

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



---

# Smart Homes and Sustainable Cities: The Design of a Low-Cost Solution for Comprehensive Home Automation

---

Hubert Kalala Mukendi and Marco Adonis

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.78058>

---

## Abstract

The challenge for smart cities is to connect as many of its inhabitants to technology enabling solutions that improve their lives. Smart homes provide all users a means of interacting and impacting their environment. In developing economies this proves challenging and these challenges are daunting and overwhelming since system costs are always a foreboding factor. The chapter addresses these challenges by providing a low-cost solution for a home energy saving measure. It introduces an overview of enabling technologies for a smart home by considering energy management, energy saving, load management and monitoring and control of living spaces. By leveraging the application of the Internet of Things (IoT) and load management strategies, the realisation of a smart home is made possible. This chapter presents a broad overview of the design and development of a web-enabled smart home solution. Web development and control systems together form the backbone of automation for modern home automation technologies such as the Internet of Things and embedded systems. The developed web-enabled home automation incorporates elements of web developed software application and digital control systems. The web-enabled interface energy saving measure is a networked system that uses web-enabled applications for enabling energy efficiency by incorporating load management, remote power consumption, monitoring and control.

**Keywords:** Internet of Things, smart home, energy management, energy saving, load management, energy efficiency

---

# 1. Introduction

## 1.1. Smart homes in a sustainable city

As more people settle in cities, the increased urbanization requires cities to become sustainable. Sustainable cities provide access to a wide range of basic services such as affordable and clean energy. Globally electricity consumption rates are increasing year on year. One way in which sustainability can be achieved is by providing an affordable range of technologies that enable the management and reduction of electricity consumption. By decreasing the demand for more electricity cities can avoid the construction of more power plants. This can be realized by adopting standards for a wider range technologies that address the reduction in electricity demand. Investment in technology, infrastructure, innovation, sustainable development and scientific research is paramount if this goal is to be achieved. Advanced tools such as information and communication technology (ICT) are able to assist the realization of sustainable cities by enabling urban dwellings to be transformed into smart homes and smart spaces. These technologies can contribute to the global efforts to provide urban citizens access to clean and affordable energy and also help the efforts to preserve the environment.

## 1.2. Elements of a smart home

As an assistive technology in smart cities, home automation is able to offer a high level of comfort, sophistication, data access, environmental controls, energy management and operational efficiency. Home automation provides an avenue for users to actively participate in accessing and controlling their environments. This is achieved by delivering to users tangible means of energy saving, perimeter security, data access and control of home appliances. As an energy management tool, home automation synchronizes the technology platform with practical benefits such as the ability to reduce home electric bills through control of home

