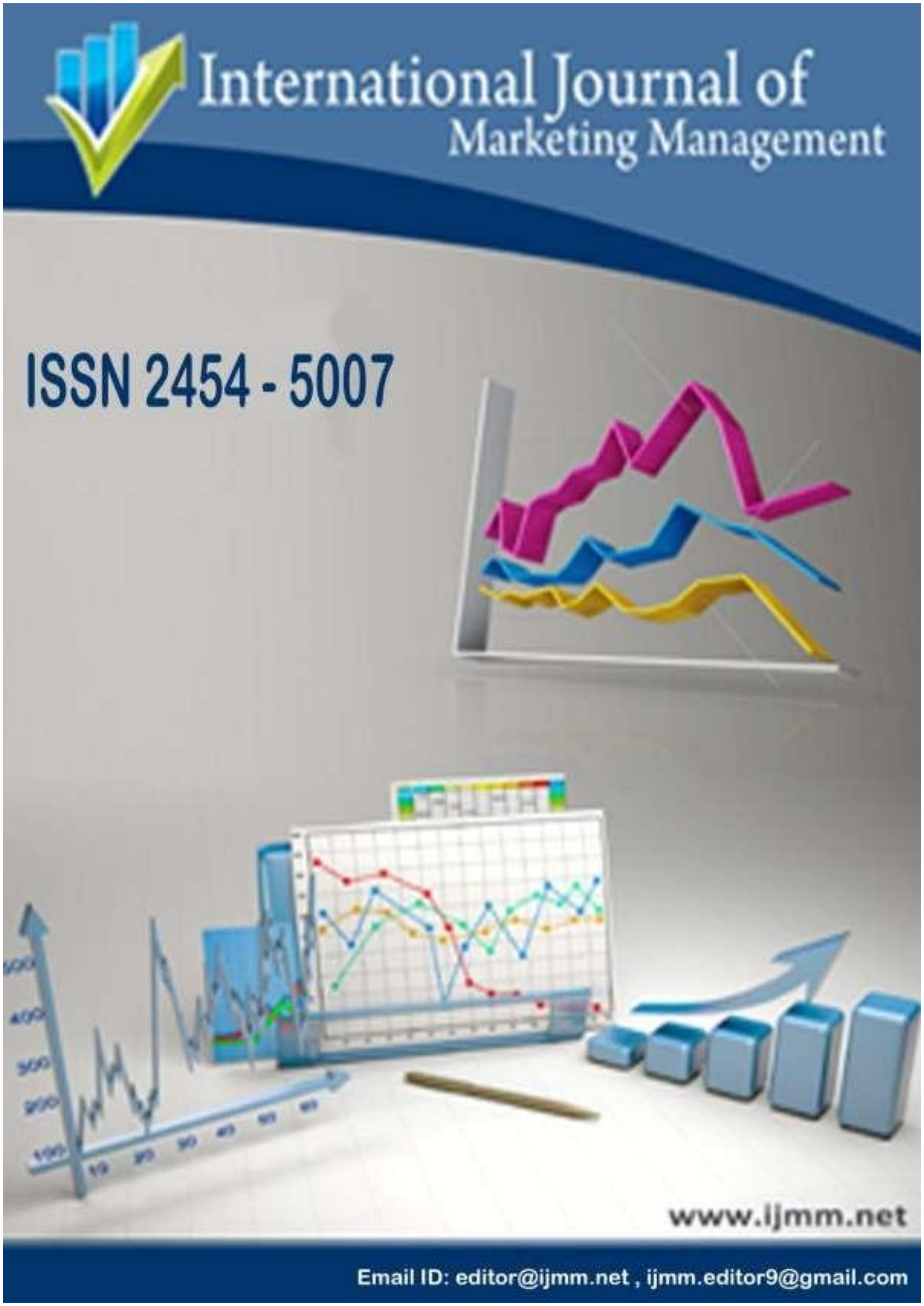
International Journal of
Marketing Management
ISSN 2454 - 5007
www.ijmm.net
Email ID: editor@ijmm.net , ijmm.editor9@gmail.com

