

Implementation of a Cloud-Based Home Automation System

***A. O. Otuoze, N. T. Surajudeen-Bakinde, A. T. Abdulrahman and O. H. Isiwepeni**

Department of Electrical and Electronics Engineering, University of Ilorin, Ilorin, Nigeria.

*Corresponding Author: otuoze.ao@unilorin.edu.ng

Tel.: +2348032777695, +2348137288435, +2347069214332, +2348036789781

Abstract

Home automations provide full control and flexible monitoring of home appliances for increased comfort, efficient energy use, costs saving, improved safety and security, and have driven the development of various smart homes technologies. This paper presents the implementation of a home automation system utilizing the Internet of Things (IoT) technology. This work tackles the problems of complexity and incompatible standards inherent in the existing home automation solutions using a robust distributed computing approach. The embedded system operates on dual Advanced Virtual RISC (AVR) microcontrollers, ATmega328p and ATmega16L, interfaced to input and output modules – sensors, actuators, Wi-Fi shield, LCD, keypad etc. – for interactions with users, appliances and the environment. The HTML5 based intuitive mobile and web applications, developed for the system, induces unrestricted flexibility in the system operations and management. The system deploys a novel Web Application Messaging Protocol (WAMP), implemented with WebSocket full-duplex and persistent connection protocol and JSON data serialization, for seamless individual applications and subsystems integration with a relatively high level of security through web service security protocol. This cloud-based home automation solution has lower implementation cost compared to the existent systems because, as it removes the expensive costs of on-premise high-end computers and dedicated public IP addresses.

Keywords: Internet of Things (IoT), Distributed Computing, WebSocket, HTML5, AVR Microcontrollers, Arduino UNO, Wi-Fi

Introduction

The demand for automation systems both in homes and industries is on the increase because of their numerous benefits such as comfort, centralized control of appliances, cost reduction, energy saving, security, and safety. Home automation provides increased quality of life for users, especially the elderly and disabled, and enables monitoring of children activities as well as controlling their accesses to resources at home.

In recent years, the Internet, especially cloud computing, is used in seemingly unlimited applications in all spheres of life. The Internet has made the interconnectivity of virtually every object possible – human social networks and even machine-to-machine (M2M) communications. The use of smart devices in daily activities is increasing exponentially, and consequently their intelligence getting enhanced, due to much higher affordability and simplicity through their connectivity [1].

The concept of the "Internet of Things" has tied in closely with the popularization of home automation [2]. The Internet of Things (IoTs) is an evolving technology which has received quite a lot of attention from researchers following the vision of a global infrastructure of networked physical objects. While this vision is enthralling, no consensus exists about its realization [3]. IoT involves integrating smart objects; embedded devices with sensors and actuators connected to the Internet. These devices are intelligently interconnected thereby necessitating new forms of communication between things and people, and between things themselves [4].

However, early smart homes suffered poor performance, high cost of ownership, complicated set-up and operation, poor management and maintenance, and in many cases, the need to rewire the home [5]. The first "wired homes" were built by American hobbyists during the 1960s but were limited by the technology at the time. The term "smart house" was first coined by the American Association of House builders in 1984 [6].

A distributed home automation system will consist of various, independent, possibly heterogeneous, computing units which are referred to as nodes. These nodes collaborate together in order to achieve the desired automation functionalities by exchanging data (passing messages) to synchronize their current state. These messages exchanged between the

nodes of the home automation distributed system are usually terse as compared to human-to-human network data [7]. This is in addition to the input and output data operation of the individual systems and applications [8]. Recently, novel home automation is fast evolving and consequently there have been different procedures with vast of them based on the wireless and Internet technologies relating to the concept of “Internet of Things”.

In this work, the problems with complexity, multiple incompatible standards and the resulting expenses in these recent systems by providing a simplified design protocol and developing a robust distributed home automation system are addressed.

Related works

Many researchers proposed many designs on the home automation technology. Although this review covers a wide variety of such works, it will focus on the major themes which recur throughout the literatures reviewed. These targets are: incorporation of wireless connectivity to home automation systems, deployment of cloud computing to home automation technology, and the dedication of Internet Protocol addresses to servers sited at the location of installation. The review of these themes will primarily focus on their application to Internet based home automation.

The works presented in [9-11] offered Bluetooth based solutions to the home automation technology using Bluetooth enabled devices to provide control without the Internet connectivity. In this case, the appliances, wired to the embedded controller are accessed and controlled by devices with built-in Bluetooth connectivity. And with the maximum range of 100 meters for Bluetooth connections, a great limitation is posed rendering the systems incapable of coping with long distance mobility and by this means restricting the system’s control to within the neighborhood.