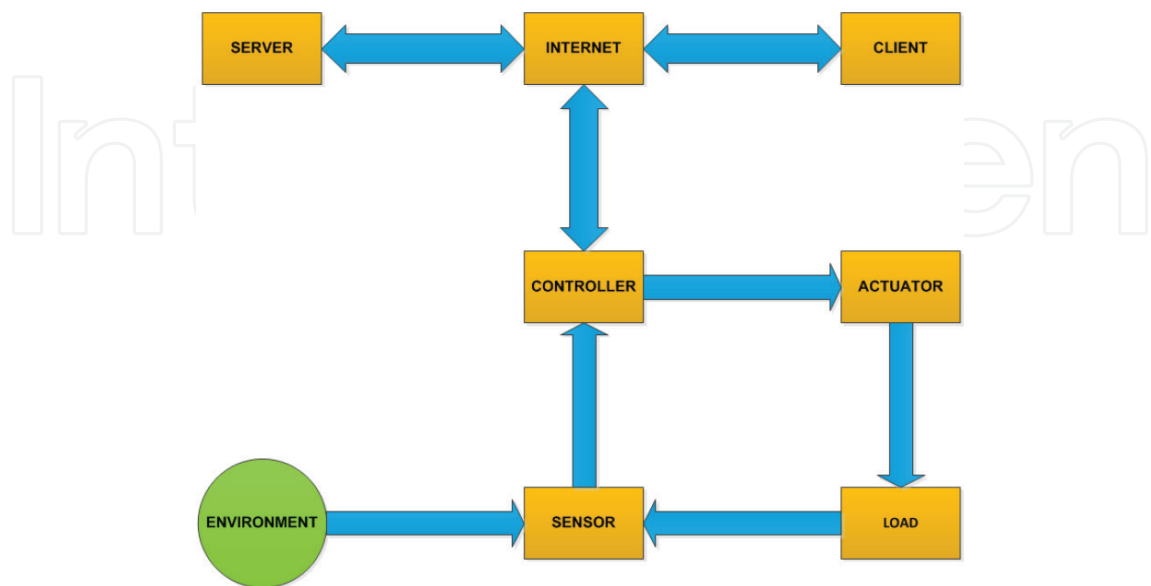


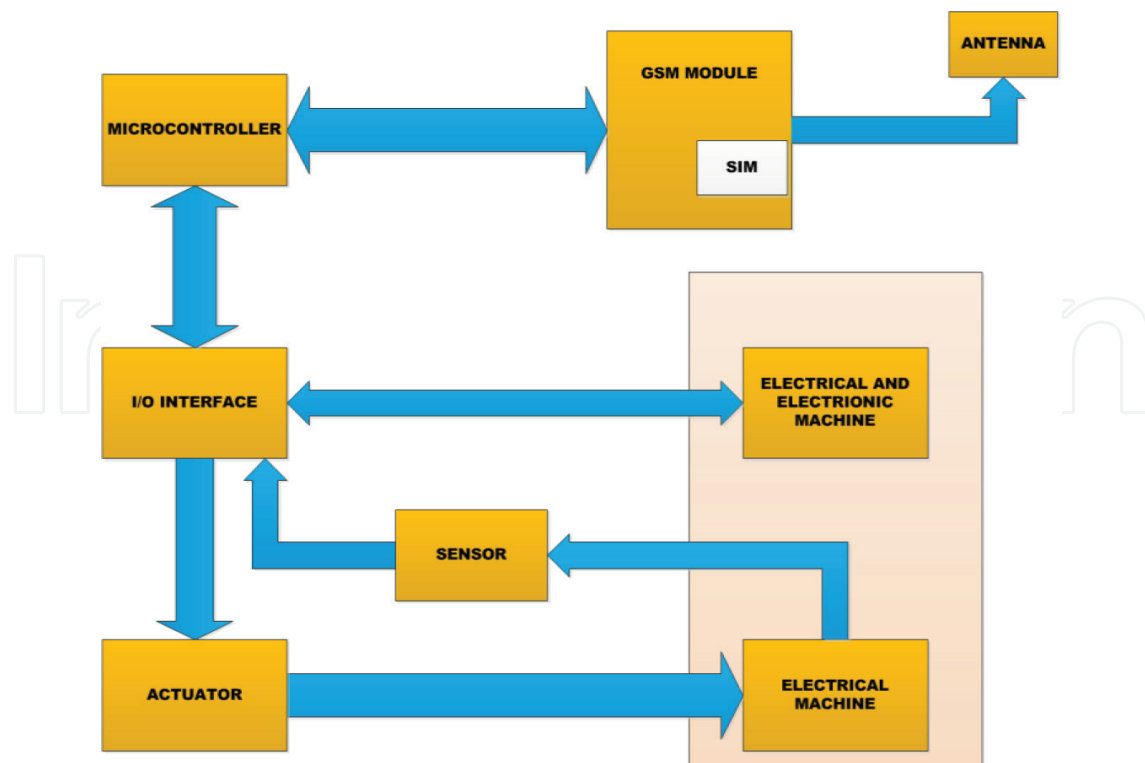
Figure 1. Home automation system elements.

electrical loads. The capability is further enhanced by the provision of remote access and resource management. This is achievable if this user desirable features include access via a web application and mobile software application. **Figure 1** indicates the typical elements of a home automation network. This represents the basic configuration encompassing sensors, actuators, controllable user loads, a main system controller and a remote user interface.

## 2. Literature review

### 2.1. Smart solutions

Modern assistive technologies should offer users smart and innovative solutions. These solutions should rely on technology to make tasks easier to manage and complete [1–3]. The rapid improvement in electronic systems such as mobile phones and home appliances are not beneficial to everyone, especially average consumers in developing countries. Energy consumption can be determined through measuring energy usage. This is accomplished using energy meters employing electrical current sensors. In electrical terms, energy consumption is the measure of how much power is used by the load side of an electrical circuit. Consumers wish to enjoy the benefits offered by smart cities such as access to information, remote sensing and automation. Facilities such as home automation can offer users some of these benefits. Recent studies have found that when users have access to their home power usage statistics, they are more readily willing to act or intervene [4, 5]. The other problem that this intervention addresses is the integrated home automation systems with customized installation where the



**Figure 2.** Mobile GSM-based home automation platform.

end users cannot make modifications for their own joy and comfort; mostly these systems are expensive and out of reach for average consumers. Inexpensive solutions are desired to address customer needs.

2.2. Home automation platforms

Recently there has been much progress made in the areas of home automation with focus on the Internet of Things (IoT) [6–8].

There have been developments in solutions presented to address home automation. The embedded controller may be provided by a host of different families of microcontrollers such as affordable variants, which include the Atmel ATmega microcontrollers [9], the Arduino [10] or Raspberry Pi [11, 12]. The propriety-based microcontrollers offer a relatively more expensive solution, which is not open-source, nor user code reconfigurable.

The communication channel has been one of the main foci. There have been solutions provided that incorporated the use of Bluetooth technology [13] as well as systems that used ZigBee communication [14]. Some of these include the use of Global System for Mobile communication or GSM network as the communication technology [15]. Such an application is presented in Figure 2.

Another variation on the communication channel targets the use of an Ethernet based Internet solution [16]. Such a system is depicted in Figure 3.

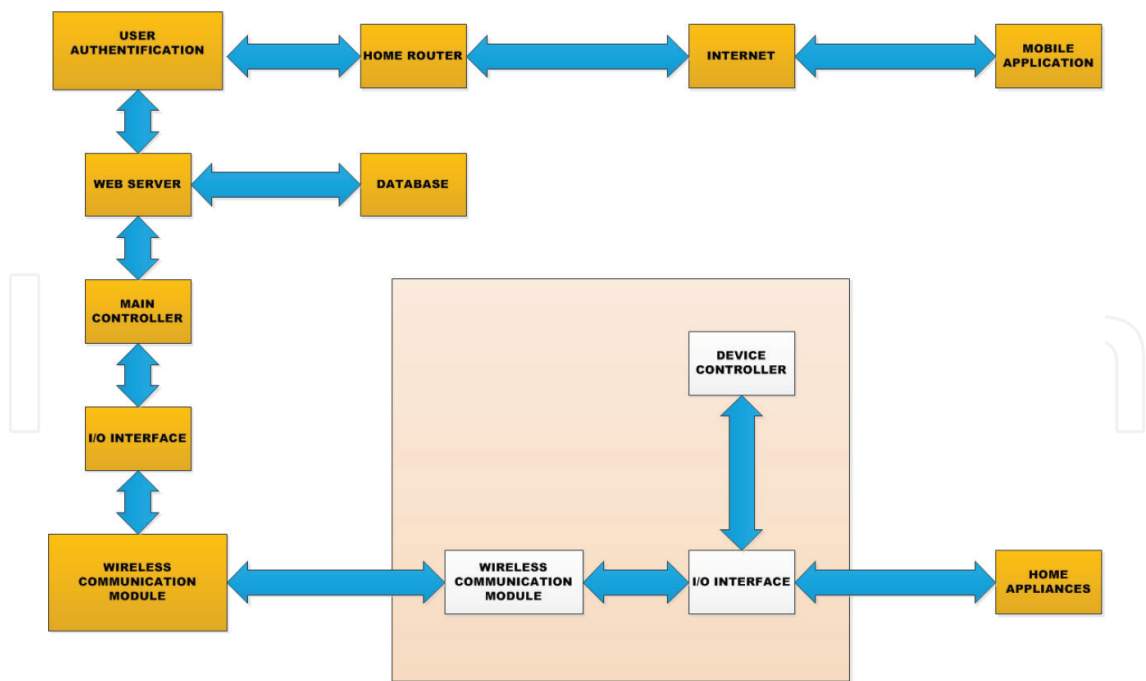


Figure 3. An Internet-Ethernet-based home automation platform.

