Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 59 (2015) 34 – 44

Procedia Computer Science

International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Design and Implementation of Web Based Home Electrical Appliance Monitoring, Diagnosing, and Controlling System

Leonard Putra, Michael, Yudishtira, Bayu Kanigoro

Computer Science Program, School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

Abstract

The purpose of this research is to control home electrical appliances while also monitor the electricity consumption using web based application using laravel php framework and MySQL database. It also provides a diagnostic feedback in case of electrical appliance's failure. It aims to lower the energy usage of the electrical appliances by cutting off the power completely during standby. Many kinds of approaches like literature studies, research and experiments were conducted to develop the prototype of the system. RaspberryPi, Arduino, relay board and CT current sensors are the hardware that used to control the use of electrical devices' energy and monitor how much it consume the energy. The application is mainly built using Laravel PHP framework and JavaScript with MySQL database. The result is a system prototype that capable to control, monitor and diagnose electrical appliances power, and lowering the total energy usage of the appliances by up to 59%.

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)
Keywords: web; monitoring; diagnosing; controlling; electrical appliances.

1. Introduction

Technology brings a huge change in people's lives. Some technologies like the Internet and web services are very useful for our lives. On the top of that, mobile devices and smart phones proliferation helps user to access them. In Indonesia, the mobile devices and smart phones are mostly used for looking for some information even though it is also used for communication, visiting social media, etc. Number of Internet users in Indonesia skyrockets are increasing from 512.000 (1998) to 25.000.000 (2007) (Qomariyah & Politik, 2009). This will help advancements in other fields such as education and business.

Other than the positive benefits, technologies are also having their own drawbacks. Electrical energy wasted is one of them. Turning off electronic devices without unplugging them from the power source will waste the energy. According to the web survey targeted to casual users in 2015, about 29% of respondent still not aware that

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)
doi:10.1016/j.procs.2015.07.335

appliances are still consuming energy in standby state, and 80% of respondent still connects their appliances to the power source when not in use. A house is typically one of the places where many electronic devices are left on standby. A research shows that television uses 37.4 W for standby mode in 17 hours, while modem uses 21 W for standby mode in 21 hours (Raj, Sudhakaran, & Raj, 2009).

Based on a research done by Bina Nusantara International University, an application is developed to monitor the power usage of home electronic appliances (Geumpana, Koentjoro, & Widjaja, 2012). Two Arduino microcontrollers are used for monitoring the power consumption and sending it directly to the Windows Azure server, so it must be always connected to the internet in order to record the power consumption data. The capability is limited to monitoring power consumption only without any controlling or diagnosing capability.

A patent for home automation patented a system that controlling home devices using a universal remote control connected to a centralized control system (United States Patent No. 7092772B2, 2006). However, there are no diagnosing and monitoring capabilities, so the user never knows about the appliance power consumption at that moment.

Based on these researches, a system will be designed and implemented to control, diagnose, and monitor the power consumption. Ease of use will be critical in this system to make user more careful when using electricity. A local, dedicated web server will be used so the system can record data even if the internet connection is down.

This system will use Raspberry Pi, Arduino Uno, noninvasive AC-current sensors, and Relay Board 4 channel as an integrated system for controlling, diagnosing, and monitoring the appliances to check their efficiency in power consumption. With the rise of the internet, a web-based system using Laravel PHP Framework with MySQL database will be developed to support multiple platforms.

2. Methodology

To monitor, diagnose and control electrical appliances, a system that can read and record the appliances power consumption as well as control the current flow to the connected appliances is needed. In this system prototype, RaspberryPi is used as the microcomputer to host the web application and keep track of the power data. The web application is built on top of the Laravel PHP framework. Using Javascript, CSS, and Bootstrap, the system user interface will be easy to use, fluid, and responsive. It can be used on any platform running a modern web browser e.g. Chrome, Firefox, Safari. Mobile web browsers are also supported. In the core system of the application, such as the background data monitoring, daemons for GPIO control and sensor, data recording will be created using Javascript, Python and C++. The final system will be called “Indigo Smart Home Automation System”.