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

IoT-Based Voice-Controlled Smart Home Automation System Using Arduino and Firebase

[1]Akshaya,[2]Akula Akshitha,[3]Geriganti Arun,[4]Jangam Shiva Sai,[5]K.Karthik

[1,2,3,4] Student,Department of ECE,Narsimha Reddy Engineering College,Misammaguda(V),Kompally-500100,Telangana State,India.
[5] Assistant Professor,Department of ECE,Narsimha Reddy Engineering College,Misammaguda(V),Kompally-500100,Telangana State,India.

Abstract

Home automation systems, in which a building can intelligently react to environmental changes and the needs of its occupants, have recently emerged with the rise of the Internet of things and smart cities. Some kind of control, ideally wireless, that facilitates communication between the structure and its occupants is required for the establishment of various parameters inside such an ecosystem. Voice controls are only one of several forms of controls. This article details the process of creating a voice assistant that can manage various electrical devices present in a home automation system. Various software and hardware tools have been used in the construction of this system, including but not limited to: Visual Studio Code, Google Firebase, MAMP, jQuery, PHP, and Arduino UNO. A web platform has been established that allows for the control of a number of sensors, including a gas detector, a presence detector, a temperature and humidity sensor (DHT11), and a relay for lighting control. The process of creating the manual connection and how it has progressed to voice control have been detailed.

Introduction

The proliferation of new technical communication schemes like Smart cities, the Internet of Things (IoT), and industry 4.0 has spurred the creation of algorithms and devices that enable communication between devices and between gadgets and their users. This paves the way for the creation of smart buildings. As a new paradigm for home interaction, home automation aims to make people's lives easier and safer by implementing various control schemes [A] and intelligent systems that communicate with their surroundings [B, B1, B2]. The comfort of an individual within such environments depends on different factors, such as lighting [B3], temperature [B4], the feeling of security [B5], among others. Various home automation systems, outlined for implementation in the scientific literature [D1, D2, D3], make use of wireless technologies for internal and external communication [E]. Just like any other kind of user interface, home automation systems provide a number of methods for users to interact with them. One of them is via the use of voice commands, which allow users to simply say what they want the system to do. Some of the existing elements that respond to voice commands are the so-called voice assistants, among which we can mention Alexa from Amazon, Siri from Apple, Cortana from Microsoft, among others [G]. The way each of these helpers functions is unique [H].

One study found that even a 5-year-old can use a voice assistant, so it's safe to say that almost everyone is using one these days (Akash et al., [H1]). There are several varieties of voice assistants available because of the significant role they may play in home automation control schemes. To illustrate our point, Prasad et al. [I] developed a voice-activated system that regulates lighting, ventilation, and an electrical outlet. In [J] a voice control system was designed that can be controlled remotely by means of a Tablet or a Smartphone with the intention of assisting disabled people. In the same way, it is possible to see how this type of control system can also be used with the intention of entering the home automation building within the IoT [K], which today is very useful. For the realization of the mentioned schemes, there are

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

many and varied types of tools for the design and implementation of the proposals, however, for the purposes of this paper, it is important to mention the use of the NodeMCU within the design of many of these controllers. . The NodeMCU is an open source platform designed for the development of IoT applications, for that reason, it has been used in different voice control systems applied to home automation design. To illustrate: A voice-based system for automating electrical devices was created by Florence et al. [L] utilizing the Dialogflow, NodeMCU, and Firebase frameworks. A voice-based control system that integrates Google Assistant was produced by Shakthidhar et al. [M] using the NodeMCU and Arduino UNO. Meanwhile, Kodali et al. [N] built a smartphone-based control and monitoring system with the goal of improving people's comfort and quality of life. The system makes use of Mongoose OS, Android app, Google Assistant, NodeMCU, MQTT, Node-RED, and IFTTT. With the intention of supporting patients, Roy et al. [O] presented their voice-operated low-cost IoT home automation proposal, using Google Assistant, Blynk, and NodeMCU ESP32; On the other hand, to reduce the use of electricity, a voice control was developed that uses the NodeMCU to operate the devices that are connected to a set of relays on a power supply [P] and in this same line there is a system power efficiency control used by the Raspberry Pi card and the NodeMCU for its operation [Q]; among other developments of this type [R]. In keeping with the theme, this paper describes the design and development of a voice control system that can be used in IoT and home automation settings. The system is built using a variety of tools, including Visual Studio Code, Google Firebase, MAMP, jQuery, PHP, and temperature and humidity sensors (DHT11). It also includes a gas sensor and a relay for turning lights on and off. The proposed system is programmed in a Host with a private client-server communication.