### 3. System design and development

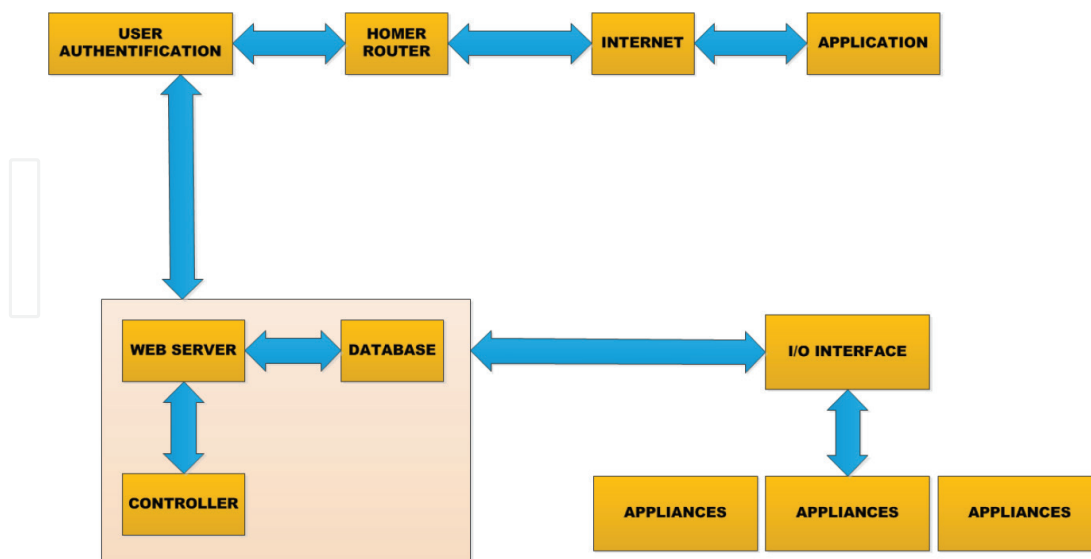
#### 3.1. Home automation solution

The system designed for this low-cost implementation uses readily available off-the-shelf components. The control algorithm is housed in an Arduino microcontroller. It is based on an open-source electronic prototyping platform and designed for ease of use. Arduino provides both the hardware components through single-board microcontrollers and also offers a software interface through its Integrated Development Environment (IDE). It is a powerful and popular embedded control solution. This makes the platform affordable for the average home user.

The solution provided here is indicated in **Figure 4**. A range of home appliances is interfaced with Arduino through its input/output terminals (I/O pins). The control and sensing actions are controlled from the microcontroller. The Arduino host the control architecture which includes the appliance database and the website server. Via the web server, users are able to access the web application on an Android device. User authentication provides added level of security allowing only authenticated users access to the system.

#### 3.2. System algorithm

The program flowchart shown in **Figure 5** depicts the decision-making capability inherent in the system solution. The user is prompted after system initialization to authenticate themselves. If the user is not previously registered in the system database, they will not be able to proceed to the next step. If authentication passes, the user enters the home interface page.



**Figure 4.** Implemented home automation system.

From this page the user is able to make system requests for data access to enable the appliance monitoring function or request appliance control status change. This allows connected devices such as user loads to be switched ON or OFF. The status change is indicated here in real-time.

3.3. The web interface design

A typical HTML document starts with a declaration defining the document type (see **Figure 6**), followed by the parent element of HTML document, the HTML tag. All HTML tags have an opening and a closing tag, tags are contained inside angle brackets <>. All visible contents of a web page are contained in the body tag between the opening <body> and closing </body> body tags. HTML documents are created using any text editing programs, mostly Notepad, Notepad++ and much more, and this document must be saved with a .htm or .html extensions. HTML document can be opened in any web browser, but each web browser responds differently to some HTML tags.