A microcontroller that can read analog data are needed to record the power data because the system will be dealing with analog data from the AC current readings. In this case, Arduino Uno and CT Current Sensor are used for obtaining the power data. Arduino Uno will act as the analog to digital converter (ADC) to provide digital data that is needed by the RaspberryPi to process. The available power monitoring library for Arduino called emonlib is used for making the process easier. This library will take care all of the calculation that need to be performed to get usable power data from the raw data of the CT current sensor. The Arduino Uno connected to the RaspberryPi via USB cable. It uses the Arduino Uno built in serial to USB interface to communicate via USB.

The diagnosing part will be done inside the RaspberryPi. The system will always monitor the power consumption, and when a certain power condition are met, the system will warns the user about the current status of the electrical appliance that is in use. With this diagnosing capability, the user will always be informed and will have more confidence about the current status of the electrical appliances that the user controls, especially if the user controls them from a remote location.

The controlling part uses a four-channel relay board, it is rated 10 A at 220 V. Because the CT current sensors are limited only to 5 A, the single appliance power consumption that can be monitored cannot be more than 5 A (1100 W at 220 V). The relay is connected directly to the RaspberryPi GPIO pins rather than the Arduino Uno digital output to minimize the cable clutter on the system because the relay board will be placed directly below the RaspberryPi.

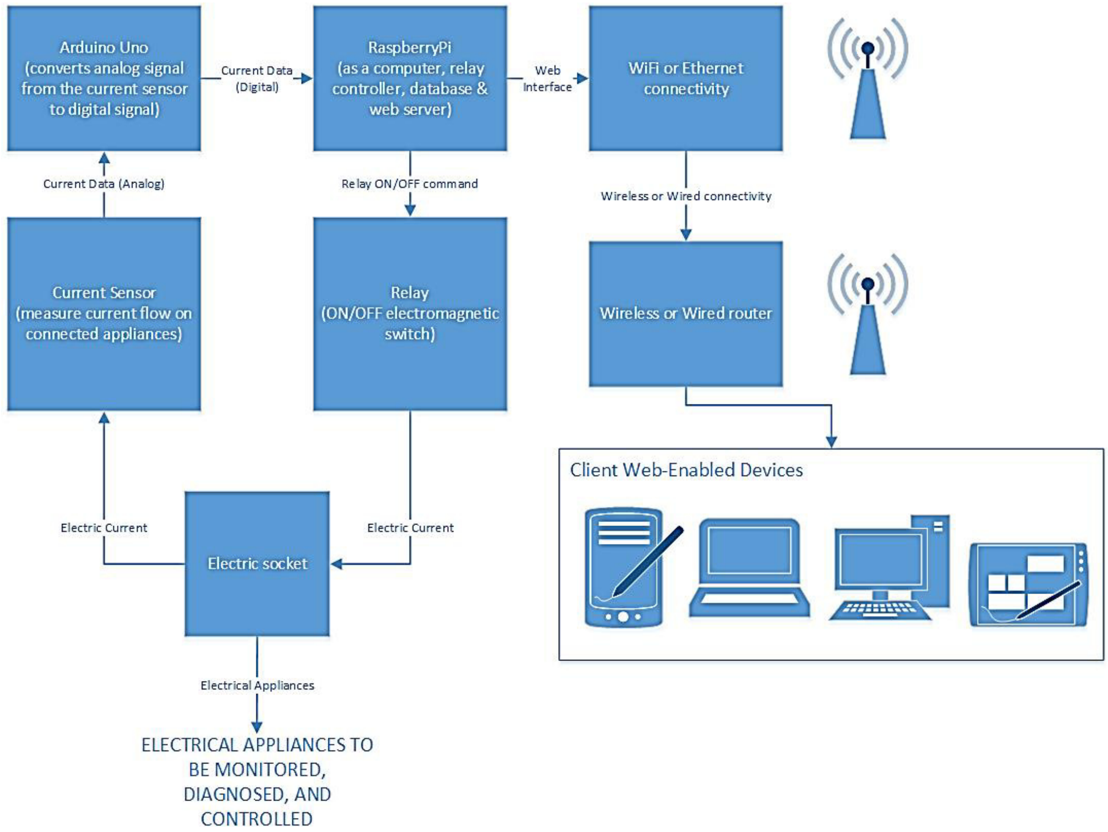

Fig. 1. System Block Diagram

Fig. 1 shows the system block diagram which is showing the connection between hardware such as Arduino microcontroller, RaspberryPi microcomputer, current sensor, router, electromagnetic relay, electric socket, and client devices. First, the current from connected electrical appliance is measured using current sensor. After that, the analog current data are sent to Arduino microcontroller to be converted become digital data. Next, the converted data are sent to RaspberryPi microcomputer. If the state is changed from on to off or vice versa, the on/off command is sent to relay board and the process will be repeated from the beginning. Finally, at the same time, the web interface will be delivered to client devices through local network or internet.