Proposal development

Here, the paper's idea and its subsequent hardware and software development are detailed. It can't be done without these things: Boards for hardware control (Arduino Uno), wireless communication (NodeMCU V2), The following items are needed: a humidity and temperature sensor (DHT11), a 5V/ KY-019 DC relay, a motion sensor (PIR), a buzzer, the sockets for the relays, the ARDUINO IDE for database connection and microcontroller programming (so we can control the systems from the NODEMCU V2), Visual Studio Code for interface programming and database connection, Google Firebase, and MAMP for PHP connection to the interface server. The proposal for the system's implementation included the light control; In addition to a proximity alarm, other helpful sensors include a temperature and humidity monitor, which can be used to keep tabs on the state of a room, crop, electrical system, etc., and a gas detector, which can detect levels of harmful gases and is important for home and business security. A photo-infrared (PIR) movement sensor and an active buzzer comprise the proximity alarm system's auditory signal. The temperature and humidity system made use of a DHT11 sensor to keep track of historical data on humidity and temperature, which might be useful for controlling other systems or just for safety reasons. At last, we have the gas detection system mocked up with an alarm that can detect very high gas levels in a kitchen. To do this, we employed a MQ2 gas sensor.

Voice control system

The suggested voice system is built around using local servers to host the web interface. In this case, local and free Apache servers are utilized, which are provided by the MAMP software. The developer, using calls in an index file, combines the logical functions of an object-oriented language like JavaScript with the aesthetic and graphic concerns of HTML and CSS. This hybrid serves as the interface for the voice assistant and allows us to interact with connected objects in the real world through the board, firehost, and its linkage with this interface, either manually or remotely. You may get a good idea of the developed proposal's structure and its components by looking at figure A.

International Journal of Marketing Management

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

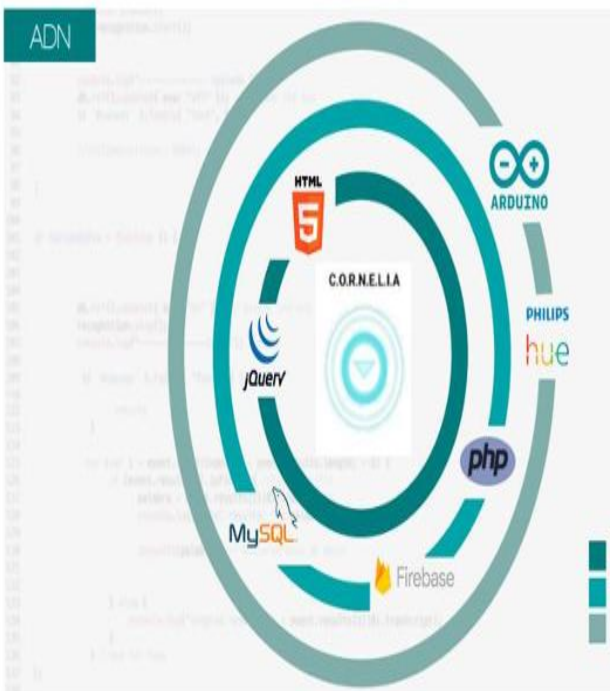

Fig. A.- General structure of the developed voice assistant.