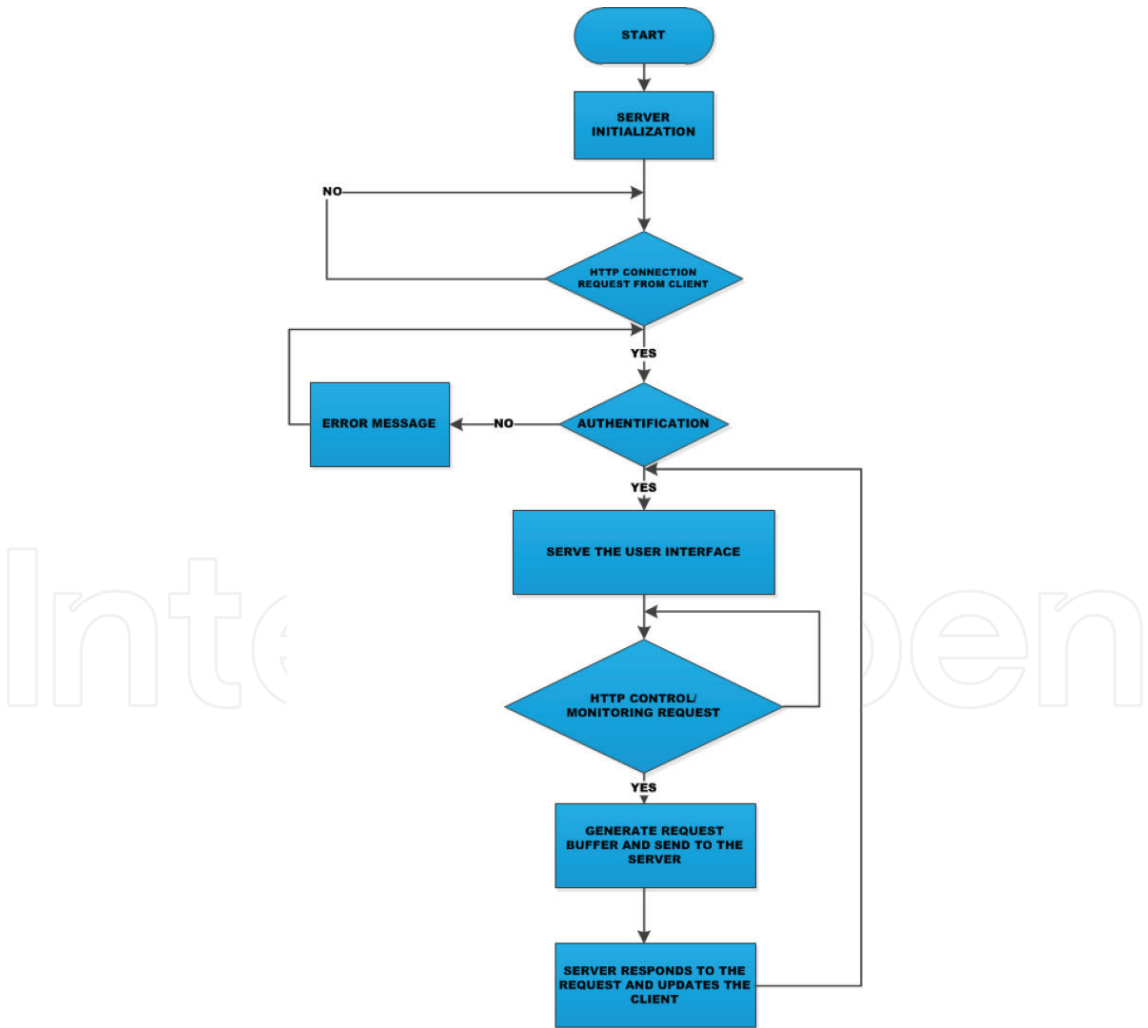


Figure 5. Program flowchart.

```
<! Doctype html>

<html>

<head>

<title>The Title Web page tile goes here</title>

</head>

<body>

<!-- THE VISIBLE CONTENT OF THE WEB PAGE GOES HERE -->

<h1>Big headings </h1>

<p>Paragraph</p>

</body>

</html>
```

Figure 6. HTML document structure.

The example shown below is the HTML structure, the first line in the HTML document should be the document type declaration, `<! Doctype html>` defines the document type and this aids the browser to correctly display the web page.

The HTML `<html>` tag is the root element of any HTML document, and it is the second line in the HTML document after the document type declaration. The HTML tag is closed last at the end of an HTML document. The head section in an HTML document is contained between the head opening tag `<head>` and the head closing tag `</head>`, the content of the head section is the style, title, and java scripts if any are used.

The body section in an HTML document is contained between the body opening tag `<body>` and the body closing tag `</tag>`, the body section contains the visible or information that need be displayed in the web page.

### 3.4. HTML design

An HTML element is made up of an opening tag `<tag_name>`, content and a closing tag `</tag_name>`. If an HTML element does not contain any content between the starting and end tag, it is called an empty element. The content portion of an HTML element can be another HTML element; in this case, the elements are said to be nested. All HTML documents are made of nested HTML elements, as shown below the HTML is an HTML element with opening tag `<html>` and closing tag `</html>`. The content of this element is another HTML element head which also has a starting `<head>` tag and a closing `</head>` tag. The head tag contains an



element, the title tag with `<title>` and `</title>` starting and closing tags respectively. The title element contains content, Home Automation Control Panel. The HTML element also contains another element that is the body element that in turn contains an image tag. Thus the figure below is good representation HTML elements and nested elements.

The style `<style>` tag is used to represent the information regarding the HTML document style. The specifications pertaining to how the web browser should display the HTML document is contained in this tag. In one HTML document, it is possible to have more than one style element.

The style `<style>` element must be contained inside the head `<head>` section of the HTML document. If a “scope” attribute is specified in the style element, then that style setting will apply only to the parent and child elements that style [8].

HTML document style could also be defined in a style sheet external to the HTML document; in this case, a link tag should be used to reference the style sheet. In the example below, `divBorder` has defined a class with red style color. This means that any content contained within an element whose class is `divBorder` will have a red color.

### 3.5. Home automation controller elements

**Figure 5** below shows a microcontroller based home automation system. The user mobile application interacts with the microcontroller via the web server using the Internet protocol. The microcontroller receives commands from the user interface and performs the required tasks based on a controlling algorithm governing the controller. The controller reads devices status and updates this data into the server for the user mobile application. Also, refer to **Figure 4** in the literature review for a typical Internet-based home automation system.

The server handles the users and ensures secure communication between the user mobile application and the controlling unit. Once a user is identified, he will then be granted access to the controlling interface (web page).

The advantages of using a microcontroller have reduced the size of circuitry, affordability, and increased flexibility. A microcontroller can be used as a substitute for other integrated circuits (IC's). It can also be easily reprogrammed to modify its functionality. The Microcontroller that was used for this project is the Arduino MEGA 2560 R3.

#### 3.5.1. The wattmeter

The Wattmeter graphic as shown in **Figure 7**, was created using the HTML platform and Java Script. The meter was designed to measure up to 30 kW of power. The green color represents the power consumption as below 10 kW. The yellow color on the gauge represents power consumption in the range above 10–20 kW. It is an indication that the user should become aware of high power consumption. The red color represents the range above 20 kW of power consumption. Here users are acutely aware that extremely high power consumption could cause severe outage. Users should immediately switch off all non-essential appliances or high power loads.

