Engineering and Technology Journal e-ISSN: 2456-3358
Volume 07 Issue 10 November-2022, Page No.-1719-1733
DOI: 10.47191/etj/v7i11.08, I.F. – 6.484

The Development and Implementation of a Prototype Internet-Based Wireless Home Automation System

Chukwuagu M.I[1], Aneke E.C.[2]
[1,2]Nnamidi Azikiwe University (UNIZIK) Awka, And Caritas University Amoriji Nike, Emene, Enugu, Nigeria.

ABSTRACT: The aim of this paper is to develop and implement a prototype home automation system that uses the Internet-of-Things technology. It is focused on the need to deal with the existing challenges of high cost of ownership, inflexible architecture, and lack of security of those smart systems that are commercially available nowadays. The implement system consists of a microcontroller which uses a web server application to connect some home appliances to the Internet through a relay module and a mobile application. The microcontroller is for receiving and processing the commands from homeowners to monitor or control the required appliances at home and for transmitting the necessary signals to the relay module for the automatic operation of the appliances. The microcontroller maintains the database of all the appliances as well as all the authorized users of the home automation system. The web server app is also used for communication between the Internet and the mobile phone. The mobile application was developed using the Android platform. The Android app acts as the software dashboard that presents the users with a list of devices to interact with in the smart home. The materials used in this paper are inexpensive and readily available. The result of this work was tested in real life to prove its effectiveness and usefulness. And the findings have not only validated the relevance of this work but also highlighted its significance in the digital world of today.

KEYWORDS: Internet cloud, home automation system, Wireless communication, The web service, Camera, Prototype system, Wi-Fi Technology.

I INTRODUCTION

Many of the existing or commercially available home automation systems are mainly conceived, developed, and implemented based on hardwired electrical appliances and installation practices. Employing a traditional wired automation system does not pose much challenge as long as the system is previously planned and installed during the physical building of the house. If, however, an already existing building is to be equipped with an automation system, too much re-engineering or rewiring effort will be needed in order to make the required modification possible. And this will also be at a huge cost to the project owner since the procurement of additional cables and wiring accessories is mandatory. Fortunately, wireless systems [1] can come handy in a problematic situation like this. In recent years, wireless technologies are increasingly being deployed worldwide. Wireless systems that are presently being used every day and everywhere include wireless home networks, GSM phones, Bluetooth-enabled garage door openers, and the "Internet of Things" systems. As of today, however, not much comparative research project on the design and implementation of Internet-based home automation system has been done.

To tackle the problem of home automation systems are currently facing three major challenges. The challenges include the following:

1. **High cost of implementation:** Unlike hardwired systems, a wireless solution does not involve the installation of expensive cables and its associated labor charges.
2. **Inflexible architecture:** A home automation system should have an inherent flexibility so as to enable the extension or reduction of its coverage to accommodate any necessary change in size or budget.
3. **Lack of security:** Any damage to installed cables can compromise the security of a non-wireless system. But a wireless solution can improve system security by using secure Wi-Fi connection to integrate all the sensitive parts of the system.

The aim of the research paper is to develop and implement a prototype Home Automation System that is capable of remotely monitoring and efficiently controlling electrical and electronic appliances in a typical modern home include:

1) To reduce the implementation cost of home automation works by eliminating the need for any expensive cabling of new house or the rewiring of an existing building.

2) To ease the process of installing a home automation system or extending its coverage area by simply

deploying wireless network nodes in places where cabling may be too difficult to carry out or even not feasible, cost-effective, or desirable;

3) To enhance the security of homes by the application of a secure Wi-Fi connection to integrate all the sensitive parts of the system.

In this Internet of Things (IoT) paper, a home automation system is developed and implement with a 6-channel relay using an ESP32 Microcontroller and Google Chrome Web Browser. For this reason, only six (6) home appliances can be controlled either with manual switches or with the Google Assistant App. If any Wi-Fi Hotspot is made available, the ESP32 will be automatically connected to the Wi-Fi and the real-time feedback of the relays can also be controlled and monitored in the Google Home from anywhere in the world.

With this Internet-based Home Automation System, it is possible, through a Web Server (Sinric Pro), to control, four (4) lighting-bulbs, a security surveillance camera, and any other home appliance that is plugged into the available wall socket. This can be achieved by using a smart phone or any other mobile device like a laptop. The AC-mains appliances are connected to the
6-channel relay which is controlled by the ESP32 Microcontroller through the installed Web

Server. Voice control commands can also be sent through the Google Assistant App.

One major limitation of this paper lies in the fact that the central server (**ESP32** Microcontroller) must be installed only in a location where there is Internet service provision (Wi-Fi) with enough signal strength to handle the required communication between the remote users and the home appliances that needed to be automatically monitored or controlled from afar.

However, this limitation does not affect the usefulness of the innovative Home Automation System to local users who can make use of the available manual switches to control the outputs and inputs of the installed 6-channel relay.
the significance of this Internet-based Home Automation System of the paper can never be overemphasized. Deploying a traditional hardwired Home Automation System may not pose a serious challenge as long as the system is planned before hand and installed during the physical implementation of the building. If, however, already existing buildings should be augmented with Home Automation Systems, this may require so much effort and a lot of costs since cabling is necessary. For obvious reasons, a wireless system is the best option in such an awkward situation.

In the past few years, wireless technologies reached their breakthrough. Wireless based systems, used every day and everywhere, ranges from wireless home networks and mobile phones to garage door openers. As of today, little comparative research of wireless automation standards has been done, although such knowledge would provide valuable information to everyone looking for the most suitable system for given requirements.

Of course, wireless systems like WLAN have become more and more common in Local Area Networking. Also, in home and building automation systems, the use of wireless technologies gives several advantages that could not be achieved using a wired network only. The benefits include reduced installation costs, easy deployment, installation, and coverage, system scalability and easy extension, aesthetical benefits, as well as integration of mobile devices. For all these reasons, wireless technology is not only an attractive choice in renovation and refurbishment of buildings but also for new Home Automation System installations.

II LITERATURE REVIEW OF RELATED WORKS

Several works have been done with various approaches deployed towards realizing home automation system. A Bluetooth-based solution was explored by J. Castro and J. Psota, in *"The Specification, Design, and Implementation of a Home Automation System."* published by

Massachusetts Institute of Technology (MIT) in 2004 for the benefit of home automation technologists who were using Bluetooth-enabled devices to provide the control without Internet connectivity. Here, the appliances which are wired to the embedded controller are accessed and controlled by devices with built-in Bluetooth facility. However, Bluetooth has a maximum range of operation of about 100m and this limitation renders the system incapable of coping with long distant mobility and by this means restricting the system control to within the neighborhood.

Again, a GSM-based solution for the communication and control of home appliances have also been offered by T. Rozita, C. KokWai and V. H. Mok in *"Smart GSM-Based Home Automation System."* as published in **IEEE Conference on Systems, Process & Control (ICSPC2013), Kuala Lumpur, Malaysia** organized in 2013. It discusses a solution where a mobile phone (or GSM modem) is integrated into the home controller and receives different commands for the control. This system suffers a serious lack of graphical user interface (GUI) for user- friendly operation. Thus, the users have to remember different short or cryptic codes for different operations. Also, a message can be delayed due to failure of mobile network operators; hence, the solution is not suitable for real-time monitoring as well as long distant control of home appliances.