Global System for Mobile (GSM) based solutions for automation of home appliances have also been presented by [12] and [13] where a mobile phone (or GSM modem) is incorporated to the home controller and receives set of AT commands for controlling different appliances. Lack of graphical user interface (GUI) for supple operation has however posed another great deal since users have to remember different short codes for

different operations. Also, a message can be delayed for long due to failure of mobile network operators; hence, the solution is not suitable for real-time monitoring as well as long distance data logging (telemetry).

The widespread use of Internet gateways at homes such as broadband modem and mobile hotspot has made remote access to control home appliances more practicable. In [14], a Wi-Fi based home automation deploying a web server which manages the connected home appliances was presented. Similar works in [5], [14] and [15] show designs in which dedicated web servers and web applications are developed to manage the devices with the Internet. The drawbacks of such design approaches, especially [5] and [14], are that deployment of a high-end computer at the client's premises will not only increase the cost of installation but also the energy consumption and space.

Also, the presence of the home server annihilates the essence of a distributed system with dependency of embedded system, which otherwise should be an independent node, on the server for connectivity. The Java written desktop applications in both [5] and [15] for the home servers are not easily upgradable and the data communications employed are not robust. While in [4], where a dedicated server at the client premises is absent, the allotment of a public IP address, which is precious and limited, makes the system to be ineffective in terms of the cost. Moreover, the deployment of RESTful based Web service, as an interoperable application layer, can be better replaced by Web Application Messaging Protocol WAMP for real-time operations and efficient computing resources use.

Flexibility is induced in the automation operations through intuitive HTML5 based mobile and web applications. Generally, this system removes the costs of a dedicated public IP address as well as of a high-end computer and provides a cost-effective solution to home automation. Also, the problems with complexity and multiple incompatible standards are addressed by providing a simplified design protocol and developing a robust distributed home automation system.

The presented work utilizes ultramodern web technologies realizing a distributed home automation system with the processes as services. A robust data communication protocol, WAMP, is deployed which ensures

seamless communication between the individual applications and systems. This novel protocol utilizes WebSocket full-duplex and persistent connection and JSON data serialization and gives the administrator(s) total control of all connected appliances. Also, a relatively high level of security was achieved by the virtue of the robust web service security protocols used.

The System Architecture

The distributed home automation system, as shown in Figure 1, comprises mainly two components:

- Hardware interface component
- Software control component

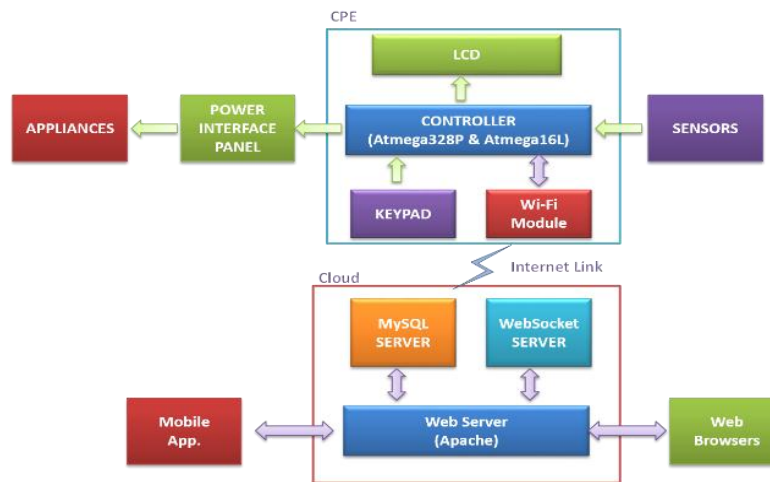


Figure 1: Overall system block diagram

Hardware Interface Component

The hardware is the physical part of the system and forms the client premises equipment (CPE). It provides the appropriate interfaces to sensors and actuators as well as other input and output devices which are coordinated by a mini embedded operating system (embedded OS) written in C/C++ programming language. The CPE provides control of

lightings, HVAC (heating, ventilating and air conditioning) appliances, security locks of gates and doors and other systems, as well as monitoring of the house environments (such as the room temperature, humidity and light intensity) through various sensors [16]. It also provides interfaces for user interactions as well Internet connectivity. As a result, the embedded hardware is further divided into: dual-core controller, power device actuators, integrated sensors, network card, and input/output user interface devices.

