMCU reaches 24 ºC.
- Price of NodeMCU board ($46–$134, USD) was the least compared to all the other boards, which makes it the preferred equipment for IoT in homes.

4.8 Related IoT projects

Most related research projects used a proprietary system that integrates Arduino as an intermediary [9],[30] or, in its absence, NodeMCU [11],[12],[31]. In contrast, in the four options proposed in this research, Home Assistant or Adafruit IoT have been used as automation platforms [32].

The research proposed in another report [33] is similar to this study because it used W5100, which is the core of Ethernet Shield and ESP8266 (similar to NodeMCU); however, that research focused on having two systems: one connected to the Internet and another that works locally with short intervals of Internet connection. Similarly, in other research [34], the Home Assistant platform was used, which works both locally and remotely, but a linkage to the Nabu Casa website was required at a cost of $5 per month.

In another article [35], a remote-control system was presented that makes use of Raspberry Pi 3, a proprietary application for Android, and electronic devices (sensors and actuators) to control the electronic equipment in a room and analyze the response time. The drawback of this approach is that it is not scalable. However, in the options in this study, both Home Assistant and Adafruit IoT have a dashboard that makes customization and scaling facile.

5 Conclusions

The alternatives proposed in this study are effective for the automation of garden irrigation in homes, but they depend on the factors analyzed in this study to implement an automated irrigation system according to user needs. The four proposed alternatives do not consume much electrical energy, costing $0.5 (USD) per month. The two options that make use of the Home Assistant platform do not need to stay on for 24 h because they have been optimized with the use of a relay to control the on time of the local system so that it is available only for 4 h per day. When Home Assistant is used with Raspberry Pi, there is a higher degree of customization because Raspberry Pi makes it possible to edit the functionalities through the dashboard or directly from the source code, which, together with the help of its global community, means a constant correction of errors and the implementation of different improvements in a reduced time. However, the remaining options with Adafruit IO must be kept on for 24 h/day because the server is in the Cloud. Although there is the advantage of constant visualization from any device through a web browser. Finally, the NodeMCU option with Adafruit IO is the alternative with the lowest power consumption and the most economical of the four alternatives in this research.

6 Acknowledgment

We would like to thank la Universidad Nacional de San Agustín de Arequipa in Peru for financing the project with contract number TD-002-2018-UNSA, making it possible to conduct this research that benefits families with home gardens.

7 References

[1] Aquae Foundation, "Key facts about water in the world", https://www.fundacionaquae.org/wiki-aquae/principales-datos-del-agua-en-el-mundo/. [Accessed Mar. 10, 2021].

[2] United Nations, "Water", https://www.un.org/es/sections/issues-depth/water/index.html. [Accessed Mar. 10, 2021].

[3] F. Darío, "Rainwater system proposal as an alternative to save drinking water in the educational institution María Auxiliadora from Caldas, Antioquia", http://bibliotecadigital.udea.edu.co/bitstream/10495/1325/1/PropuestaSistemaAprovechamientoAguaLluvia.pdf

[4] J. Quintana, "Propuesta de un Sistema de Aprovechamiento de Agua de Lluvia para agricultura urbana en la terraza de los laboratorios de la ciudadela el Provenir de la Universidad Distrital Francisco José de Caldas", Universidad Distrital Francisco José de Caldas, 2017. https://repository.udistrital.edu.co/handle/11349/6216

[5] M. Marroquín and J. Trejo, "Diseño de Hormigón permeable para el aprovechamiento de agua lluvia en superficies de uso peatonal", Universidad Central del Ecuador, 2019. http://www.dspace.uce.edu.ec/handle/25000/20271

[6] V. Vimos and E. J. Sacoto Cabrera, "Results of the implementation of a sensor network based on Arduino devices and multiplatform applications using the standard OPC UA", IEEE Latin America Transactions, vol. 16, no. 9, pp. 2496–2502, 2018, https://doi.org/10.1109/TLA.2018.8789574

[7] S. Quincozes, T. Emilio, and J. Kazienko, "MQTT protocol: Fundamentals, tools and future directions", IEEE Latin America Transactions, vol. 17, no. 9, pp. 1439–1448, 2019, https://doi.org/10.1109/TLA.2019.8931137