Theoretical framework: With the popularity of the Internet gateways at homes such as broadband modem and mobile hotspot, remote access for controlling home appliances is increasingly becoming practicable. In *"Design and Implementation of a Wi-Fi Based Home Automation*

System," written by E. Ahmed and A. H. Karim and published by **World Academy of Science, Engineering and Technology vol. 6, pp. 1856-1862** in 2012, a Wi-Fi based home automation solution is presented. The arrangements in that system usually pose a resource bottleneck as they require complicated network traffic routing for remote operations. A similar architecture was also described by D. N. Pratiksha, G. G. Jayashree, U. K. Pornima, and G. B. Amol in "*Design and Implementation of cloud-based Home Automation," which was* published in **International Journal of Engineering Research & Technology (IJERT), vol.**

3, no. 2, pp. 2059-2062 in February 2014. The publication in that journal describes how local web servers are deployed at home with applications developed to manage the devices over the Internet. The drawbacks of these setups are that, deployment of a high-end computer will not only increase the cost of installation but also the energy consumption and the space required for the system by virtue of its size. The developed interface applications running on the home servers are not easily upgradable and the data communication protocols employed are not robust and scalable to support any future demand.

While there are no dedicated severs at the client premises in R. Piyare's *"Internet of Things: Ubiquitous Home Control and Monitoring System using Android-based Smart Phone"* as published by **International Journal of Internet of Things, pp. 5-11** of 2013, the allotment of a public IP address makes the system expensive and chokes the limited addressing resources. Moreover, the deployment of Representational State Transfer (RESTful) based web service, as an interoperable application layer does not offer a full-duplex communication for real-time operations. To improve the previous designs, a Cloud-Enhanced Home Controller (CEHC) architecture where the localized resources are augmented with cloud scheme has been proposed by Y. Igarashi, M. Hiltunen, K. Joshi, and R. Schlichting in *"An Extensible Home Automation Architecture based on Cloud Offloading"* published on pages 187-194 of the **18th IEEE International Conference on Network-Based Information Systems (NBiS) report** of September 2015.. Although, the work attempted to provide a flexible source of rich applications in the growing automation technology, it overlooked the associated pressing security issues.

Furthermore, an implementation of a fully cloud-based solution is presented by I. Korkmaz, S. Kumova, A. Gurek, C. Gur, C. Gurakin and M. Akdeniz, in *"A Cloud based and Android Supported Scalable Home Automation System,"* as published in 2014 by **Computers and Electrical Engineering, Elsevier on pages 112-128**. Their own solution is leveraging on the Google Cloud Messaging (GCM) service for communication between the distributed cloud platforms. GCM is a free service that allows messages transfer in server-client based applications as described in Google's *"Google Cloud Messaging (GCM),"* for Google Developers. [Available **Online**]. The solution uses Extensible Messaging and Presence Protocol (XMPP). Although, the Push technology outsmarts the polling and long polling techniques, it is a heavyweight protocol streaming Extensible Mark-up Language (XML) and its big specification sees no complete implementation. In addition, unless a particular contract with Google is considered with some charges applied, there is no restriction from using the system's data for other purposes other than storage without users' consent.

Unlike the study carried out in 2013 by R. Piyare as contained in his *"Internet of Things: Ubiquitous Home Control and Monitoring System using Android-based Smart Phone"*, the research work presented in this report carefully deploys robust and scalable protocols to ensure seamless communication between the ESP32 microcontroller and the connected home appliances. Remarkably, this setup provides a simplified model of a flexible home automation system, and it eliminates the costs of a dedicated public IP address as well as of a high-end computer, thereby providing a cost-effective and secure alternative to all the other home automation systems that are readily found in today's marketplace.

III MATERIAL AND METHOD

The aim of this research paper is to develop and implement a prototype Home Automation System that is capable of remotely monitoring (checking) and efficiently controlling (changing) the operational status of certain electrical and electronic appliances in a typical modern home. As graphically depicted in the block diagram of the system (see **Figure 1.0** below), this aim is effectively achieved with the use of a versatile microcontroller and a relay module that connects all the applicable domestic devices to the Internet through a web server (**Sinric Pro**) and a mobile device (such as smart phone) which is powered by an Android application.