By constructing trees, which enable a parent node to be connected to other systems or leaf nodes, and which in turn contain different conceptual (semantic) attributes around the system they represent, the Firebase application may establish a connection to the system database. The process of creating the database and connecting it to the code in the Arduino IDE, which originates from the NodeMCU board and, therefore, the hardware control system, is detailed below. The following is an example of a database that is built from the ground up and linked to a program using the aforementioned libraries and the syntax for invoking them: • Firebase, where a Google account is associated with the creation of a project bearing the name of the Parent Node—CASA1 in this example. Database creation is complete. Beginning with the parent node, the tree is constructed by adding its son nodes. The son nodes link the database to the sheets that represent the various hardware systems that will be used. • Data connectivity via the Wi-Fi network is implemented in a real-time system.

Since this Firebase feature enables us to utilize and, therefore, remotely operate any device linked to the network, it forms the foundation of the project. Here we'll build the database with the assumption that our NodeMCU board will be linked to a relay switch, which in turn will be attached to a light bulb that can be manually switched on and off using the objects' structure: Branching out from the son node "Room 1"—in this example, FOCO 1 AND FOCO—represent the NodeMCU pins used in the code. 2. This method, a database is created that will be manually controlled, similar to a NodeMCU Serial Monitor, or two lights in a room that have HIGH and LOW values. • The connection is established by attaching the Wi-Fi credentials that the NodeMCU board will use to communicate with the database's firehost.

These credentials include both an authentication and a password. A project's credentials specification is associated with the database URL. This completes the definition of the database host. • The current minimal set of considerations consists of the names of the son nodes and their corresponding branches; they will serve as the connections between the hardware and the cloud, where it may be controlled via a web interface. • As previously said, we can get the voice assistant to identify the systems via commands and invocations and put them at our beck and call by using the ID or route recorded in Firebase of each device. • In order to link the NodeMCU (ESP8266), the necessary internet credentials, including the Wi-Fi network's name and password, must be set up. Note that this system's setup was developed to allow the NodeMCU to access the Firebase database as a Serial Monitor. An actuator was designated as the output for this setup, along with a 115200 bauds serial transmission. We program the hardware using the routes specified for its identification in Firebase after it is set to start connecting to both Firebase and operate as a Serial Monitor. Then, we continue to connect it to the Wi-Fi network using the ESP8266. Figure B displays the status of each output actuator in addition to the specified paths.

Sensor and actuator system

Assembling the circuit shown in figure C allowed us to operate the various actuators using the remote Serial Monitor in Firebase. Each actuator is linked to a separate pin on the ESP8266 board. In figure D, we can notice the changes achieved in the humidity and temperature sensor, which indicate that the program starts to check the transmission data is appropriate after it has been constructed and performed. We did something different to operate and monitor the PIR

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

sensor. Here, we created the alarm variable to directly receive the boolean string from the "casa1/alarma" route. After conditioning the variable for the "activating the alarm" event, give it some time to read the sensor, which will provide two potential responses. The first is a "Movimiento no detectado" (Movement NOT detected) message that is sent when the sensor registers a logic 0 and a "Movimiento detectado" (Movement detected) message that is sent when the sensor registers a logic 1. Also, the sensor won't start operating until the alarm is manually activated on Firebase, even if it has a buzzer tone already. Figure E clearly displays this modification.