[8] A. Andrada Tivani, R. Murdocca, C. Sosa Paez, and J. Dondo Gazzano, "Didactic prototype for teaching the MQTT protocol based on free hardware boards and node-RED", IEEE Latin America Transactions, vol. 18, no. 2, pp. 376–382, 2020, https://doi.org/10.1109/TLA.2020.9085293

[9] M. Ali, Z. Nazim, W. Azeem, M. Harrom, A. Hussain, K. Javed, and M. Tariq, "An IoT based approach for efficient home automation with Thing Speak", International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 6, 2020. https://thesai.org/Downloads/Volume11No6/Paper_15-An_IoT_based_Approach_for_Efficient_Home_Automation.pdf. https://doi.org/10.14569/IJACSA.2020.0110615

[10] M. B. I. Astutiningtyas and M. M. Nugraheni Suyoto, "Automatic plants watering system for small garden", International Journal of Interactive Mobile Technologies, vol. 15, no. 2, p. 200, 2021, https://doi.org/10.3991/ijim.v15i02.12803

[11] S. Rao, A. Smitha, and K. Kulkarni, "Smart Phone based cost effective visitor management system for smart offices", International Journal of Interactive Mobile Technologies, vol. 12, no. 6, pp. 112–123, 2018, https://doi.org/10.3991/ijim.v12i6.9476

[12] R. K. Jain, B. Gupta, M. Ansari, and P. P. Ray, "IOT Enabled Smart Drip Irrigation System Using Web/Android Applications", 2020 11th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2020, 2020, https://10.1109/ICCCNT49239.2020.9225345

[13] A. Palza Monroy and M. Quequesana Miranda, "Declaration of the countryside as a Cultural heritage of the city of Arequipa", Universidad Nacional de San Agustín de Arequipa. http://repositorio.unsa.edu.pe/handle/UNSA/10674

[14] La Republica Newspaper, "Arequipa: Alert about possible lack of water", Available https://larepublica.pe/sociedad/1303268-arequipa-alertan-posible-falta-agua/. [Accessed Feb. 21, 2020].

[15] P. López-Roldán and S.Fachelli, "Quantitative social research methodology", Universitat Autònoma de Barcelona, 2015. https://pagines.uab.cat/plopez/content/III.2

[16] Official Website of the National Institute of Statistics and Informatics, “Arequipa alberga 1 millón 316 mil habitantes”, http://m.inei.gob.pe/prensa/noticias/arequipa-alberga-a-1-millon-316-mil-habitantes-9903/. [Accessed Nov. 5, 2020].

[17] Adafruit official page, “What is Adafruit IO”, https://learn.adafruit.com/welcome-to-adafruit-io/what-is-adafruit-io. [Accessed Aug. 5, 2020].

[18] D. Norris, Home Automation with Raspberry Pi_ Projects Using Google Home, Amazon Echo, and Other Intelligent Personal Assistants, 1st ed, 2019.

[19] R. Singh, A. Gehlot, L. R. Gupta, B. Singh, and M. Swain, Internet of Things with Raspberry Pi and Arduino, 2019. https://doi.org/10.1201/9780429284564

[20] G. Tojeiro, “Taller de Arduino un enfoque práctico para principiantes”, Editorial Marcombo, México 2015.

[21] Panamahitek, “Control tools for arduino: the relay”, http://panamahitek.com/herramientas-de-control-para-arduino-el-relay-o-relevador/. [Accessed Dec. 2, 2020].

[22] Naylampmechatronics, “Water flow sensor tutorial”, https://naylampmechatronics.com/blog/47_tutorial-sensor-de-flujo-de-agua.html. [Accessed Dec. 22, 2020].

[23] Naylampmechatronics, “Selenoid valve ½” 12V DC”, https://naylampmechatronics.com/valvulas/314-valvula-solenoide-12-12v.html. [Accessed Dec. 22, 2020].