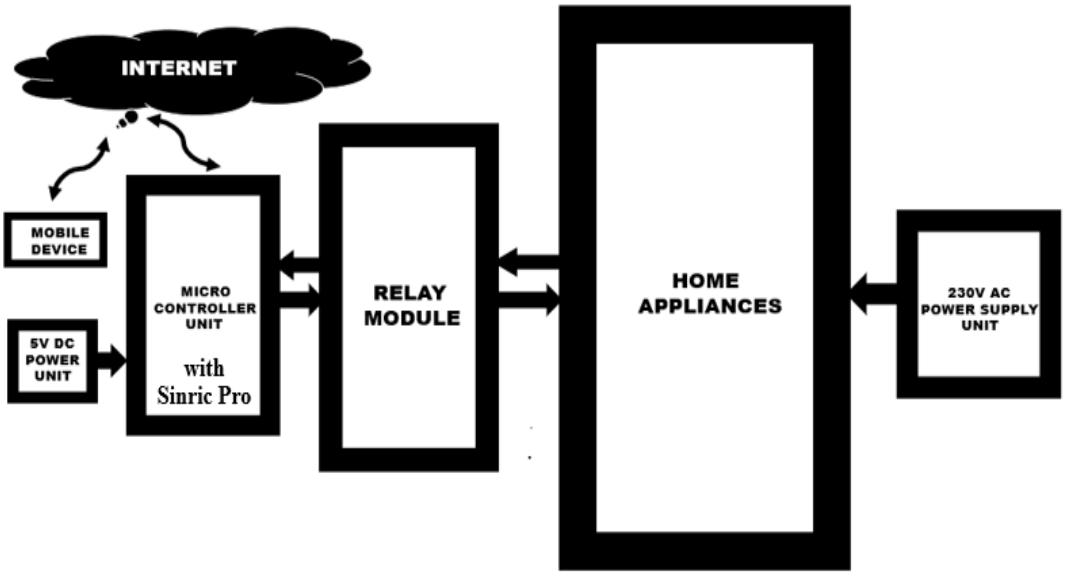


Figure 1. Block Diagram of the Prototype Home Automation System

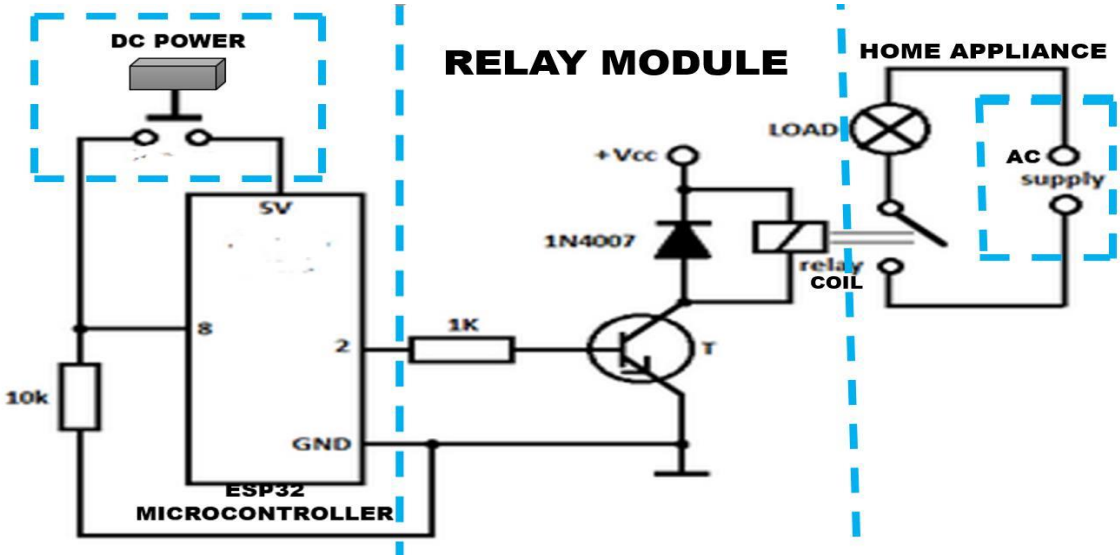


Figure 1.1: The Circuit Diagram of the Work

Basically, the research paper is conceived and developed to achieve the following three objectives:

1) To reduce the implementation cost of home automation projects by totally eliminating the usually high cost of cabling and drastically reducing the recurrent cost of having a dedicated public IP address for each of the domestic appliances to be automated as well as the cost of a high-end computer for the monitoring and control of the home appliances.

2) To ease the process of installation by eliminating the need for the expensive cabling of a new house or the rewiring of an existing building and to extend the coverage area of home automation systems by simply deploying wireless network nodes in the areas where cabling may not be feasible, cost-effective, or desirable for aesthetic reasons.

3)To enhance the security of homes by maintaining the database of all the applicable domestic appliances as well as all the legitimate users of the home automation system and integrating a security surveillance camera in the system.

The method for realizing the first objective: As stated above, the first objective of this paper is to limit the cost of ownership of a home automation system. And being completely a wireless solution, no cable is required for implementing the Internet-based home automation system. Since no amount of money is required to be spent on cabling, this simply means that the first objective of this work has inherently been fulfilled but on a partial basis. This is because there are yet other aspects of the wireless solution that have avoidable cost implications to deal with.

For instance, the concept of Internet-of-Things technology makes it mandatory for each of the home appliances involved to have a public IP address for proper identification and operation. In itself, the recurring cost of having a dedicated public IP address for each home appliance to be automated is prohibitive. However, the use of a web server (Sinric- Pro, to be precise) in the implementation of this work has immensely helped to considerably reduce the huge cost that is typically associated with the idea of assigning a public IP address to each domestic appliance. The method adopted entails setting up a Sinric Pro account and carefully going through the process as described below:

Sinric Pro account setup: For this smart home paper, a Sinric Pro Account was created and used. And the six (6) devices to be automated were added to the Sinric Pro account. The **Google Home App** from Google Play Store was also downloaded and installed. Then after creating the Home in the Google Home app the whole setup was connected to the Sinric Pro with the Google Home app to control the appliances with the Google Assistant. Note that before connecting the Google Home, all the devices in Sinric Pro have to be added first by going through the following steps to add Sinric Pro with Google Home:

1. Tap on the "+" icon, then select **Set up device**.
2. Tap on **Works with Google**
3. Search for Sinric Pro, then tap on **Sinric Pro**.
4. Enter the email id & password used for the Sinric account, then tap on **Sign in**.

After this, the next step is to return to the home screen of Google Home App and then all the devices in your Google Home App can be seen.

Controlling devices with Google Home or Google Assistant: After connecting the Sinric Pro, go to the home page of the Google Home app to see the connected devices from Sinric Pro. Now, if the ESP32 is connected to a Wi-Fi, the appliances can be controlled from Google Home app. At this stage, Google Assistant can be verbally asked to control each of the appliances and it will respond accordingly by either coming ON or going OFF.

The final result of the foregoing procedures is that all the home appliances to be monitored and controlled with the prototype Internet-based home automation system can be separately identified by the microcontroller without having a dedicated public IP address for each of them.

The method for realizing the second objective: The second objective of this work is all about facilitating the process of installing a given home automation system and extending the coverage area of the system. Since the prototype system is essentially based on a wireless technology (i.e., Internet), the difficulties that are often involved in the electrical wiring of a new building or the rewiring of an existing house are absolutely not part of the challenges facing the developer.

The development and installation of a home automation system for a new building simply involve a detailed consideration of the quality and availability of the required data communication network service. And to satisfactorily design an Internet-based home automation system, the developer has to, first of all, analyze the necessary hardware and software components that are involved starting from the home appliances to the user's smartphone. **Figure 3.2** (shown below) represents the overall system layout of a typical home automation system where everything is controlled wirelessly. The analysis covers the overall system layout, the technologies usually deployed for the software part of the work, the web security and the communication interface of the Internet.

The figure below (**Figure 1.1**) shows the structural develop as well as the overall layout of the model Internet-based home automation system which essentially comprises of four parts – the Internet, the users' mobile devices, the home appliances, and the Home |Automation System itself

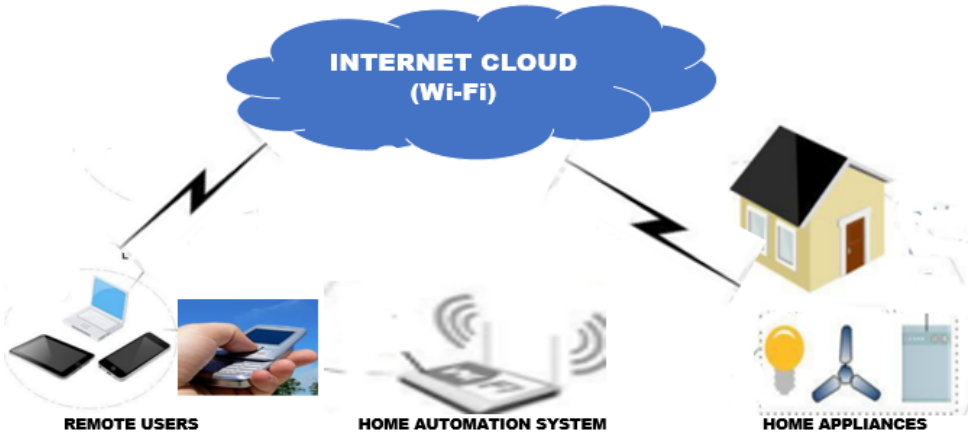


Figure 1.1: Overall Sytem Layout

As clearly shown in the block diagram of this work (see **Figure 1.0** above), the hardware component of the prototype system is basically made of seven parts as listed below:

1. The hardware Interface Module (**BESTEP 6-Chennel 5V Relay Module**),
2. The Microcontroller Unit (**ESP32S Development Board**),
3. The DC Power Supply Unit (5v Power Bank),
4. The 220V 50Hz AC Power Supply Unit
5. The Internet Cloud
6. The Mobile Device (Smart Phones, Laptop, or Computer)
7. The Home Appliances (Bulbs, Camera, TVs, Doors, Fridge, Etc.)