Questionnaire distribution is conducted at the beginning and the end of the development process to create more accurate results and evaluation. This questionnaire data will be used for determining the user requirements at the beginning of the development, and to measure the system overall quality at the end of the development process. Other fact finding methods such as literature studies and prototyping were also conducted.

3. System Hardware Design

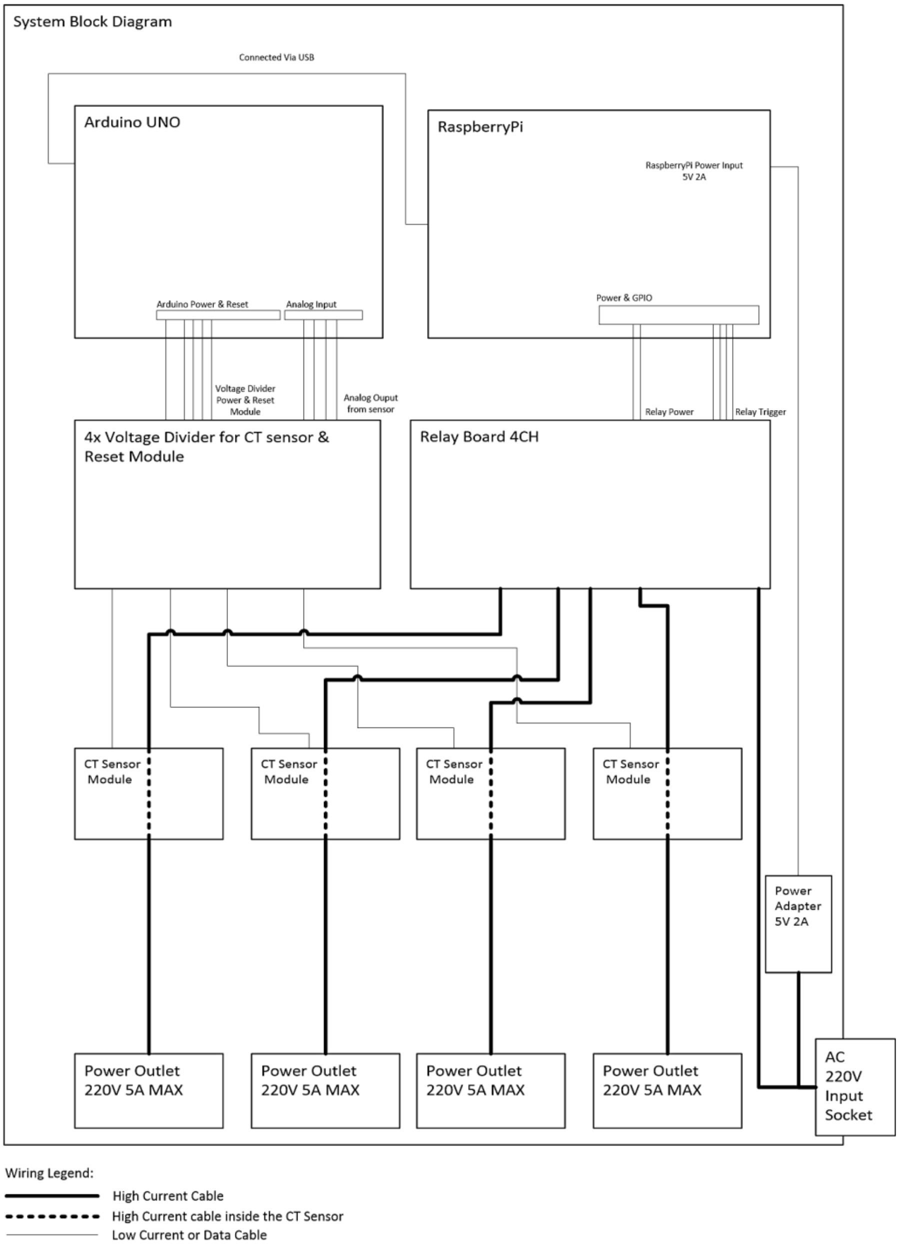

Fig. 2. System Wiring Diagram

Fig.2. shows the diagram of the system prototype. This complete system will be housed in a metal box with a door and lock to prevent the system from intrusion (For outdoor use). There are five main active part of the system, the RaspberryPi, Arduino Uno, Voltage divider for CT sensor, CT sensor and the four-channel relay board. The Arduino is connected to the RaspberryPi computer via a USB cable.