Dual-core controller

The dual-core controller for the embedded hardware system is jointly formed by two Atmel Advanced Virtual RISC (AVR) microcontrollers: ATmega328P and ATmega16L. The ATmega328P is the core of the Arduino UNO microcontroller development board which facilitates easier and faster development. The ATmega328P is chosen to serve as the primary controller and due to its few pins count, the auxiliary controller (ATmega16L) supplements for multiple peripheral interfaces.

Arduino UNO is an embedded system development board based on ATmega328P microcontroller. It has a total of 28 pins with 14 digital input/output pins (6 of which can be used as PWM outputs) and 6 analogue inputs. The board runs on an external crystal oscillator of 16MHz and because most of the recent 8-bit AVR microcontrollers take an oscillator cycle per instruction. With this, the system will run 16 Mega Instructions per second (MIPS) [17].

The UNO board contains everything needed to support the microcontroller. This include a voltage regulator (which can provide 3.3 and 5.0 V derived from a 6-20 V external power supply through a power jack), a reset button, on-board programmer interfaced via universal serial bus (USB) and so on. The board differs from other existing boards in that it features ATmega16U (programmed as a USB-to-serial) in lieu of the conventional FTDI USB-to-serial driver chip [17].

The ATmega328P has 32 Kilobytes of flash memory for storing the program code, 2 Kilobytes of static random-access memory (SRAM) and 1 Kilobyte electronically erasable programmable read-only memory (EEPROM) [18]. These features and other facilities enable the Arduino board interconnect with a computer, another microcontroller as well as

other communication devices.

The ATmega16L is a low-power (by the virtue of the type 'L'-lower power) Complementary Metal-Oxide Semi-conductor (CMOS) 8-bit microcontroller based on the AVR enhanced RISC architecture, just as ATmega328P, executing powerful instructions in a single clock cycle to achieve 1 MIPS per MHz. This allows the project design to optimize power consumption versus processing speed (the two foremost factors in hardware design). This functions as an auxiliary microcontroller for the interface to the liquid crystal display (LCD) and the keypad as it has 40 pins out of which 32 pins are input/output (grouped into ports of 8 pins) [19].

The two microcontrollers exchange information via their highly flexible serial communication devices, Universal Synchronous and Asynchronous Serial Receiver and Transmitter (USART). USART offers multi-processor communication in full duplex operation at an agreed data transfer speed - the band rate.

Power device actuators

These are power interface devices through which the home appliances are powered as the controllers can only provide a maximum current of only few milli amperes (mA). The actuators used in the project are mainly relays. Although light emitting diodes (LEDs) are used to indicate the states of the digital output pins, they cannot drive any power devices.

A relay is an electromechanical device which switches high-voltage, current or power devices with small electrical signals usually from digital electronic circuits. Using a power transistor with the microcontroller will allow the control of outputs that require more current or higher voltages than the microcontroller can provide. The signal from the digital circuit is usually pre-amplified through switching transistors (bipolar transistors as Darlington arrays).

Integrated sensors

For a home automation system, it is a good choice to have ways of monitoring the ambient meteorological conditions (but not limited to that). Quantities such as temperature and light intensity measurements enable

the system to interact with its environment and as a smart device, it should be able to control the environment based on these measurements.

One of the components used is LM35, a precision integrated circuit temperature sensing device with an output voltage linearly proportional to the Centigrade temperature. It has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from output to obtain a convenient Centigrade scaling [20]. The sensor is powered with 5 V and outputs 10 mV per degree Centigrade. This output voltage is read by one of the six multiplexed channels of the 10-bit ADC of the ATmega328P microcontroller.

Similarly, light dependent resistor (LDR) is used to measure the ambient light intensity by changing its resistance value as the intensity of the photons that fall on it reduces. At full darkness, the resistance value goes as high as in the order of mega Ohms ($M\Omega$), and almost approaching infinity. While at full brightness, the resistance tends towards zero Ohms [21]. Basically, cell resistance falls with increasing light intensity. With the changing resistance value, this device can be put in a simple voltage divider circuit with another suitable variable resistor and the midpoint output voltage is read with another channel of the ADC of the microcontroller.

Another sensor used, which is, however, digital, is the Passive Infra-red (PIR) Motion Detector. This sensor detects any motion in the vicinity by any animated object (such as human or domestic animal) that radiates Infra-red. This implies that if any inanimate object falls off or is swept by the draught of the fan, the device will not sense it. The PIR detector snaps the Infra-red signal in its range few times and discovers the changes in the orientation of such source of Infra-red. As soon as any gesture is detected, the digital output pin of the device rises digitally high and can be optionally delayed so as to be read by a digital pin of the microcontroller [22].