To extend the coverage area of a home automation system in order to fully realize the second objective of this paper, all that is required to be done is to install as many network nodes as possible and to increase the capacity of the **hardware interface module** (i.e. the relay module) so as to connect the needed number of home appliances. In the development and implantation of the prototype system presented in this report, only a 6-channel relay module was used. But the hardware interface module can readily be upgraded or downgraded to meet the particular need of any end-user or the scope of any budget.

The hardware interface module: The hardware interface module for this paper is the relay module which provides the appropriate interface for controlling the appliances in the home as illustrated with a single channel relay module shown in **Figure 1.2** below.

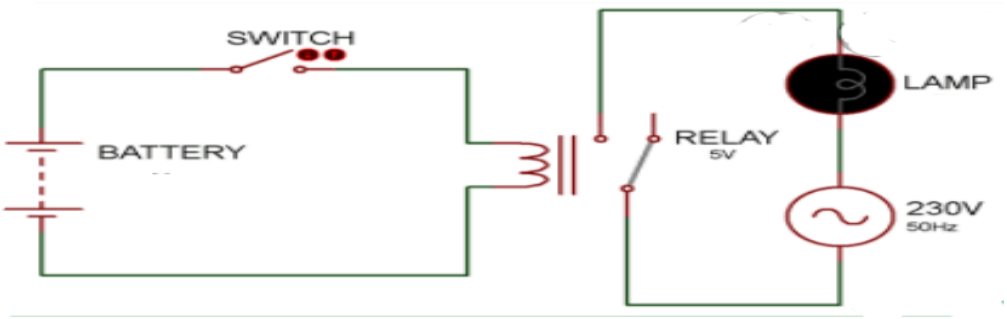
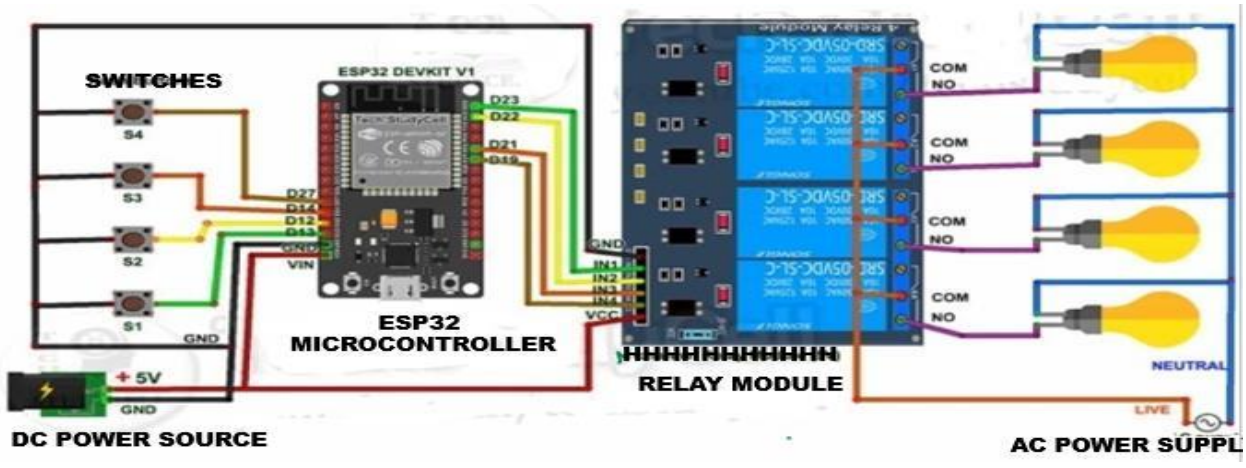


Figure 1.2: A Typical Circuit Diagram of a Relay

A 6-channel relay module is used in the implementation of this work. This relay module is, in turn, coordinated by the ESP32 microcontroller as shown in Figure 3.5 which is the circuit diagram of a section of the construction.

It is possible for the microcontroller to handle more than one relay module to control as many lighting bulbs, security surveillance cameras, as well as many other different types of home appliances. Please note that the hardware interface modules are meant for the user's interactions with the home appliances as well as for the Internet connectivity.



The hardware component of a typical Internet-based Home Automation System consists of the microcontroller, the DC power supply unit, the relay module, and the AC power supply unit. The controller is a ubiquitous, low-power, 32-bit microprocessor-based ESP32 Wi-Fi module which is a self-contained SoC with integrated Transmission Control Protocol/Internet Protocol (TCP/IP) stack used as an application processor and allows our embedded unit to communicate over wireless network. This module is chosen because of its powerful on-board processing and storage capabilities that allow for interfacing with the sensors and other application-specific devices through its general-purpose input/output units (GPIOs) with minimal development upfront and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry while it also contains a self-calibrated RF allowing it to work under all operating conditions and requires no external RF parts. The embedded Wi-Fi module has features such as 802.11 b/g/n, Wi-Fi direct (P2P), soft-AP, 81 Mb RAM, up to 160 MHz speed,
1 Mb flash memory and+19.5 dBm output power. The home appliances are controlled by the power actuators which are majorly relays; electromechanical elements that switch high- voltage, current or power devices with small electrical signals (pre-amplified through transistors) usually from digital controller circuits.

In the develop of this work, a 6-channel relay module is interfaced via a shift register to allow the control with fewer pins from the controller. Some Light Emitting Diodes (LEDs) serve as the indicators for the states of the digital output pins and for configuration. For monitoring the ambient meteorological conditions, quantities such as temperature and light intensity can also be measured. Such measurements enable the system to interact with its environment and, as a smart device, should be able to control the environment based on these measurements.