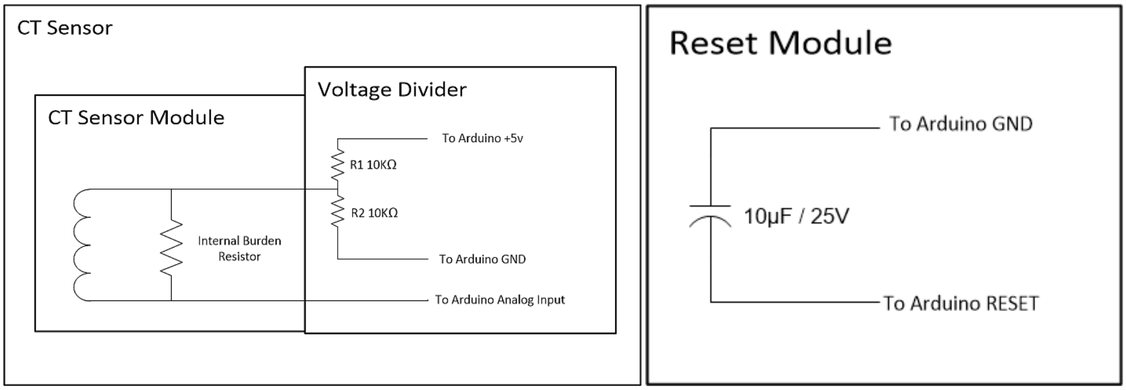

Fig. 3. CT sensor and Reset Module Circuit

Fig. 3 shows that CT sensors will firstly connected to the Voltage divider in order to normalize the AC signal that is coming from the CT sensors, as the Arduino can only read DC signal. It will create a reference voltage of about 2.5 V. After connected to the Voltage divider, the CT sensors will output its signal to the Arduino analog input pin 0-3. There are also reset modules on this Voltage divider board, the purpose of the reset module is to prevent Arduino system to reset every time a serial connection is opened. The reset module contains a capacitor that is connected to Reset and GND pin of the Arduino.

The relay board below the RaspberryPi connected with a 6 pin cable, it connects to the GPIO connector of the RaspberryPi. Two of the cables is used to carry 5 V power to energize the relay magnetic switch, the other 4 cables will carry GPIO digital signal to switch the ON/OFF state of the relay. The RaspberryPi is connected to the 10 W power adapter located on the bottom right of the system. The system will have four power outlets on the bottom, it can output maximum 5 A (1100 W) of power at 220 V. The power inlet is on the bottom right of the system, it uses a standard IEC 60320-1 C13 socket that can handle up to 10 A of current at 220 V. For system network connectivity, it uses USB WiFi dongle that is connected to the RaspberryPi USB port.

4. System Software Design

The Indigo Smart Home Automation system application design will be described on this section. First after the system is powered on, the operating system will initialize. The operating also creates a special file system called tmpfs file system and mounts it to the daemon working directory to store temporary daemon data. The tmpfs file system is used because it will write the data to the memory (RAM) instead of the flash memory (SD Card). This method will prolong the life of the SD Card as the SD Card has a limited write cycle. After the operating system has initialized properly, the StateDaemon and PowerDaemon process will be started to collect the data. The collected data will be read by the Indigo Background Process and stored into the database. The web server (Apache) will be started to provide user interface for the user via a web page.

Laravel PHP Framework is used for building the web based application. Laravel PHP Framework is using MVC (Model View Controller) as the programming development. The model will represent the database structure with the attributes of the database. The view part will handle the user interface that will be shown to the user. The controller part will act as the connector between the view and model part and also will handle the input from the user through the view part, handle all of the commands, such as to turn on or off the GPIO of the RaspberryPi, calculate power consumption, record the power used, and store it into the database through the model. The controllers communicate with the RaspberryPi by using the ‘exec’ PHP syntax in which the command will be sent to the RaspberryPi’s system shell directly. This Laravel framework is associated with AJAX and Javascript to handle the user-side interface, such as updating the current power consumption periodically. The power consumption data, which are recorded by the controller, are stored into the MySQL database by the model.