[24] Official page of Home Assistant, “Installing home assistant”, https://www.home-assistant.io/hassio/installation/. [Accessed Dec. 26, 2020].

[25] Official page of GitHub, “Hassio-addons”, https://github.com/home-assistant/hassio-addons/tree/master/configurator. [Accessed Feb. 6, 2021].

[26] Mosquitto, “Eclipse Mosquitto”, https://mosquitto.org/. [Accessed Feb. 6, 2021].

[27] GitHub, “Home Assistant Add-on: Samba”, https://github.com/home-assistant/hassio-addons/tree/master/samba. [Accessed Feb. 6, 2021].

[28] Arduino course page, “Tutorial 31: Soil moisture sensor”, http://cursoarduino.proserquisa.com/2016/10/19/tutorial-31-sensor-de-humedad-del-suelo/. [Accessed Feb. 16, 2021].

[29] R.Vega, “Configuring Arduino and MQTT”, https://ricveal.com/blog/arduino-mqtt/. [Accessed Feb. 6, 2021].

[30] S. Sayanthan, T. Thiruvaran, and N. Kannan, “Arduino based soil moisture analyzer as an effective way for irrigation scheduling”, 2018 IEEE 9th Int. Conf. Inf. Autom. Sustain. ICIAfS 2018, pp. 1–4, 2018, https://doi.org/10.1109/ICIAFS.2018.8913355

[31] Z. Wan, Y. Song, and Z. Cao, “Environment dynamic monitoring and remote control of greenhouse with ESP8266 NodeMCU”, Proc. 2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2019, no. Itnec, pp. 377–382, 2019, https://doi.org/10.1109/ITNEC.2019.8729519

[32] P. Mathivanan, G. Anbarasan, A. Sakthivel, and G. Sevam, “Home Automation using Smart Mirror”, Proc. of International Conference o System Computation Automation and Networking, 2019. https://doi.org/10.1109/ICSCAN.2019.8878799

[33] M. V. Moise, L. M. Ionescu, and P. M. Svasta, “Comparison between deploying an IoT system designed to be used online and one designed to be used offline”, in 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging, SIITME 2018 - Proceedings, 2019, pp. 161–164, https://doi.org/10.1109/SIITME.2018.8599279

[34] J. Esquicha-Tejada and J. Copa-Pineda, “Integration of an IoT system–Photovoltaic system to optimize the consumption of drinking water in the irrigation of gardens in the City of Arequipa”, Proc. LACCEI Int. Multi-conference Eng. Educ. Technol., no. July 2020, pp. 27–31, 2020. https://doi.org/10.18687/LACCEI2020.1.1.212

[35] I. S. Areni, A. Waridi, Indrabayu, C. Yohannes, A. Lawi, and A. Bustamin, “IoT-Based of Automatic Electrical Appliance for Smart Home”, International Journal of Interactive Mobile Technologies, vol. 14, no. 18, pp. 204–211, 2020. https://doi.org/10.399l/ijim.v14i18.15649

8 Authors

José David Esquicha Tejada is a Systems Engineer from the Universidad Catolica de Santa María, UCSM (2008). He holds a Second Specialty Professional Degree in Systems Auditing and Information Security at UCSM (2019). Master in Strategic Telecommunications Management at the Miguel de Cervantes European University (2013). He is a candidate for a Doctor of Environmental Sciences and Renewable Energies at the Universidad Nacional de San Agustín de Arequipa. He is currently an Assistant Professor at UCSM in the Faculty of Physical and Formal Sciences and Engineering. His research interests include the Internet of Things (IoT) and educational technology.

Juan Carlos Copa Pineda is an electronic engineer (1996), Master in engineering (2005). He received the degree of Dr. in Energy Engineering, at the Universidad Nacional de San Agustín de Arequipa, Peru (2014). He is currently a professor at the Universidad Catolica de Santa Maria and Principal Professor at the National University. His research fields include electronic control of electrical machines, power electronic static converters, automation of energy and power systems for the generation, transmission, distribution, and end-user segments.

Article submitted 2021-03-15. Resubmitted 2021-08-28. Final acceptance 2021-08-28. Final version published as submitted by the authors.