Circuit Diagram of the prototype system: The circuitry for the prototype Home Automation System is as shown in Figure 1.2 below.
Pins **D23, D22, D21 & D19** GPIO are used to control the 6-channel relay module. And the GPIO **D13, D12, D14 & D27** pins are connected with switches to control the relay module manually. The **INPUT_PULLUP** function in Arduino IDE is used instead of using the pull- up resistors with each push button. As per the source code, when the control pins of the relay module receive a **LOW** signal the relay will **turn on** and the relay will **turn off** for the **HIGH** signal in the control pin. A 5V 5Amp mobile charger is used to supply the circuit.
Required components for the ESP32 projects are as follows:

1. ESP32 Development Kit
2. 6-channel 5V Relay Module
3. A manual Switches or Push button

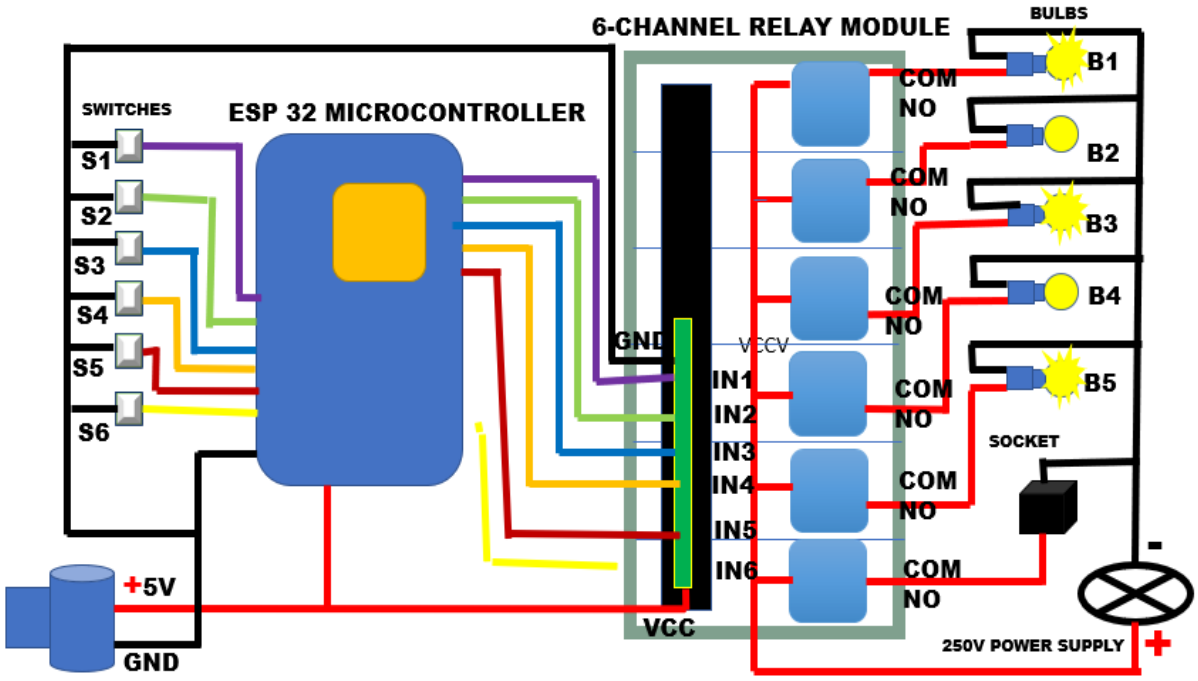


Figure 1.3: Circuit Diagram of the Internet-based Home Automation System

IV. The method for realizing the third objective: The third and final objective of this paper deals with the important issue of access control and the security of lives and property at homes. And the approach adopted in this paper for the attainment of this particular objective is firmly hinged on the software control component of the home automation system as well as the integration of a security surveillance camera to the system.
The prototype system of this project was diligently developed such that only authorized users can login to the system (locally, or remotely) in order to manage, monitor, control, or automate the operation of all the selected home appliances. If the system detects intruders, it can immediately alert the homeowner and then disable the login capability for a while.

V The software control component: This consists of a web server (**Sinric Pro**), Android-driven client applications, and the database software embedded in the microcontroller. The web server presents the core functionality for managing, controlling, authenticating and monitoring the home automation system processes while the mobile client applications provide the Graphical User Interfaces (GUIs) for client's operations. Structural design involving the operational use of design patterns and frameworks is cleverly deployed in the software system development.

The front-end applications include the web and mobile applications which provide graphical interfaces for the control and monitoring of user's equipment and sensors. The web application is open-source using the HTML5 technology – HTML, PHP, JavaScript, CSS and MySQL. It is launched through a web browser of a smart phone like the Google Chrome.

The mobile application is an adaption of the web application using the cross-platform development framework (PHP Cordova). Features such as flexibility, intuitiveness, memory efficiency, and uncluttered operation were considered for greater user experience. The embedded software is written in C/C++ and it coordinates the input and output devices interfaced to the ESP32 microcontroller. It also runs embedded mini web and WebSocket servers to support offline operations taking advantage of the dual-mode (access point and station) of the on-board Wi-Fi radio. This is to augment the system in case of link interruption to the Internet (cloud) services and to enable independent use within the home.

The programming code for the prototype System: The source code for the prototype Home Automation System is hereby provided in **Appendix.**

The communication interface: Wi-Fi technology is selected to be the network infrastructure that connects together the web server and the hardware interface module. Wi-Fi is chosen in order to improve system security (by using secure Wi-Fi connection), and to increase system mobility and scalability.

The web service security: A standard web service security technique was employed in the applications as well as in the communications between them. These include authentication system built into the web and mobile applications and as such, an unauthorized user cannot gain entrance into the mobile and web applications. To further secure the WebSocket implementation, a one-time password (OTP) mechanism is deployed. Upon valid authentication on the application console, a randomly generated 50-character long token is firstly sent to the WebSocket server for communication authorization to allow further exchange of messages.

The security surveillance camera: A digital security camera is integrated into the prototype system to enhance the security of lives and property at home. The camera is to help in the detection of intruders or in alerting the homeowner of any danger that may be lurking in and around the home environment. Once an unsafe act or unsafe condition is captured by the surveillance camera, a necessary security measure can promptly be taken against the threat to prevent the occurrence of the potential fatality or loss of property.

IV RESULTS AND DISCUSSIONS

The analysis in this implementation is specifically focused on the results of this research paper which can be summarized as follows

1) A comparative reduction in the cost of the implementation and ownership of home automation systems due to the fact that no wiring or rewiring is required to execute any wireless system and neither the recurrent cost of having a dedicated public IP address for each of the domestic appliances to be automated nor the cost of installing, operating, and maintaining a high-end computing facility for the monitoring and control of the home appliances will any longer be incurred in the course of putting an Internet-based home automation system in place;

2) An easier way of installing a home automation system in a new or existing building as well as extending the coverage area of the system by simply deploying additional wireless network nodes in places where cabling may not be feasible, cost-effective, or desirable for aesthetic reasons;

3) An enhanced security of homes as a result of the maintenance of the database of all the relevant domestic appliances together with all the legitimate users of the home automation system and the integration of a security surveillance camera in the home automation system.

Discussions on each of the results of the paper: Each of the above-mentioned results of the research paper is

further discussed below for the sake of emphasis

Reduction in the costs of implementation and ownership: Without the need for any cabling or re-cabling, implementing a wireless home automation has turned out to be more cost-effective and cheaper when compared with the cost of executing an equivalent hardwired paper. As clearly stated method, the wireless solution presented in this paper report does not require any amount of money to be spent on cabling. But then again, there are yet other aspects of the wireless solution that have some avoidable cost implications which were also properly addressed to form part of the first pleasant outcome of the research paper.

First and foremost, the concept of Internet-of-Things technology makes it mandatory for each of the home appliances involved to have a public IP address for proper identification and operation. In itself, the recurring cost of having a dedicated public IP address for each home appliance to be automated is prohibitive. However, the use of a web server (Sinric- Pro) in the implementation of this project has immensely helped to reduce, to the barest minimum, the huge cost that is typically involved in assigning a dedicated IP address to each and every appliance in a home.