Bootstrap web template is used for building the user interface of the system. By using bootstrap, the web user interface will be responsive so that the web interface will automatically adjust the size of their content based on the size of the media used for accessing the web interface. This responsive ability makes this web interface can be viewed by the user comfortably.

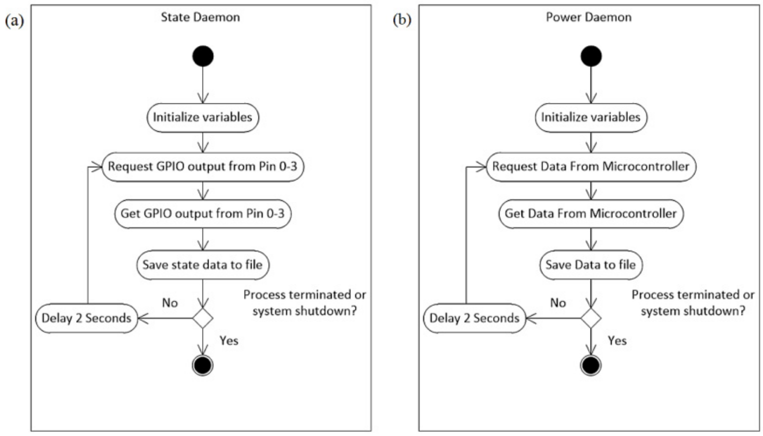

Fig. 4. (a) StateDaemon process diagram (b) Power Daemon process diagram

From the Fig. 4.(a) above, the StateDaemon process will first initialize the variables, and then requesting the GPIO output from the RaspberryPi Pin 0 to 3. The RaspberryPi will return the value for each pins in Boolean data type. The value of 1 means that the appliance is in the on state, and the value of 0 means that the appliance is in the off state.

Fig. 4.(b) explains how the PowerDaemon process work. The PowerDaemon process will first initialize its variables. After the initialization complete, the daemon will request the power consumption data from the microcontroller using serial communication. The microcontroller (Arduino) will respond to the request made earlier. All of these data from the daemons will be saved into file on the tmpfs directory, this process is repeated every two seconds until the system shuts down.

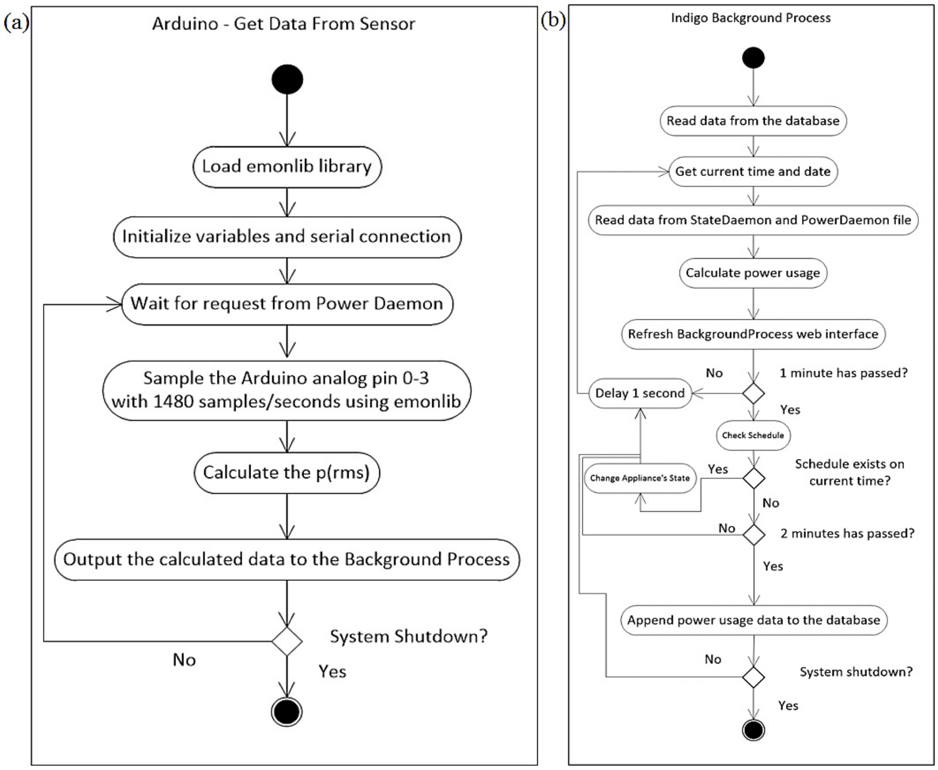

Fig. 5. (a) Arduino sensor data reading process diagram (b) Indigo Background Process diagram