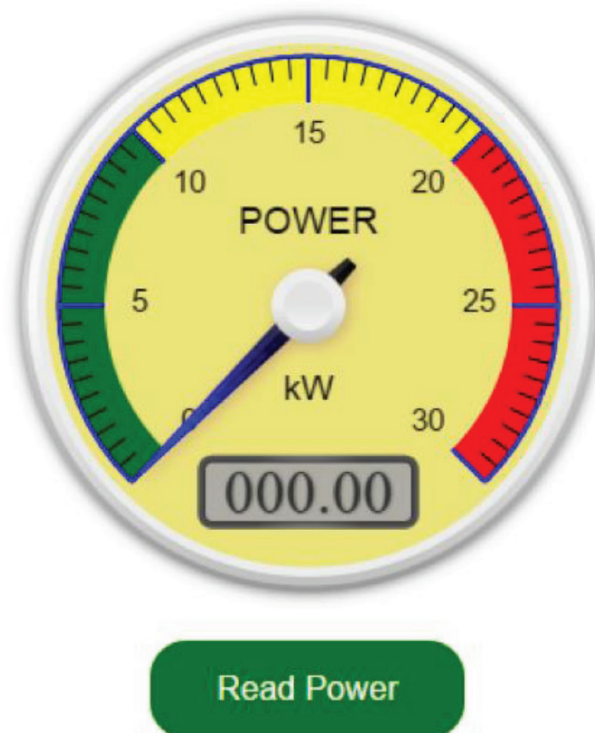
- When the “Read Power” button is clicked, a “PowerControl()” function is called. The PowerControl function generates a random together with an HTTP “GET” request. The request buffer is made up of a “GET” method, “Power” command and a random number.
- When the web server replies to the request, the “PowerControl ()” function reads the HTTP response and changes the reading the gauge with the one contained in the response from the server.
- The random number is generated and attached to the request buffer every time the function is called to avoid caching of commands on the web browser.

### 3.5.2. Appliance control

A range of home appliances can be controlled via this application. These appliances are interfaced to the application controller via relays. This functionality allows users full control over appliance ON/OFF status.

Appliance control as shown in **Figure 8** below, this section of the user interface enables a person to turn ON/OFF appliances, in this case, a light bulb, a stove, a heater and a television. The system is designed in such a way that a picture representing the current state of the appliance is updated on the user interface.

## Power Consumption



**Figure 7.** The wattmeter user interface.

- When the Light “ON” button is clicked, the “BulbControlON()” function is called. This function generates a random number together with an HTTP “GET” request and sends it to the controller which is also the web server. The request buffer is made up of a “GET” method, “BulbON” command and a random number.
- When web server replies to the request, the “BulbControlON()” function reads the HTTP response and changes the state picture of the bulb with the one contained in the response from the server.
- When the Light “OFF” button is clicked, the “BulbControlOFF()” function is called. This function generates a random number together with an HTTP “GET” request. The request buffer is made up of a “GET” method, “BulbOFF” command and a random number.
- When web server replies to the request, the “BulbControlOFF()” function reads the HTTP response and changes the state picture of the bulb with the one contained in the response from the server.
- When the Stove “ON” button is clicked, the “StoveControlON()” function is called. This function generates a random number together with an HTTP “GET” request. The request buffer is made up of a “GET” method, “StoveON” command and a random number.
- When web server replies to the request, the “StoveControlON()” function reads the HTTP response and changes the state picture of the stove with the one contained in the response from the server.
- When the Stove “OFF” button is clicked, the “StoveControlOFF()” function is called. This function generates a random number together with an HTTP “GET” request. The request buffer is made up of a “GET” method, “StoveOFF” command and a random number.
- When web server replies to the request, the “StoveControlOFF()” function reads the HTTP response and changes the state picture of the stove with the one contained in the response from the server.
- When the Heater “ON” button is clicked, the “HeaterControlON()” function is called. This function generates a random number together with an HTTP “GET” request. The request buffer is made up of a “GET” method, “HeaterON” command and a random number.

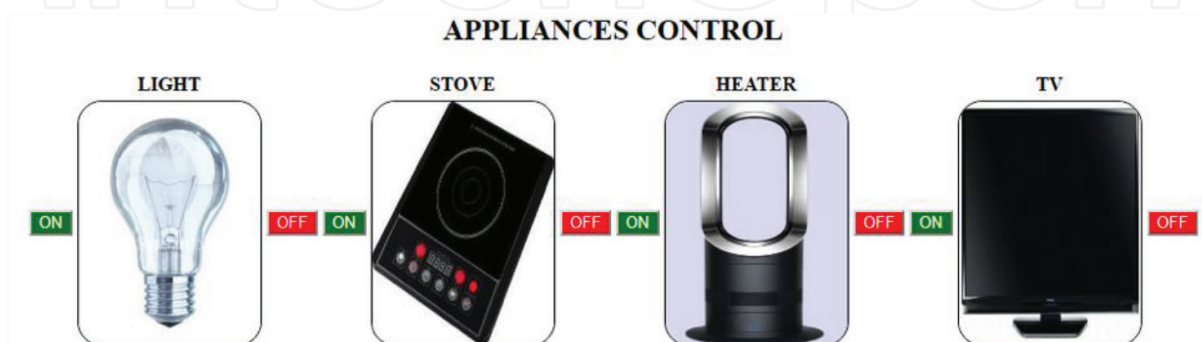
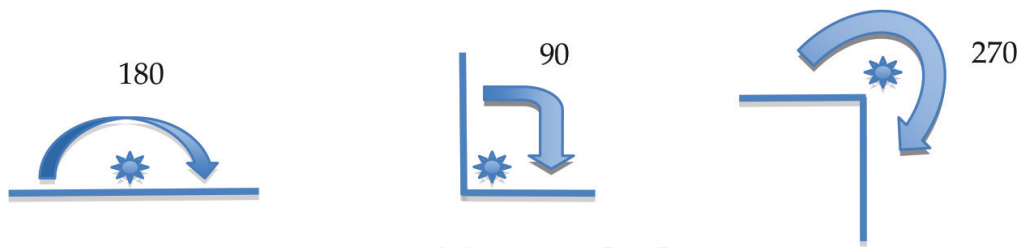


Figure 8. Home appliance control panel.



**Figure 9.** Security camera position and exterior light brightness control panel.

- When web server replies to the request, the “HeaterControlON()” function reads the HTTP response and changes the state picture of the heater with the one contained in the response from the server.
- When the Heater “OFF” button is clicked, the “HeaterControlOFF()” function is called. This function generates a random number together with an HTTP “GET” request. The request buffer is made up of a “GET” method, “HeaterOFF” command and a random number.
- When web server replies to the request, the “HeaterControlOFF()” function reads the HTTP response and changes the state picture of the heater with the one contained in the response from the server.
- When the TV “ON” button is clicked, the “TvControlON()” function is called. This function generates a random number together with an HTTP “GET” request. The request buffer is made up of a “GET” method, “TvON” command and a random number.
- When web server replies to the request, the “TvControlON()” function reads the HTTP response and changes the state picture of the tv with the one contained in the response from the server.
- When the TV “OFF” button is clicked, the “TvControlOFF()” function is called. This function generates a random number together with an HTTP “GET” request. The request buffer is made up of a “GET” method, “TvOFF” command and a random number.
- When web server replies to the request, the “TvControlOFF()” function reads the HTTP response and changes the state picture of the tv with the one contained in the response from the server.