Network card

Wireless Fidelity (Wi-Fi) shield utilizing Roving Network (RN 171) module connects the CPE to client home network. It basically forwards the entire packet between the USART, with which it is interfaced with the controller, to the TCP/IP transport by radiating the information signal to the nearest network access point. It features an independent antenna which can cover a wider range and transmit stronger signals.

The RN171 is a standalone embedded 802.11 b/g Wireless Local Area Network (WLAN) module. The module incorporates an on-board TCP/IP networking stack, cryptographic accelerator, power management subsystem, real-time clock, versatile sensor interface, 2.4 GHz transceiver, and RF power amplifier (with power output up to 12 dBm). With the RN171, designers can embed Wi-Fi and networking functionality rapidly into virtually any device [23].

Input/output user interface devices

For greater flexibility, the user should be able to interact with the system. Such interactions include managing the connectivity, monitoring the operations as well as debugging the system. In order to achieve this, a liquid crystal display (LCD) and a keypad are interfaced to the ATmega16L to serve as the output and input devices respectively. The LCD has 4 rows with each displaying 20 characters. While the keypad utilizes a 4-by-4 matrix multiplexed switches [24].

A dedicated port (8 pins) of the microcontroller is dedicated for the keypad (4 pins for the rows and 4 pins for the columns). Another port is used for both the data bus (LCD in 4-bit mode) and the control lines [25]. Both the ports are interfaced directly to the ATmega16L.

The Software Control Unit

This part consists of a web server and client applications. The web server presents the core for managing, controlling, authenticating and monitoring the distributed system processes. The client applications provide graphical user interfaces (GUIs) for client's operations. Email notification is featured in the client application to allow the relay of instant messages in case of any security or safety breach. The whole software sub-system

would be open-source based applications implemented using the HTML5 technology. HyperText Mark-up Language (HTML), Hypertext Preprocessor (PHP), JavaScript, Cascade Style Sheet (CSS), and MySQL Database Query Language make up the HTML5 technology.

Software is used for a long time, updated and maintained by people who did not write them. For this reason, structural design involving the operational use of design patterns [26] [27] and frameworks is deployed in the software system development. Since the overall system is a distributed system, there is need for a systematic approach in the development of the software which directs the operations of all the hardware involved. Therefore, this subsystem is further divided into the following components: frontend GUI applications, backend server applications, and embedded controller system firmware. And for the application to meet the expectations of the end users, the following are essential features the software should have:

- Flexibility in use
- Intuitive Graphic User Interface (GUI)
- Efficient memory usage
- Clear and uncluttered operation

Frontend GUI applications

The frontend applications include the web application and mobile application which provide graphical interfaces for the control and monitoring of user's equipment and sensors. Both the applications are developed using HTML5 web technology; HTML, CSS, JavaScript, PHP and MySQL. The web application can be launched through the web browsers of smart gadgets such as personal computers, tablets, PDAs, smartphones, etc., while the mobile application is an adaption of the web application using the cross-platform development framework, PHP Cordova (also referred to as PhoneGap).

Model-View-Controller (MVC) design pattern is used here with the model as the data saved in MySQL database, View is made up of the HTML, CSS and JavaScript contents, and the Controller are the business logics of the application implemented using PHP [28] [29]. The web applications are developed using a framework called Laravel. Laravel is a free, open

source PHP web application framework with expressive, elegant syntax, designed for the development of MVC patterned applications. It is released under the MIT license, with its source code hosted on GitHub [30]. It is a PHP framework with powerful modern features as authentication, routing, sessions, queuing, and caching for web development at warp speed [31]. Laravel is accessible because it is an open source, yet providing powerful tools needed for large, robust applications.

Backend server applications and services

The backend applications provide the core for managing, controlling, authenticating and monitoring the distributed system processes. It is also implemented using Laravel framework and is written in PHP. The backend services include data communication using Web Application Communication Protocol (WAMP) taking advantage of Ratchet PHP framework. Backend service is a cloud computing that provides the communication and data management facilities for the operation of mobile and web applications.

Backend applications and services provide the web and mobile applications a link to the backend cloud storage while also providing features such as user management, push notifications, and integration of distributed processes.

Embedded controller system firmware

The hardware subsystem is composed of different electronics modules which are interfaced to and are coordinated by the dual-core microcontrollers (ATmega32 and ATmega16L). Components such as the LCD, keypad, relay modules, Wi-Fi shield and all of the sensors as well as the actuators are interfaced to the microcontrollers using embedded C/C++ language.