Fig. 5.(a) explains about how the sensor data are calculated into a meaningful power format that can be used to measure power consumption on the appliances. Firstly, the Arduino microcontroller will load the required emonlib library to make the power calculation more straightforward and easier. After that, the variable and the serial connection via USB will be initialized with a baud rate of 115200. The microcontroller will wait for the request from the PowerDaemon process. After the request has been received, the microcontroller will sample its analog pin that is connected to the sensor (pin 0-3) at 1480 samples per second. The data will be processed with the emonlib library function called calcIrms. The output data from the function will be calculated once again to get the p(rms) value. After the p(rms) value has been calculated, the data will encoded into serial data that will be sent to the PowerDaemon. This process is repeated until the system shuts down.

Fig. 5.(b) explains about the Indigo Background Process. This process will first read the latest available data from the database. After reading the latest data, the process will get the current system time and date. This background process will gather the data from files that are produced by the StateDaemon and PowerDaemon process. The background process will perform addition operation with the power data, the background process interface will refresh every time the data are added successfully. This process is repeated every second. This process also check for every minute if there is an active schedule, if active schedule found, the system will switch the appliance state. For every two minutes, all of the calculated data will be appended to the database until the system shuts down.

5. Result and discussion

5.1 Application User Interface

The Indigo Home Automation System application is a responsive web application that will work in every modern browser, including mobile browser. Below are some example of the user interface design. Three of the most useful pages, the Admin Panel, Room Page and Scheduling will be shown.

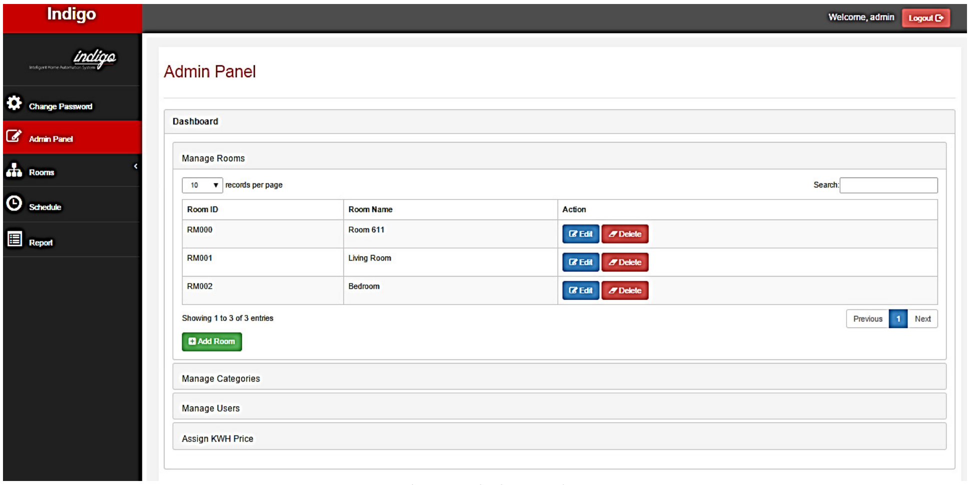

Fig. 6. Admin Panel

Admin panel page is used for managing rooms, categories, users and assigning kW·h price. This page can only be accessed by user with admin and user role. Admin and user can add, edit and delete rooms and categories in this page. Manage Rooms tab contain a list of the currently saved room and several buttons to edit, delete and add a new room. A new room can be added easily by clicking the add room button and assigning a new room name, the Room ID will be automatically generated by the system. The same process is applied to the edit room, the only difference is the edit room will change the current room name only, without altering its Room ID. The Manage Categories tab will contain a list of appliance category. Appliances can be categorized based on their type. They can also be added, edited or deleted with the same process as the Manage Rooms do. Manage Users page is a special page that can only be accessed by Admin. It contains list of registered users with several options to edit user, delete user, change user's password and activate or deactivate the user account. In addition, the user's role can also be changed with 3 selection, Admin, User and Guest. Admin can access the entire system settings and access the Manage Users tab. User can also access most of the system settings, except for the Manage Users tab. The guest can only access the Room Page, the guest can only control and monitor the current appliance, without the capability of deleting, editing or adding a new appliance. Lastly, admin and user can also assign kW·h price that will be used to calculate the usage cost on the Report Page.