### 3.5.3. Camera position and light brightness control

This section of the user interface as shown in **Figure 9** presents the user with means to adjust the position of a camera as well as adjust the brightness of a light. The adjustment is accomplished through two position sliders.

The camera position slider ranges from 0 to 100 with 0 representing the leftmost position of the camera or 0° position and 100 representing the rightmost position of the camera or 100° position. These positions represent the wide range of movement possible for the camera viewing position. A camera placed on a flat wall would have 180° viewing angle. Similarly a camera

placed in a corner position would either have a 90 or 270° viewing angle. See **Figure 10** for illustration of camera position and possible viewing angles.

The light brightness position slider ranges from 0 to 100% with 0% representing the light OFF and 100% representing the light ON or bright.

When the camera position or light brightness slider thumb is moved, the “PositionBrightnessControl (DeviceName, Value)” function is called. This function generates a random number together with an HTTP “GET” request. The request buffer is made up of a “GET” method, “DeviceName” command, “Value” parameter containing the desired position or brightness depending on the DeviceName and finally a random number.

This section of the user interface does not receive any response from the web server, it updates the current slider’s value as it changes on the inner HTML document.

3.5.4. Intrusion detection

The intrusion detection section of the user interface as shown in **Figure 11** consists of a “CheckEntrance” button, the entrance names and the entrance status fields.

When the “CheckEntrance” button is clicked, the “IntrusionDection()” function is called. This function generates a random number together with an HTTP “GET” request. The request buffer is made up of a “GET” method, “Intrusion” command and a random number.

When web server replies to the request, the “IntrusionDection()” function reads the HTTP response and changes every entrance status field with the current state of the entrance contained in the response from the server.

Finally, the complete user interface is a web page that can be accessed via a web browser or using the android web-enabled mobile application created using App Inventor. The android app will be discussed in the next section.



Figure 10. Camera position and possible viewing angles.

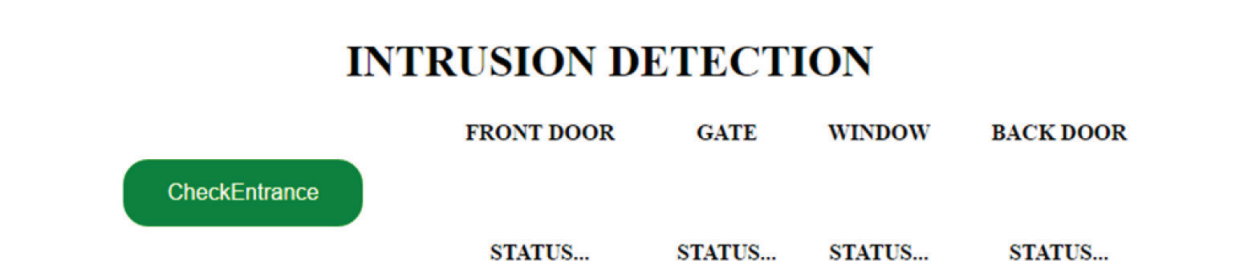


Figure 11. Security camera position and exterior light brightness control panel.



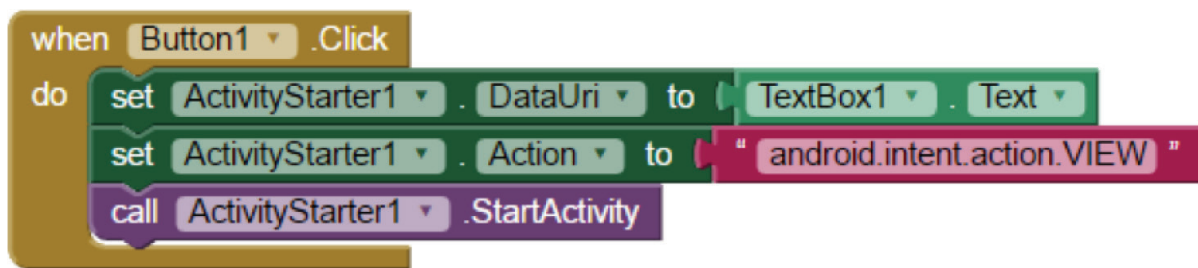


Figure 12. Designer blocks for Android app.

### 3.5.5. Android application design

To design the Android the web interface, the open-source Android application, APP Inventor, is used to create a simple android application that browses to a web user interface.

This App presents the user with an input field where the user can enter the IP address or domain of the web interface server of the home automation system controller. There is also a clickable button labeled "Click Here." Once this button is clicked, the app requests the web interface from the server.

The app was designed using the layout components, that is the vertical components who's width was set to fit the parent (fit the screen).

Text elements from the user interface were used as text input field to enter the desired web server IP address. The name describing the designer and the app itself were designed using the label component also from the user interface section.

The user clickable button component also found in the user interface section was used, its background color, size, and text were changed in the component editing section.

The web component responsible for loading the web page is found under activity starter section and it is an invisible component, meaning it is not seen in the designer phase of the app.