Arduino sketch running on the core microcontroller is written with the rich libraries that come with Arduino. Such APIs include WiFly, Soft Serial, Arduino JSON, WebSocket etc. The ATmega16L is programmed to run a micro operating system (micro-OS) which drives some output and input operations. Such include serial communication with the Arduino, input from the keypad and the results display on the LCD. Figure 2 illustrates

the operating flowchart for the CPE.

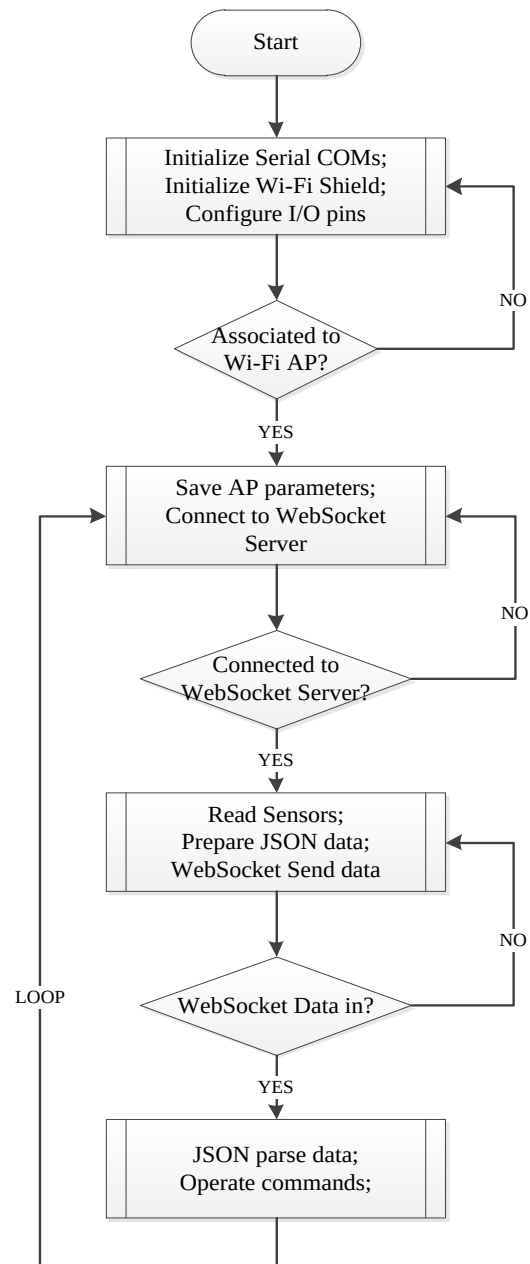


Figure 2: CPE operational flowchart

Communication Protocols

The goal here is to describe ways of connecting nodes making up the distributed system. Conventionally, networking technologies are described following an abstract model called the Open System Interconnection Reference model (OSI Reference Model or OSI Model). This model contains seven layers to describe a network architecture. From the highest to the lowest, these layers are: application, presentation, session, transport, network, datalink, and physical layer [32].

WebSocket and JavaScript object notation (JSON)

WebSocket is found to be the most suitable for real-time bi-directional, full-duplex, persistent connection from a web browser to a server. Where the web application communication protocol (WAMP) is an open standard WebSocket sub protocol which provides application routing that works with different languages [33]. With WAMP, distributed systems can be built out of applications components which are loosely coupled and communicate in soft real-time [34]. It is built over WebSocket communication protocol and JSON data serialization. JSON is an open, language-independent, human-readable, data interchange format to ideally inter-operate, store, and transfer data between systems. It is designed to be simple, generic, structured, human and machine readable, and used over the Internet. It uses a textual data format with Unicode encoding and it is fat free as opposed to its XML counterpart [35]. A typical example of a JSON formatted packet is given as;

```
{ "user_id": "1", "port": "2", "state": "0" }
```

Web service and security

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format; specifically, Web Service Definition Language (WSDL) as XML used in Simple Object Access Protocol (SOAP). Other protocols include Representational State Transfer (RESTful) which employs HTTP and JSON; and Web Application Messaging Protocol (WAMP) which employs WebSocket and JSON. However, WAMP is a much more lightweight protocol than XML SOAP and RESTful. Web services provide a standard means of interoperating

between different software applications, running on a variety of platforms or frameworks [36].

Web service security is a protocol to ensure the confidentiality and integrity of WS transactions. There are various security protocols acting at different network layers. Cryptography involves an encryption process that transforms the information into seemingly meaningless data by using an algorithm and a key. The encrypted data can only be decrypted with the key or by the extremely long process of testing all the possible keys [37].