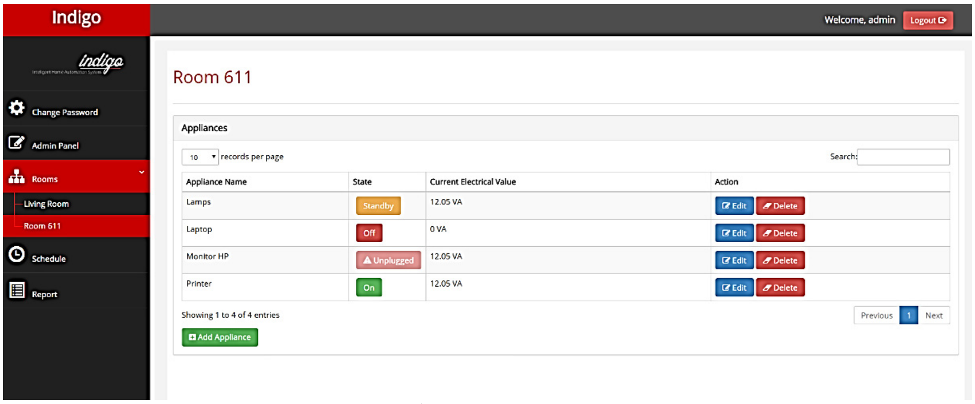

Fig. 7. Room Page

This room page will show the appliances in this room with their current state, current electrical value, and action. User can add, edit and delete appliance in this page. User also can control the state of the appliances by clicking the state button. The on state means that the appliance connected is turned on, the standby state means the appliance is off but still connected to the power source and still consuming energy, the off state means that the appliance is not connected to the power source and not consuming energy, and unplugged state means the appliance is not connected to the power source or something unusual happened with the appliance. The user with guest role cannot add, edit or delete appliance.

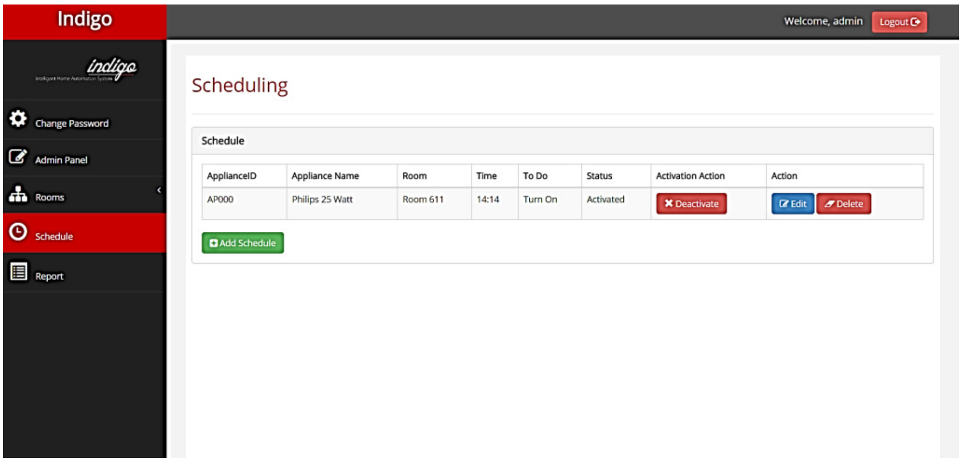

Fig. 8. Scheduling Page

In the Scheduling Page, the user can redefine which appliance that will be scheduled. The scheduling function will always repeat the schedule every day. The time of the day and the schedule action can be specified, whether it is to turn on or off the appliance at the selected time. The scheduling function can also be activated or deactivated independently for every schedule, so if a schedule wants to be disabled temporarily without deleting it, it can be done with Activation Action button.