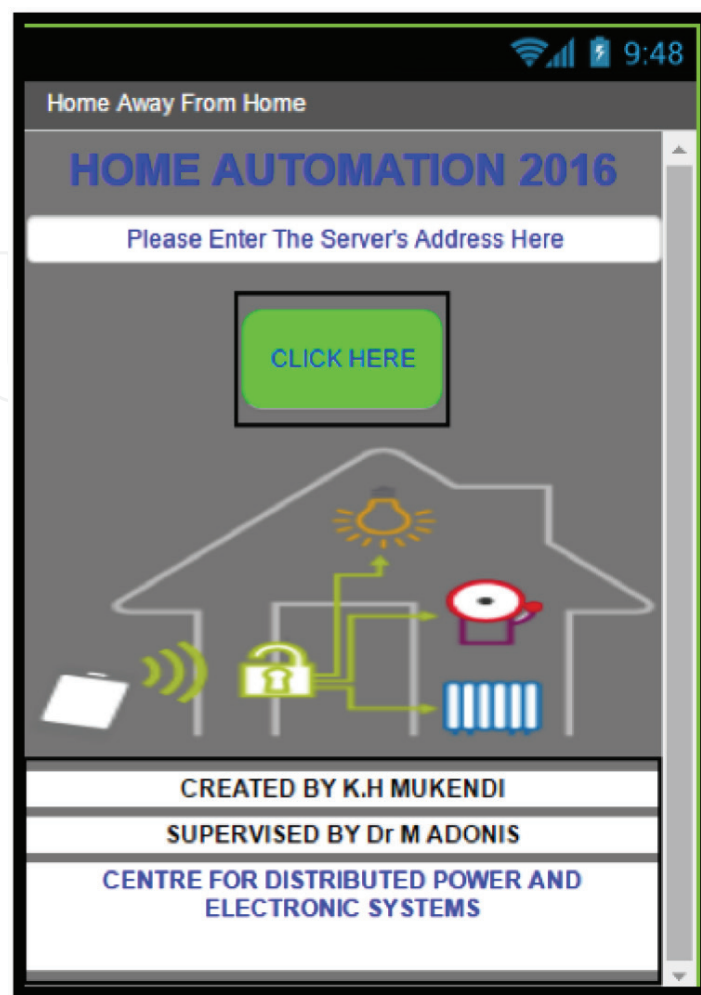
Once satisfied with the design and user interface of the app as shown in **Figure 12**, it is now time to move to the block section where blocks representing component's functionalities are grouped to create the desired function.

The user must enter the IP address assigned to the system to be controlled or its domain name, then the user must press the "CLICK HERE" button. As shown in **Figure 7** below, in this case, once the "CLICK HERE" button is clicked, an activity starter blocks are called.

First the "set ActivityStarter1.DataUri to" block is called and to it is attached the "TextBox1.Text."

Secondly the "set ActivityStarter1.Action to" block is called and to it is attached the "Text."

Finally the "call ActivityStarter1.StartActivity" block is called and the App will load the web page specified in the text field if it exists and the phone has Internet connectivity, else the app will return an error.



**Figure 13.** Android application interface panel.

### 3.5.6. Graphic user interface

The Android mobile user interface landing page for user authentication is shown in **Figure 13**. User authentication is applied to all users of the web app. The web server checks the credentials entered by the user and compares them with the information on its database, if the username and the associated password correspond to the ones in the database then access is granted to the user, else the user is requested to try again. When a user clicks a certain button, or requests that the status of a certain device is change, a corresponding function is called which generates a specific request buffer to send to the server. The server interprets the request buffer and controls the devices accordingly while updating the web client on the current state on the devices.

After successful user authentication, the user is directed to the home automation control interface. The fully completed and integrated user web interface monitoring and control panel is depicted in **Figure 14**. This interface houses all the various elements that encompass the total controller capability.

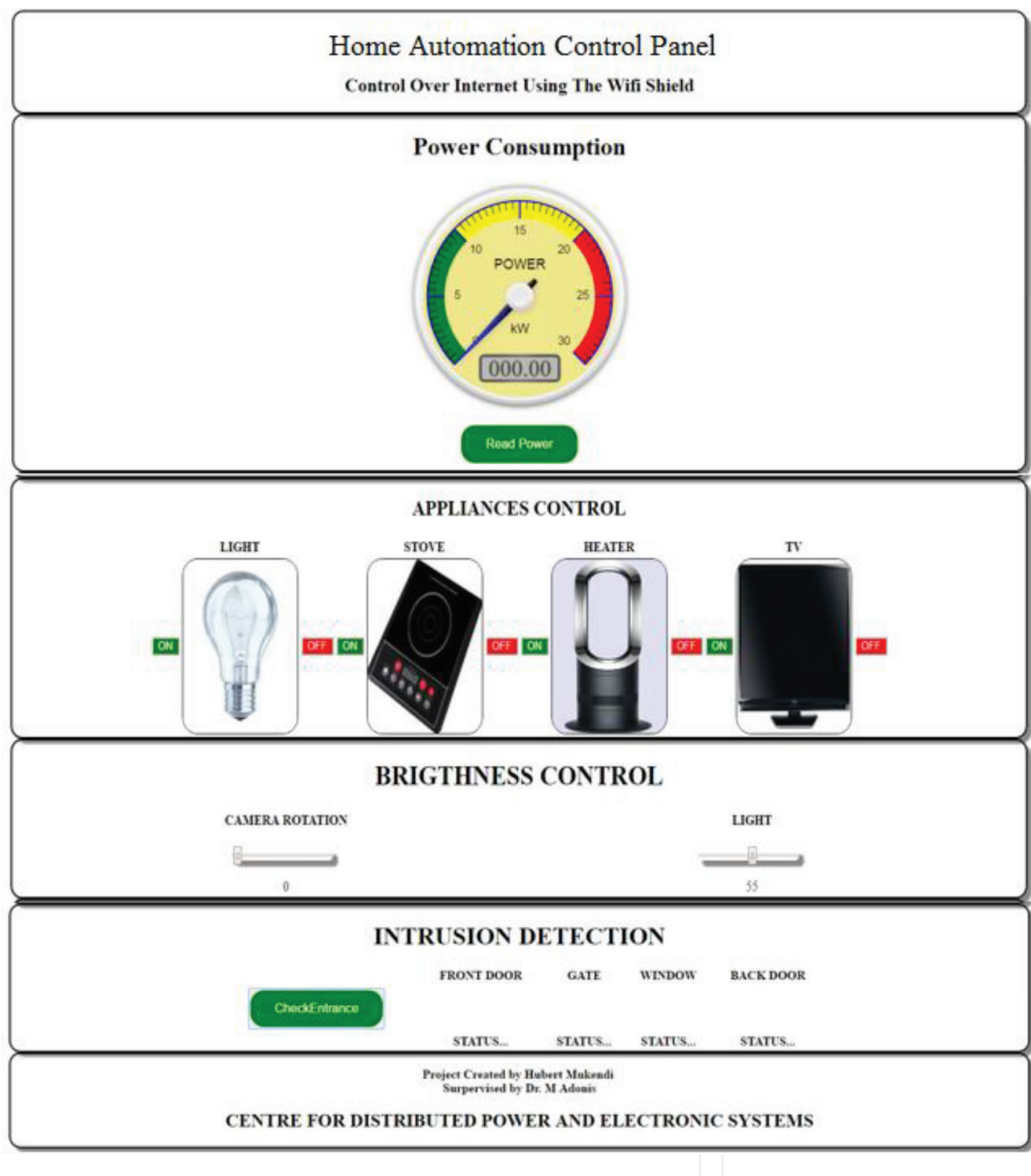


Figure 14. Complete user interface and control application.