There are different ways to authenticate a party. A common secret key can be exchanged, as in the case of password or passphrase. Nowadays, it is the most common way to authenticate a user. The main drawback is that the key has to have high entropy and be long enough. Otherwise, the key can be potentially guessed by trying all the possible keys. The password authentication is supported by HTTP, but the sent password is not encrypted. WebSocket is authenticated through the use of ticket token technique where the server gives an authorized user a one-time password (OTP).

System Implementation

Hardware Construction

Most of the hardware components are modules by the virtue of the integrated design and development adopted for the project. Therefore, standards data bus and jumper wires are used in routing all the network paths for the embedded hardware except the ATmega16L which was placed on an IC socket soldered on a Vero board and the integrated sensors as the LM35 and the LDR which were also soldered and mounted carefully on the casing.

Each of the hardware modules were tested as they were acquired and all were confirmed to be good and usable for the project. Also, during each of the stages of construction, the modules were tested and each was confirmed to work as required independently as well as conjointly. Figure 3 shows the coupled CPE in a transparent casing revealing the embedded hardware system.



Figure 3: Hardware system close-up

Software Development

For development efficiency, many frameworks were used in this project. These include Laravel PHP framework, Ratchet PHP WebSocket framework, PhoneGap mobile apps framework, and Twitter Bootstrap CSS framework. The design pattern employed is Object-Oriented pattern while the higher-level design architecture is the Model-View-Controller (MVC) pattern.

HyperText Markup Language (HTML) is used for the presentation of the contents of the frontend applications. Cascaded Style Sheet (CSS) provides further styling rules for the HTML contents to enhance the overall feel and look of the GUI applications. While a database Simple Query Language (SQL) takes care of the data (or model) in combination with PHP for the dynamic contents updating of the applications' contents. Therefore, in this project, tools that are well suited for each of the components are employed and such include: C/C++ languages for embedded controller firmware; JavaScript (JS) language for both the client-side web and mobile applications; PreHyperText (PHP) is used for the backend server-side web applications and services scripting.

The programs as well as the scripts written for the software were tested and all possible bugs were corrected during each stage in the development. Both the compilation time and the run-time errors were checked and corrected. The exception errors in the software (the applications as well as

the hardware firmware) were handled carefully and correctly.

Overall, the software applications were developed to be foolproof and really user-friendly. This was achieved through sophisticated built-in tests (BIT) embedded in the applications. After thorough codes inspection and sanitation, the applications were loaded on the local server and finally on the cloud virtual private server (VPS). The applications as well as the services are up and running now and this prototype is accessible via [<http://cb.djade.net>]. Figure 4 shows the logs of the WebSocket server while Figure 5 reveals the feel and look of the mobile/web GUI application.

```
C:\Windows\system32\cmd.exe - ssh nicemate@52.11.161.49 -p...  
:: [SERVER STATUS]  
WebSocket Server Starting...  
WebSocket Server Up and running!  
  
:: [NEW CONNECTION]  
Console ID: 1  
Console ID: p16NU2U1JQaaEMBzRVj1htWqg3cM2qfb1JUbjiYh  
Resource ID: 224  
Remote Address: 41.190.3.156  
  
:: [MESSAGE CONTENT]  
Message: <"id":1,"port":10,"state":0>  
Resource ID: 224  
Remote Address: 41.190.3.156  
  
:: [MESSAGE CONTENT]  
Message: <"id":1,"port":12,"state":1>  
Resource ID: 224  
Remote Address: 41.190.3.156  
  
:: [MESSAGE CONTENT]  
Message: <"id":1,"port":11,"state":1>  
Resource ID: 224  
Remote Address: 41.190.3.156
```

Figure 4: WebSocket server log terminal

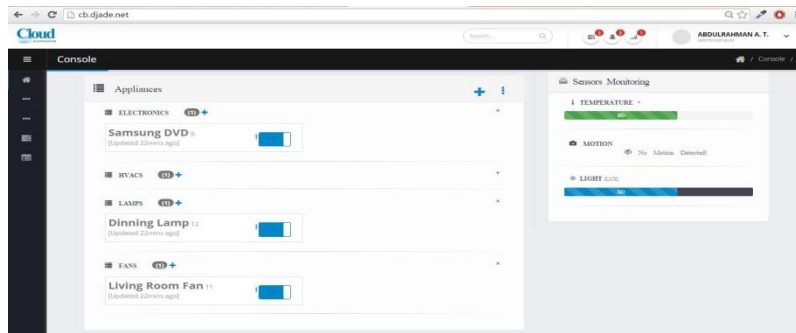


Figure 5: Mobile/web application console

Conclusion and Recommendations

In this work, the state-of-the-art web technologies were utilized and structured to render the whole home automation system a distributed type with the processes as services. The cloud portion of the distributed system involves the web applications integrated with data management and repositories as well as communication interfaces. Great flexibility was induced in the automation operations through HTML5 based web applications and services development for intuitive GUI mobile and web applications.