Secondly, in the design of a wireless home automation system, hundreds of network nodes may be needed to provide the required automation to all the appliances in a building. But the user may just require a competitive performance to be delivered at a low system cost (compared to hardwired networks). Again, there is need for protocols to scale up to a high node count to ensure prompt signal delivery for quality response time. These needs were fully met in the design and construction of the prototype system presented herein.

Ease of installing a system and extending its coverage: Another important outcome of the research paper being reported here is the total elimination of the difficulties that are often associated with the electrical wiring of a new building or the rewiring of an existing house in the course of implementing a home automation construction. The empirical performance of the prototype system has indicated that, from the installation, scalability and flexibility point of view, an Internet-enabled home automation system is better than most of the commercially available systems around the world. Scalability is the ability of a system, network, or process to handle varying amount of workload in an efficient and effective manner. A home automation system must have an inherent flexibility to be enlarged or reduced to accommodate any necessary change in size. For example, the upgrade or downgrade of any scalable Internet-based home automation system by adding or removing hardware interface (i.e. relay) module has been demonstrated to be an easy, seminal, and systematic task.

Even if, the end-user of the system intends to add new hardware interface modules out of the coverage of the central access point, repeaters or managed wireless LAN will perfectly solve that problem. The main function of the web server installed in the microcontroller is to manage, monitor, and control all the distributed system components. This enables hardware interface modules to execute their assigned tasks (through actuators), and to feed the server with triggered events (from sensors). In setup mode, a user can add and remove hardware interface modules, and can create basic macros involving simple triggers and to customize the macros to perform complex series of events. Macros can be activated manually or in response to certain signals from devices like thermostats, motion sensors, smoke-detectors, and security surveillance cameras. A user can also program macros to activate at random. This feature allows a system to turn the lights on or off at random or at semi-random intervals.

In normal operational mode, the hardware interface modules (i.e. relay modules) feed the web server with some events and also execute their pre-programmed macros. Hardware interface modules are directly connected to home appliances through hardwire connections. Hardware interface modules have the capabilities of controlling energy management systems like lighting, thermostats and HVAC (Heating, Ventilation, And Cooling) systems, and security systems (door locks, cameras, motion detectors, fire alarms, etc.)

The devices of a home automation system are usually dispersed over large areas. Since transceivers must not consume so much power, they cannot be built with a transmission range sufficient for sensors to reach associated controllers or actuators directly. Furthermore, they may also rely on a network of access points and a hardwired backbone infrastructure consisting of sensitive receivers mainly. With the home automation system in place, users can easily manage system locally or remotely through easy web-based interface.

Enhanced security of home and authentication of users: This is the third and final result of this paper. It has satisfactorily addressed the key issue of access control and the security of lives and property at homes.

The prototype system of this paper was diligently designed such that only authorized users can login to the system (locally, or remotely) in order to manage, monitor, control, or automate the operation of all the selected home appliances. If the system detects intruders, it can immediately alert the homeowner and then disable the login capability for a while.

The Wi-Fi technology is selected to be the network infrastructure that connects together the web server and the hardware interface module. Wi-Fi is chosen in order to improve system security (by using secure Wi-Fi connection), and to increase system mobility and scalability.

A standard web service security technique was employed in

the applications as well as in the communications between them. These include authentication system built into the web and mobile applications and as such, an unauthorized user cannot gain entrance into the mobile and web applications. To further secure the WebSocket implementation, a one-time password (OTP) mechanism is deployed. Upon valid authentication on the application console, a randomly generated 50-character long token is firstly sent to the Web Socket server for communication authorization to allow further exchange of messages.

To further enhance the security of lives and property at the home front, a security surveillance camera is also integrated into the home automation system to facilitate the process of monitoring the surrounding environment of the home. And once an unsafe act or unsafe condition is captured by the surveillance camera, a signal will be sent to the system to trigger an alarm that will deter any intruder or trespasser from wreaking any havoc. A necessary security measure may also be promptly taken against the threat so as to prevent the occurrence of the potential fatality or loss of property. The versatility of an Internet-based home automation system is, indeed, limited only by the imagination of the System Analyst or Designer.

It should be noted that in order to have a fully functioning security surveillance system, all of the following needs must be met:

a) The requisite number of cameras.
b) Hard drive storage for video footage.
c) Appropriate power and cables.
d) Appropriate video recording software and/or hardware.
e) At least one sort of video monitor to view the footage.

CONCLUSIONS

In this paper report, the method of developing and implementing an Internet-based Home Automation System with the use of the Internet-of-Things (IoT) technology is discussed. How to deploy the emergent technology to integrate all the controllable devices in the home automation system is carefully explained.

Essentially, a home automation system involves the use of computer or communication gadgets to automatically monitor, control, or regulate the operation of certain electrical and electronic appliances that are installed in a home. Such household fittings include electric bulbs, door locks, Heating, Ventilation, and Cooling (HVAC) systems, television sets, etc. The list of the items also includes the alarm system from the home security system, smoke detectors, CCTV cameras, and all the other devices that can be properly connected to the smart home system's architecture and accessed via the relay module and the web server installed in the microcontroller where the IP address associated with each group of appliances is embedded.

At producing a reliable, cost-effective, secure, ubiquitously-accessible, auto-configurable, and remotely-controlled home automation system. The approach adopted in this project is novel and has the potential to achieve the target of controlling home appliances remotely using the Internet technology to interconnect different parts of a system, satisfying user needs and operational requirements. Since the new Internet- enabled solution is more efficient when compared with any of the previously existing home automation systems, the following conclusions can be drawn from the final outcomes of the research paper:

1. Home automation is soon going to be the next big step in the lives of homeowners. Once the technology is made both readily available and largely affordable, users will key into it because most homes have access to the Internet and many people have smartphones. What is left now is to create a unified home automation system where the home appliances are all connected allowing the homeowner to control every aspect of their functions.
2. With the use of mobile devices, homeowners can now automate their respective homes for enhanced comfort, safety, and security.
3. Home automation system can help users to save money. Their energy bills will be reduced as they can remotely turn off devices, which they are not using. This is a huge convenience, and the users will have complete control of their household appliances and devices, without any additional effort or extra cost.
4. Smart home devices are no longer a luxury, but a necessity for every household owner. Installing smart home devices not only offers the users peace of mind (knowing that their home is safe and secure) but also provides them added time to spend with their loved ones, instead of belaboring themselves over routine or boring household chores.
5. In a nutshell, a smart home system can help people to increase their productivity and reduce the average time they spend on household tasks. Home automation systems are generally meant for efficiency and effectiveness in the monitoring and control of household appliances.
6. Although the actual link characteristics depend on the available Internet connection strength, the system performance is guaranteed even at relatively poor network service conditions.
7. Finally, an Internet-based home automation system is better (from the scalability and flexibility point of view) than most of the commercially available home automation systems. Great flexibility in the automation operations is attained through web applications and services development for intuitive GUI mobile and web applications.