## 4. Conclusion

Home automation represents a big shift in user data access and system control. As an assistive technology for smart cities, it allows users the autonomy and flexibility to interact with their immediate environment. The solution presented, required the use of the ubiquitous and



low cost Arduino microcontroller. This unit is a powerful and versatile embedded platform with which to structure an intervention as described here. The implementation provided here showcases a low-cost solution for home automation. It is an effective tool for all home inhabitants to exercise control and influence over their monthly electric bills. With the application of this affordable and functional user interface, a home can easily be transformed into a smart home. It is also a solution to transform any liveable space into a smart space by offering access to measured data and allowing users regulation of the electrical loads. This comprehensive home automation solution further provides lighting dimming control and access to perimeter security through features such as security camera control and intruder detection alerts.

## Author details

Hubert Kalala Mukendi\* and Marco Adonis

\*Address all correspondence to: mukendikalalahubert@gmail.com

Cape Peninsula University of Technology, Cape Town, South Africa

## References

- [1] Royston S. Smart energy technologies in everyday life: Smart Utopia? *Technology Analysis & Strategic Management*. 2014;**26**(10):1242-1247. ISSN 0953-7325 (Informa UK Limited). DOI: 10.1080/09537325.2014.975789
- [2] Karpov AA, Lale A, Ronzhin AL. Multimodal assistive systems for a smart living environment. *SPIIRAS Proceedings*. 2014;**4**(19):48. ISSN 2078-9599 (SPIIRAS). DOI: 10.15622/sp.19.3
- [3] Shahrestani S. Assistive IoT: Enhancing Human Experiences. *Internet of Things and Smart Environments*. Springer International Publishing. 2017. pp. 11-35. DOI: 10.1007/978-3-319-60164-9\_2
- [4] Vastardis N, Kampouridis M, Yang K. A user behaviour-driven smart-home gateway for energy management. *Journal of Ambient Intelligence and Smart Environments*. 2016;**8**(6):583-602. ISSN: 1876-1372 (IOS Press). DOI: 10.3233/ais-160403
- [5] Liska M, Ivanic M, Volcko V, Janiga P. Research on smart home energy management system. In: 2015 16th International Scientific Conference on Electric Power Engineering (EPE). Kouty nad Desnou, Czech Republic: IEEE; 2015. DOI: 10.1109/epe.2015.7161102
- [6] Kishore P, Veeramanikandasamy T, Sambath K, Veerakumar S. Internet of things based low-cost real-time home automation and smart security system. *International Journal of Advanced Research in Computer and Communication Engineering*. 2017;**6**(4):505-509. ISSN 2278-1021 (Tejass Publishers) DOI: 10.17148/ijarcce.2017.6497

- [7] Mao X, Li K, Zhang Z, Liang J. Design and implementation of a new smart home control system based on internet of things. In: 2017 International Smart Cities Conference (ISC2). Wuxi, China: IEEE; 2017. DOI: 10.1109/isc2.2017.8090790
- [8] Vimal N, Mayank P, Dipesh T, Sanket P, Yogita M. A review: Internet of things(IoT) based smart home automation. International Journal of Recent Trends in Engineering and Research. 2017;**3**(3):231-236. ISSN 2455-1457. DOI: 10.23883/ijrter.2017.3072.gbqb7
- [9] Malarvizhi C, Kalaipoonguzhali V, Anitha J. Microcontroller ATmega 328P and GSM based advanced home security system. International Journal of Smart Home. 2017;**11**(6): 11-20. ISSN 1975-4094 (Science and Engineering Research Support Society). DOI: 10.14257/ijsh.2017.11.6.02
- [10] Visalatchi S, Sandeep K. Automated smart metering. Smart energy metering and power theft control using Arduino & GSM. In: 2017 2nd International Conference for Convergence in Technology (I2CT). Mumbai, India: IEEE; 2017. DOI: 10.1109/i2ct.2017.8226251
- [11] Patchava V, Kandala HB, Babu PR. A smart home automation technique with raspberry Pi using IoT'. In: 2015 International Conference on Smart Sensors and Systems (IC-SSS). Bangalore, India: IEEE; 2015. DOI: 10.1109/smartsens.2015.7873584
- [12] Smita M, Ravi KB. Internet of things: Smart home automation system using raspberry Pi. International Journal of Science and Research (IJSR). 2017;**6**(1):901-905. ISSN 2319-7064. DOI: 10.21275/art20164204
- [13] Asadullah M, Ullah K. Smart home automation system using Bluetooth technology. In: 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT). Karachi, Pakistan: IEEE; 2017. DOI: 10.1109/icieect.2017.7916544
- [14] Yi X-J, Zhou M, Liu J. Design of smart home control system by Internet of Things based on ZigBee. In: 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA). Hefei, China: IEEE; 2016. DOI: 10.1109/iciea.2016.7603564
- [15] Das H, Saikia LC. GSM enabled smart energy meter and automation of home appliances. In: 2015 International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE). IEEE; 2015. DOI: 10.1109/epetsg.2015.7510071
- [16] Wenbo Y, Quanyu W, Zhenwei G. Smart home implementation based on Internet and WiFi technology. In: 2015 34th Chinese Control Conference (CCC). IEEE; 2015. DOI: 10.1109/chicc.2015.7261075