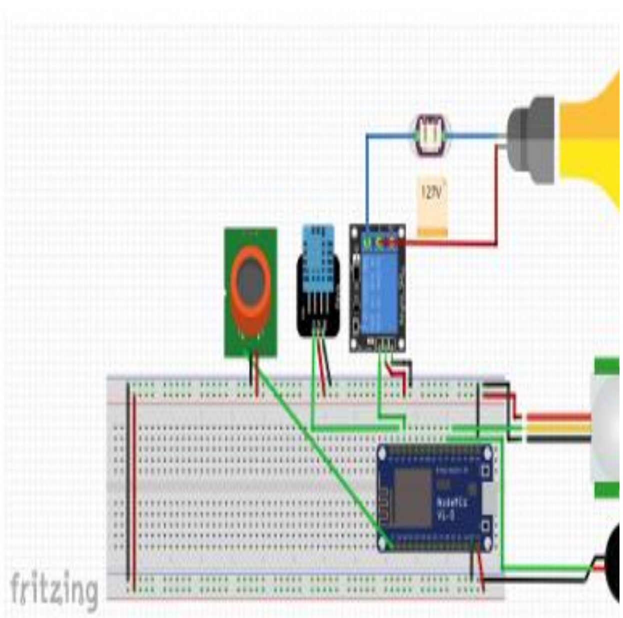

Fig. C.- Test system connection circuit, already using the NodeMCU.

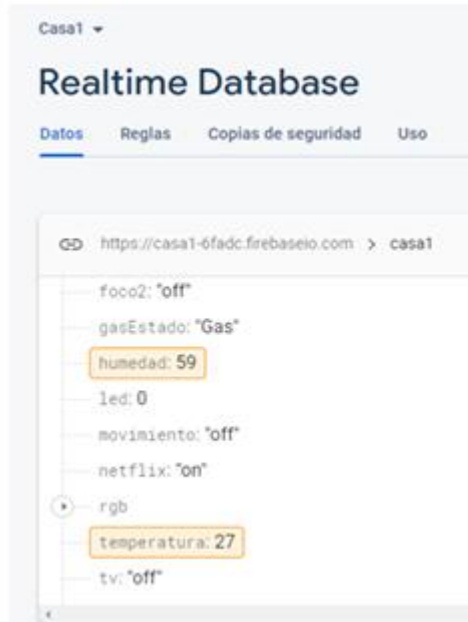

Fig. D.- DHT11 sensor data, obtained through communication with the host designed for remote control.

Fig. E.- Control of the “alarma” variable.

Although the alert and movement routes are not stored together in the database, they are connected in the programming because Firebase organizes the nodes that we create alphabetically. Thanks to the setString function of the Firebase class, we were able

International Journal of Marketing Management

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

to route the hardware ID to the condition of the measured ranges, which was necessary for defining certain detection ranges that correspond to the alerts that will be issued in the database. This allowed the logic of the gas sensor program to simply display certain messages depending on the measured value. There is now a waiting period associated with every message display. Changing the gas detection by feeding the sensor gas via a "lighter" is shown in figure F, and the host's message is shown in figure G.

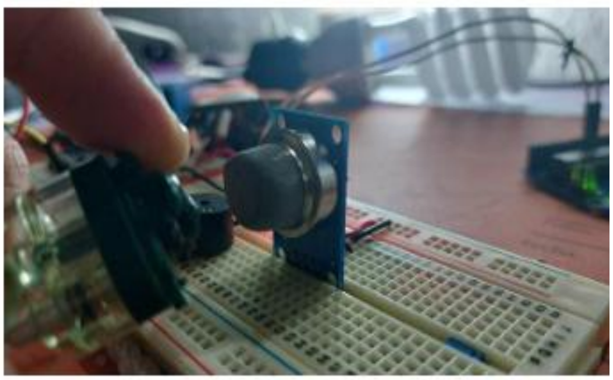

Fig. F.- The gas sensor is stimulated by means of a igniter to verify that the host is receiving the information and displaying it correctly..

Fig. G.- The Host showing the change of the state of the gas, depending on the information received from the gas sensor: “gas detectado” (gas detected) and “Peligro, se detectan altas cantidades de gas” (Danger, high amounts of gas are detected).