Contribution to knowledge: The ultramodern Home Automation System presented in this work report utilizes

the Internet-of-Things technology which is gradually becoming fashionable lately. The research work is borne out of the need to deal with the existing challenges of lack of security, inflexible architecture, and high cost of implementation and ownership of those smart systems that are commercially available nowadays. The prototype system consists of a microcontroller that connects some home appliances to the Internet through a relay module and a mobile application. The microcontroller is for receiving signals from the Internet for controlling the domestic appliances with the relay module. The microcontroller, through a web server application, maintains the database of all the appliances and all the authorized users of the home automation system. The web server app is also used for communication between the Internet and the mobile phone. An Android app acts as the software console that presents the user with a list of all the devices to interact with in the smart home.

REFERENCES

1. J. Castro and J. Psota, "The Specification, Design, and Implementation of a Home Automation System," Introductory Digital Systems Laboratory, MIT, 2004.
2. T. Rozita, C. KokWai and V. H. Mok, "Smart GSM Based Home Automation System," in IEEE Conference on Systems, Process & Control (ICSPC2013), Kuala Lumpur, Malaysia, 2013.
3. E. Ahmed and A. H. Karim, "Design and Implementation of a Wi-Fi Based Home Automation System," World Academy of Science, Engineering and Technology, vol. 6, pp. 1856-1862, 2012.
4. D. N. Pratiksha, G. G. Jayashree, U. K. Pornima G. B. Amol, "Design and Implementation of Cloud based Home Automation," International Journal of Engineering Research & Technology (IJERT), vol. 3, no. 2, pp. 2059-2062, Feb. 2014.
5. R. Piyare, "Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone", International Journal of Internet of Things, pp. 5-11, 2013.
6. Y. Igarashi, M. Hiltunen, K. Joshi, and R. Schlichting "An Extensible Home Automation Architecture based on Cloud Offloading" 2015 18th IEEE International Conference on Network-Based Information Systems (NBiS), pp.187-194, Sept. 2015.
7. Korkmaz, S. Kumova, A. Gurek, C. Gur, C. Gurakin and M. Akdeniz, "A Cloud based and Android Supported Scalable Home Automation System," Computers and Electrical Engineering, Elsevier, pp. 112-128, 2014.
8. Google, "Google Cloud Messaging (GCM)," Google Developers. [Online]. Available: https://developers.google.com/cloud-messaging/gcm. [Accessed Jun. 7, 2022].,
9. WAMP, "The Web Application Messaging Protocol," WAMP. [Online]. Available: http://wamp-proto.org/. [Accessed Jun. 01, 2016].
10. WAMP, "The Web Application Messaging Protocol," WAMP. [Online]. Available: http://wamp-proto.org/. [Accessed Jun. 01, 2016].