Similarly, modular design concept was adopted in the embedded hardware development for better functionality and greater reliability. A robust data communication protocol to ensure seamless communication between the individual applications and systems was deployed. This novel protocol is WAMP and it utilizes WebSocket full-duplex and persistent connection as well as JSON data serialization. A relatively high level of security by the virtue of the robust web service security protocol deployed was realized. Overall, the system provides a cost-effective solution to home automation as the costs of a dedicated public IP address and a high-end computer, as present mostly in other solutions, are removed.

This work aims at refining the knowledge domain for the home automation technology and the future works need to address the security challenges and concerns associated with the evolving Internet of Things (IoT) concept. It is also imperative to consider embedded hardware built on greater processing power and storage capacity to stack up the necessary

communication protocols in units of small form factor. The use of sophisticated Wi-Fi resources with higher reliability will ensure better throughput in the physical layer of the communication process. Finally, incorporation of independent offline capabilities into the system will offer cloud offloading and also ensure continuous operations in the events of constrained system resources and low bandwidth, high latency, unreliable networks or even total connection disruption.

References

- Hilton S. "Progression from M2M to the Internet of Things: An Introductory Blog", Bosch Connected World Blog, Oct. 2010 [Online]. Available: <http://blog.bosch-si.com/categories/Internetofthings/2012/10/progression-from-m2m-to-Internet-of-things-an-introductory-blog/>. [Accessed Feb. 20, 2015].
- Wikipedia, "Home Automation," Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/wiki/Home_automation. [Accessed Dec. 12, 2014].
- Kortuem G., Kawsar F., Fitton D. and Sundramoorthy V. "Smart objects as building blocks for the Internet of things", Internet Computing, vol. 14, pp. 44-51, 2010. IEEE.
- Rajeev P. "Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone", International Journal of Internet of Things, vol. 2 no. 1, pp. 5-11, 2013. DOI: 10.5923/j.ijit.20130201.02
- Jemiseye W. "Design and Implementation of Smart Home" Unpublished B.Eng. Thesis, Dept. of Electrical and Electronics Engineering, University of Ilorin, Ilorin, Nigeria, June 2014.
- Harper R. "Inside the Smart Home: Ideas, Possibilities and Methods" First Chapter in the book, Inside the Smart Home, Harper and Richard (eds.), pp. 1-13, Springer, 2003.
- Francis, D. "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", California: Apress Open, pp.5, 2013.
- Germis A. "An Open Home Automation System Analysis and Implementation of New Solutions for HomePort" M.Sc. Thesis, Dept. of Computer Science, Aalborg University, Aalborg, Denmark. Sept 2009 - May 2010.
- Piyare R. and Tazil M., "Bluetooth based home automation system using cell phone," 2011 IEEE 15th International Symposium on Consumer Electronics (ISCE), 2011, pp.192-195.
- Naresh D., Chakradhar B. and Krishnaveni S. "Bluetooth Based Home Automation and Security System Using ARM9", International Journal of