The Artyom and Firebase interface Up until now, the system's design has only focused on remote operation via a host, devoid of voice instructions. This was done to ensure sufficient connection between the host and the sensor system's hardware. The time has come to integrate a voice assistant into your system. Web design tools like Artyom, HTML, CSS, and JavaScript and its extensions were utilized to construct the interface and voice control system. This allows us to interact with our interface using voice commands and, consequently, generate responses that enable interaction with physical objects. A web interface was developed using HTML, CSS, and JavaScript. It will be hosted by free local web servers for practical and economical reasons. From this interface, we will physically and audibly operate the integrated hardware pieces. The graphs and tables, as well as the aesthetic design, were created using HTML and CSS. CSS is a language that allows us to define the aesthetic aspect of the interface and has many other uses, such as configuring its format, color, size, and more. A table was created for the graphic presentation. The first column is called "modulo" and it represents the hardware that needs to be controlled. For example, "Foco sala" would be the room focus, "Foco Recamara" the bedroom focus, "Focos WC" the bathroom focus, etc. The second column details the location where the hardware is mounted. The third column has dynamic buttons that allow users to view and interact with the status of each system. Figure H shows the representation.

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

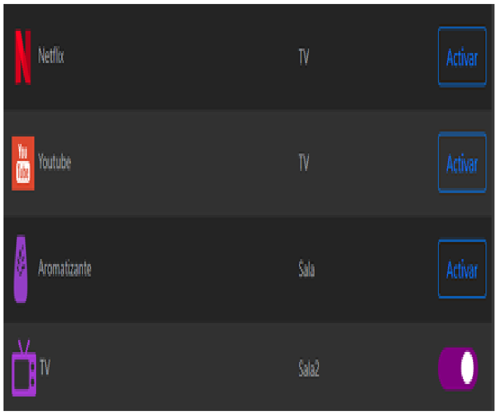

Fig. I.- Commercial pages to which the designed system redirects.

This interface is extensible, so we may add features as needed by incorporating additional apps. The flavoring and TV symbols were inserted for demonstrative reasons only, as shown in picture I. Keeping with the original plan, this interface will monitor and adjust the property's environmental conditions. To do this, a dynamic meter was created to display the temperature and humidity values from the DHT sensor, as previously mentioned. Figure J illustrates this point.

With the visual and aesthetic aspects of the interface designed taken care of, the next step is to connect all the buttons and modules to their corresponding database labels in real time. This will allow for the control of the interface through preset voice commands. To implement the functional aspects of the interface, we will use Artyom.js and Firebase scripts within the JavaScript files. Figure L shows some of the typical answers and orders to operate electronic equipment, while Figure K shows the icon that awaits the user's request.

RESULTS

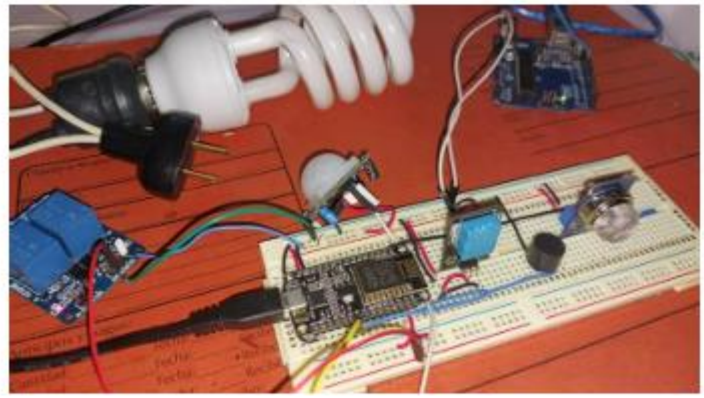

Fig. N.- Final physical representation of the final circuit controlled through the NodeMCU, already using voice commands.

Fig. O.- Web interface, full version.

Conclusion

This article detailed the steps used to create a voice-controlled assistant for Internet of Things (IoT) home automation devices. The NodeMCU ESP 8266 and the Artyom.js library were both required to do this. A web application hosted on a server is the brains of the operation, allowing it to talk to the monitored and controlled sensors and actuators via the Internet. It was feasible to monitor the sensor data and use it to

ISSN 2454-5007, www.ijmm.net
Vol. 14 Issue. 2, April 2026

create control reactions. One important consideration when sending any kind of data over the Internet is the level of security that must be maintained during transmission. A system that employs a local host makes this possible. It was feasible to test the communication and the behavior of the physical equipment in response to Internet requests by creating a physical model. An acceptable level of connectivity between all the components of the system was achieved by the use of a variety of software and hardware tools that were blended throughout the design process. From first manual control of physical elements over the Internet to final voice control of devices, the whole development process is detailed in the paper. We can add more aspects to monitor and control based on the user's demands since the system is expandable. There is room for more sensors; this article only makes use of four (counting the damper as one).