APPENDIX
SOURCE CODE OF THE PROTOTYPE SYSTEM

```
// Uncomment the following line to enable serial debug output

//#define ENABLE_DEBUG

#ifdef ENABLE_DEBUG

#define DEBUG_ESP_PORT Serial

#define NODEBUG_WEBSOCKETS

#define NDEBUG

#endif

#include <Arduino.h>

#include <WiFi.h>

#include "SinricPro.h"
```

```
#include "SinricProSwitch.h"

#include <map>

#define WIFI_SSID         "CARITAS UNIVERSITY"

#define WIFI_PASS         "EE/2017/597"

#define APP_KEY           "a830e721-2d72-4c5f-8af9-d7aaa7163621"      // Should look like

"de0bxxxx-1x3x-4x3x-ax2x-5dabxxxxxxxx"

#define APP_SECRET        "44824d04-9764-40f8-9584-9273b48f4e9e-9ce985b9-23f8-4d62- b7c6-0b5898f1694a"        //
Should look like "5f36xxxx-x3x7-4x3x-xexe-e86724a9xxxx-
4c4axxxx-3x3x-x5xe-x9x3-333d65xxxxxx"

//Enter the device IDs here

#define device_ID_1   "62c005680aec232058f3ed99"    // BEDROOM

#define device_ID_2   "62deaffa4dd95ec7bdb9bd5a"   //  TOILET

#define device_ID_3   "62deadee0bb100538664c435"   //  KITCHEN

#define device_ID_4   "62c003ce0aec232058f3ec71"   //  PARLOR

#define device_ID_5   "62c008bc0aec232058f3f147"   //  SOCKET

#define device_ID_6   "62deae3a4dd95ec7bdb9bb3a"   //  CAMERA

// define the GPIO connected with Relays and switches

#define RelayPin1 23  //D23 BEDROOM

#define RelayPin2 22  //D22 TOILET

#define RelayPin3 21  //D21 KITCHEN
#define RelayPin4 19  //D19 PARLOR

#define RelayPin5 18  //D18 SOCKET

#define RelayPin6 5  //D5   CAMERA

#define SwitchPin1 13  //D13

#define SwitchPin2 12  //D12

#define SwitchPin3 14  //D14

#define SwitchPin4 27  //D27

#define SwitchPin5 26  //D26

#define SwitchPin6 25  //D25
```

```
#define wifiLed   2   //D2

// comment the following line if you use a toggle switches instead of tactile buttons

//#define TACTILE_BUTTON 1

#define BAUD_RATE   9600

#define DEBOUNCE_TIME 250

typedef struct {      // struct for the std::map below int relayPIN;
  int flipSwitchPIN;

  } deviceConfig_t;

  // this is the main configuration

  // please put in your deviceId, the PIN for Relay and PIN for flipSwitch

  // this can be up to N devices...depending on how much pin's available on your device ;)

  // right now we have 4 devicesIds going to 4 relays and 4 flip switches to switch the relay manually
  std::map<String, deviceConfig_t> devices = {

  //{deviceId, {relayPIN,  flipSwitchPIN}}

  {device_ID_1, {  RelayPin1, SwitchPin1 }},

  {device_ID_2, {  RelayPin2, SwitchPin2 }},

  {device_ID_3, {  RelayPin3, SwitchPin3 }},

  {device_ID_4, {  RelayPin4, SwitchPin4 }},

  {device_ID_5, {  RelayPin5, SwitchPin5 }},

  {device_ID_6, {  RelayPin6, SwitchPin6 }}

  };

  typedef struct {      // struct for the std::map below

  String deviceId;
  bool lastFlipSwitchState;

  unsigned long lastFlipSwitchChange;

  } flipSwitchConfig_t;

  std::map<int, flipSwitchConfig_t> flipSwitches;     // this map is used to map flipSwitch PINs to deviceId and handling debounce
  and last flipSwitch state checks
  // it will be setup in "setupFlipSwitches" function, using informations from devices map void setupRelays() {
  for (auto &device : devices) {           // for each device (relay, flipSwitch combination)
```

```
int relayPIN = device.second.relayPIN; // get the relay pin pinMode(relayPIN, OUTPUT);          // set relay pin to OUTPUT
digitalWrite(relayPIN, HIGH);}}

void setupFlipSwitches() {
for (auto &device : devices)  {                  // for each device (relay / flipSwitch combination) flipSwitchConfig_t flipSwitchConfig;
// create a new flipSwitch configuration flipSwitchConfig.deviceId = device.first;                    // set the deviceId
flipSwitchConfig.lastFlipSwitchChange = 0;          // set debounce time flipSwitchConfig.lastFlipSwitchState = true;       // set
lastFlipSwitchState to false (LOW)-- int flipSwitchPIN = device.second.flipSwitchPIN;     // get the flipSwitchPIN
flipSwitches[flipSwitchPIN] = flipSwitchConfig;       // save the flipSwitch config to
flipSwitches map

pinMode(flipSwitchPIN, INPUT_PULLUP);                 // set the flipSwitch pin to INPUT}}

bool onPowerState(String deviceId, bool &state)

{

Serial.printf("%s: %s\r\n", deviceId.c_str(), state ? "on" : "off");

int relayPIN = devices[deviceId].relayPIN; // get the relay pin for corresponding device digitalWrite(relayPIN, !state);              //
set the new relay state
return true;

}

void handleFlipSwitches() {

unsigned long actualMillis = millis();                          // get actual millis
for (auto &flipSwitch : flipSwitches) {

// for each flipSwitch in flipSwitches map

   unsigned long lastFlipSwitchChange = flipSwitch.second.lastFlipSwitchChange;  // get the timestamp when flipSwitch was
pressed last time (used to debounce / limit events)
if (actualMillis - lastFlipSwitchChange > DEBOUNCE_TIME) {                 // if time is >

debounce time...

     int flipSwitchPIN = flipSwitch.first;                          // get the flipSwitch pin from configuration
     bool lastFlipSwitchState = flipSwitch.second.lastFlipSwitchState;              // get the lastFlipSwitchState
     bool flipSwitchState = digitalRead(flipSwitchPIN);                       // read the current flipSwitch state
     if (flipSwitchState != lastFlipSwitchState) {                    // if the flipSwitchState has changed...
#ifdef TACTILE_BUTTON

if (flipSwitchState) {                                  // if the tactile button is pressed

#endif

        flipSwitch.second.lastFlipSwitchChange = actualMillis;                    // update lastFlipSwitchChange time
        String deviceId = flipSwitch.second.deviceId;                  // get the deviceId from config
        int relayPIN = devices[deviceId].relayPIN;                    // get the relayPIN from config
bool newRelayState = !digitalRead(relayPIN);                  // set the new relay State

digitalWrite(relayPIN, newRelayState);                        // set the trelay to the new
state
```

```
SinricProSwitch &mySwitch = SinricPro[deviceId];                    // get Switch devie
from SinricPro

mySwitch.sendPowerStateEvent(!newRelayState);                       // send the event

#ifdef TACTILE_BUTTON

}

#endif

        flipSwitch.second.lastFlipSwitchState = flipSwitchState;                                    // update lastFlipSwitchState}}}}void
setupWiFi(){Serial.printf("\r\n[Wifi]: Connecting"); WiFi.begin(WIFI_SSID, WIFI_PASS); while (WiFi.status() !=
WL_CONNECTED)
{Serial.printf("."); delay(250);}digitalWrite(wifiLed, HIGH); Serial.printf("connected!\r\n[WiFi]: IP-Address is %s\r\n",
WiFi.localIP().toString().c_str());}
void setupSinricPro(){for (auto &device : devices){const char *deviceId = device.first.c_str(); SinricProSwitch &mySwitch =
SinricPro[deviceId]; mySwitch.onPowerState(onPowerState);
}SinricPro.begin(APP_KEY, APP_SECRET); SinricPro.restoreDeviceStates(true);
}void setup(){Serial.begin(BAUD_RATE); pinMode(wifiLed, OUTPUT); digitalWrite(wifiLed, LOW); setupRelays();
setupFlipSwitches();setupWiFi(); setupSinricPro();}void loop(){SinricPro.handle();handleFlipSwitches();}bool
onPowerState(String deviceId, bool &state){Serial.printf("%s: %s\r\n", deviceId.c_str(), state ? "on" : "off");int relayPIN =
devices[deviceId].relayPIN; // get the relay pin for corresponding device digitalWrite(relayPIN, !state);          // set the new relay
state
return true;}void handleFlipSwitches() {unsigned long actualMillis = millis();                              // get actual millis for (auto
&flipSwitch : flipSwitches) {                                       // for each flipSwitch in flipSwitches map unsigned long
lastFlipSwitchChange = flipSwitch.second.lastFlipSwitchChange;  // get the timestamp when flipSwitch was pressed last time
(used to debounce / limit events) if (actualMillis - lastFlipSwitchChange > DEBOUNCE_TIME) {          // if time is
>debounce time...int flipSwitchPIN = flipSwitch.first;                                  // get the flipSwitch pin from configuration
    bool lastFlipSwitchState = flipSwitch.second.lastFlipSwitchState;                   // get the lastFlipSwitchState bool
flipSwitchState = digitalRead(flipSwitchPIN);                                  // read the current flipSwitch state if (flipSwitchState !=
lastFlipSwitchState) {                          // if the flipSwitchState has changed...#ifdef TACTILE_BUTTONif (flipSwitchState) {
// if the tactile button is pressed#endifflipSwitch.second.lastFlipSwitchChange = actualMillis;                 // update
lastFlipSwitchChange time String deviceId = flipSwitch.second.deviceId;                 // get the deviceId from config int
relayPIN = devices[deviceId].relayPIN;                          // get the relayPIN from config
bool newRelayState = !digitalRead(relayPIN);                    // set the new relay State

digitalWrite(relayPIN, newRelayState);                          // set the trelay to the new
state
SinricProSwitch &mySwitch = SinricPro[deviceId];                                    // get Switch devicefrom
SinricPromySwitch.sendPowerStateEvent(!newRelayState);                                  // send the event#ifdef
TACTILE_BUTTON}#endifflipSwitch.second.lastFlipSwitchState = flipSwitchState;                   // update
lastFlipSwitchState}}}}void setupWiFi()
{Serial.printf("\r\n[Wifi]: Connecting"); WiFi.begin(WIFI_SSID, WIFI_PASS); while (WiFi.status() !=
WL_CONNECTED){Serial.printf(".");delay(250);}digitalWrite(wifiLed, HIGH);Serial.printf("connected!\r\n[WiFi]: IP-Address
is %s\r\n", WiFi.localIP().toString().c_str());}
void setupSinricPro(){for (auto &device : devices){const char *deviceId = device.first.c_str(); SinricProSwitch &mySwitch =
SinricPro[deviceId];mySwitch.onPowerState(onPowerState);}SinricPro.begin(APP_KEY,APP_SECRET);
SinricPro.restoreDeviceStates(true);}voidsetup(){Serial.begin(BAUD_RATE);pinMode(wifiLed,OUTPUT);
digitalWrite(wifiLed, LOW); setupRelays(); setupFlipSwitches(); setupWiFi();setupSinricPro();}void loop(){SinricPro.handle();
handleFlipSwitches();}
```