- Engineering Trends and Technology (IJETT), vol. 4, no. 9, pp. 4052-4058, Sept., 2013.
- Castro J. and Psota J. "The Specification, Design, and Implementation of a Home Automation System" Introductory Digital Systems Laboratory, MIT, 2004.
- Malik SH, Khiyal AK, and Erum S. "SMS Based Wireless Home Appliance Control System (HACS) for Automating Appliances and Security", Issues in Informing Science and Information Technology, vol. 6, 2009.
- Rozita T, KokWai C. and Mok VH. "Smart GSM Based Home Automation System", IEEE Conference on Systems, Process & Control (ICSPC 2013), Kuala Lumpur, Malaysia, 2013.
- Ahmed E and Karim AH. "Design and Implementation of a WiFi Based Home Automation System" World Academy of Science, Engineering and Technology, vol. 6, pp. 1856-1862, 2012.
- Pratiksha DN, Jayashree GG, Pornima UK. and Amol GB. "Design and Implementation of Cloud based Home Automation", International Journal of Engineering Research & Technology (IJERT), vol. 3, no. 2, pp. 2059-2062, Feb. 2014.
- Kastner W., Neugschwandtner G., Soucek S., and Newmann HM. "Communication systems for building automation and control" Proceedings of the IEEE, vol. 93, no. 6, pp.1178–1203, Jun. 2005. DOI: 10.1109/JPROC.2005.849726
- Article on Arduino, "Arduino UNO" Available: <http://www.Arduino.cc/en/Main/ArduinoBoardUno>. [Accessed January 12, 2015].
- Atmel Corporation, "8-bit AVR Microcontroller with 4/8/16/32K Bytes In-system Programmable Flash: ATmega48A/PA/88A/PA/168A/PA/328/P". ATmega328 datasheet. Available: <http://www.datasheetspdf.com/datasheet/ATmega328.html> [Accessed: January 10, 2015]
- Atmel Corporation, "8-bit AVR Microcontroller with 16K Bytes In-system Programmable Flash: ATmega16& ATmega16L". ATmega16 datasheet. Available: <http://www.alldatasheet.com/datasheet-pdf/pdf/78532/ATMEL/ATMEGA16.html> [Accessed: January 10, 2015]
- Texas Instruments, "LM35 Precision Centigrade Temperature Sensors". LM35 datasheet. SNIS159E. Aug., 1999. (Revised Jan. 2015). Available: <http://www.ti.com/lit/ds/symlink/lm35.pdf> [Accessed: January 18, 2015]
- RS, "Light Dependent Resistors". NORP12 datasheet. March, 1997. Available: <http://uk.rs-online.com/web/p/ltr-light-dependent-resistors/0651507/> [Accessed January 20, 2015].
- Adafruit Learning System, "PIR Motion Sensor". PIR datasheet. Available: <https://learn.adafruit.com/pir-passive-infrared-proximity->

- [motion-sensor/overview](#) [Accessed: February 12, 2015]
- Microchip, “2.4 GHz Std. 802.11 b/g Wireless LAN Module”. RN17 datasheet. Available: <https://www.element14.com/community/docs/DOC-71941/1/24-ghz-ieee-std-80211-bg-wireless-lan-module> [Accessed: February 12, 2015].
- Parallax Incorporated, “4x4 Matrix Membrane Keypad”. 4x4 keypad datasheet.V1.2.Available: <https://www.parallax.com/sites/default/files/downloads/27899-4x4-Matrix-Membrane-Keypad-v1.2.pdf> [Accessed: February 16, 2015]
- Hitachi, “Systronix 20x4 LCD Brief Technical Data”.HD44780U datasheet.Available: http://www.systronix.com/access/Systronix_20x4_lcd_brief_data.pdf [Accessed: February 16, 2015]
- Alexander Christopher et al., *A Pattern Language*, New York: Oxford University Press, 1977.
- Erich, G. E., *Design Patterns: Elements of Reusable Object-Oriented Software*, USA: Addison-Wesley, 1994.
- UC Berkeley IST, “Model View Controller: A Design Pattern for Software,” Berkeley IST 2004. Available: <https://ist.berkeley.edu/as-ag/pub/pdf/mvc-seminar.pdf>. [Accessed May 3, 2015].
- Eckstein, R., “Java SE Application Design with MVC,” Oracle. [Online]. Available:<http://www.oracle.com/technetwork/articles/javase/index-142890.html>. [Accessed May 2, 2015].
- Otwell, T. “Laravel Documentation” Laravel. Available: <http://www.laravel.com>. [Accessed April. 21, 2015].
- Otwell, T., “Laravel/laravel” Github. Available: <https://github.com/laravel/laravel>. [Accessed: May 2, 2015].
- International Telecommunication Union, “ITU. X.200: Information technology - open systems interconnection - basic reference model: The basic model”. Available: <http://www.itu.int/rec/T-REC-X.200-199407-I/en>. [Accessed: January 20, 2015].
- “Web application messaging protocol, WAMP”. Available: <http://www.wamp.ws>. [Accessed March 15, 2015].
- Ratchet, “Introduction to Websockets,” Ratchet. Available: <http://socketo.me/docs/>. [Accessed February 25, 2015].
- JSON, “Introducing JSON,” ECMA-404: The JSON Data Interchange Standard. Available: <http://www.json.org>. [Accessed February 25, 2015].
- Working Group, “Web Services Architecture”, W3C Working Group Note. Available: <http://www.w3.org/TR/2004/>. [Accessed January 12, 2015].
- Ross JA. “Security Engineering: A Guide to Building Dependable Distributed Systems”, Wiley, 1st ed., Jan. 2001