References

[1]. Balasubramanian, Karuppanan, and Akin Cellatoglu. "Analysis of remote control techniques employed in home automation and security systems." IEEE Transactions on Consumer Electronics 55.3 (2009): 1401-1407.

[2]. Sun, Qingquan, et al. "A multi-agent-based intelligent sensor and actuator network design for smart house and home automation." Journal of Sensor and Actuator Networks 2.3 (2013): 557-588. [B1] Wang, Jin, et al. "Bio-inspired ant colony optimization based clustering algorithm with mobile sinks for applications in consumer home automation networks." IEEE Transactions on Consumer Electronics 61.4 (2015): 438-444.

[3]. Baby, Cyril Joe, Faizan Ayyub Khan, and J. N. Swathi. "Home automation using IoT and a chatbot using natural language processing." 2017 Innovations in Power and Advanced Computing Technologies (i-PACT). IEEE, 2017.

[4]. Teymourzadeh, Rozita, et al. "Smart GSM based home automation system." 2013 IEEE conference on systems, process & control (ICSPC). IEEE, 2013.

[5]. Ayan, Onur, and Belgin Turkay. "Smart thermostats for home automation systems and energy savings from smart thermostats." 2018 6th International Conference on Control Engineering & Information Technology (CEIT). IEEE, 2018.

[6]. Jose, Arun Cyril, and Reza Malekian. "Smart home automation security: a literature review." SmartCR 5.4 (2015): 269-285.

[7]. Stolojescu-Crisan, Cristina, Calin Crisan, and Bogdan-Petru Butunoi. "An IoT-based smart home automation system." Sensors 21.11 (2021): 3784.

[8]. Sriskanthan, Nadarajah, F. Tan, and Advait Karande. "Bluetooth based home automation system." Microprocessors and microsystems 26.6 (2002): 281-289.

[9]. Al-Ali, Abdul-Rahman, and Mohammad Al-Rousan. "Java-based home automation system." IEEE Transactions on Consumer Electronics 50.2 (2004): 498-504.

[10]. Asadullah, Muhammad, and Ahsan Raza. "An overview of home automation systems." 2016 2nd international conference on robotics and artificial intelligence (ICRAI). IEEE, 2016.

[11]. Mittal, Yash, et al. "A voice-controlled multi-functional smart home automation system." 2015 Annual IEEE India Conference (INDICON). IEEE, 2015.

[12]. Hoy, Matthew B. "Alexa, Siri, Cortana, and more: an introduction to voice assistants." Medical reference services quarterly 37.1 (2018): 81-88.

[13]. López, Gustavo, Luis Quesada, and Luis A. Guerrero. "Alexa vs. Siri vs. Cortana vs. Google Assistant: a comparison of speech-based natural user interfaces." Advances in Human Factors and Systems Interaction: Proceedings of the AHFE 2017 International Conference on Human Factors and Systems Interaction, July 17− 21, 2017, The Westin Bonaventure Hotel, Los Angeles, California, USA 8. Springer International Publishing, 2018.

[14]. Akash, S., Neeraj Jayaram, and A. Jesudoss. "Desktop based Smart Voice Assistant using Python Language Integrated with Arduino." 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2022.

[15]. Prasad, Dusi Durga, et al. "Voice controlled home automation." 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE, 2021.

[16]. Zainuddin, Aznilinda, et al. "Appliance Control with IOT-Arduino of Voice Command Detection for Mobility Impaired People." International Journal of Interactive Mobile Technologies 15.23 (2021).