5.2 Appliance Power Saving Evaluation

This test was conducted to show how much the reduction in power consumption that can be made possible with Indigo smart home automation system. The test appliances that is used on this test are based on the typical living room home appliances, including audio systems, TV and set top box. The test is done three times for each of the tested appliance:

- First, the appliance will be always plugged in to the standard power socket for 24 hours (Simulated by turning on the appliance in Indigo system all the time), after plugging in, the appliance will turned on for four hours with casual usage scenario. After four hours of turning on has passed, the appliance will left in standby mode for the rest of the remaining time (Twenty hours in this case). After 24 hours passed since the appliance plugged in, the data on that day will be collected.

- For the second test, the equipment will be connected to the Indigo system for 24 hours, turned on with the following settings on the Indigo:

  Scheduling: 1. Active, turn off at 00.00
  2. Active, turn on at 06.00

  With the settings above, the equipment will be automatically turned off at 00.00 and turned back on at 06.00 (Six hours in off state). This test is used to automatically eliminate standby power draw during off-peak hours without user intervention. After 24 hours passed since the appliance plugged in, the data on that day will be collected.

- For the third test, the equipment will be connected to the Indigo system for 24 hours, turned on for four hours, and then turned off in the system. After 24 hours passed since the appliance plugged in, the data on that day will be collected.

There are four devices that connected to the Indigo Home Automation System. This test is to maximize the usage of the system because the system prototype can only accommodate four appliances.

Table 1. Typical Living Room power usage

Appliances	Standard (Always plugged in) (W·h)	With Indigo Scheduling Function (W·h)	Immediately turn off with Indigo after use (W·h)	Energy Savings with Scheduling (%)	Energy Savings with immediate turn off (%)
Simbadda CST 9000 Series Speaker	493	390	138	20,89	72
Sony Home Theatre System	726	526	220	26,17	69,69
ZTE IPTV Set Top Box	106	72	21	32,07	80,18
LG Smart TV 42”	190	181	166	4,73	12,63
Total (with additional 72Wh system power consumption included)	1515	1251	617	17.42	59,27

On average, the Simbadda CST 9000 series multimedia speaker system uses about 35 W of power when playing sounds from auxiliary input at medium volume level. This appliance has no standby state, it only has an on or off state. However, the power button is located on the backside of the subwoofer unit, making it hard to reach. Therefore, the system most likely to be turned on all the time, even without any sounds playing, which consumes about 18 W of power. The power draw on this appliance when not in use is relatively high. The table above shows that we can save electricity for about 21% when using our system in scheduling mode. By manually turn off the appliance with the Indigo off switch immediately after each use, the savings can be pushed even higher to about 72%. This test proves that bad power button placement on electrical appliances can cause the appliance to waste a lot of power, even while not in use. The Indigo system can fix this problem by making the power management easily accessible for the user.

The Sony Home Theatre System on average uses about 30 – 60 W when receiving audio from FM Radio or connected as an audio amplifier for the LCD TV. The power usage pattern are very variable according to the audio output level. The appliance has a standby mode with the LCD displaying the “Sony Demo” and the time of the day. The standby power mode takes a considerable amount of power at about 25 W. According to the power usage record, with the scheduling function activated to shut down the appliance completely for six hours, Indigo can cut down the power consumption by 26%. If the appliance is immediately turned off with Indigo system after each use, it proves even better energy saving at 69%.

The ZTE IPTV Set Top Box that used on this test consumes about 5 W when turned on, receiving cable TV video broadcast via Ethernet connection. Surprisingly, the standby power is almost the same as the ON power consumption at 4 W. This equipment draws a relatively low power. However, because of the standby consumption that is almost the same as the ON power, Indigo system can saves the power usage up to 32% when using scheduling mode and up to 80% when using the Indigo turn off button after the set top box has been used.

The LG Smart TV 42” used on this test consumes about 30-50 W when turned on with 75% screen backlight power and displaying content from the ZTE IPTV Set Top Box. The standby power consumption for this TV is very low at 1 W. This 2014 made TV has an Energy Star certification label, it should uses very little power when in standby mode in order to meet the certification standard. As the table shows, with this low standby power consumption, the Indigo system cannot give a huge amount of impact on power savings. Still, the system can saves some power, albeit only by a small margin at 4.7% when using the scheduling function, and 12% when turned off completely using the turn off button on the Indigo system after finishes watching the TV for four hours.

From the tests above, the system can save the electricity as much as 80% and as low as 4.7%, it depends on the appliance standby power consumption. The appliance standby power consumption is directly proportional to the energy that can be saved using the Indigo Home Automation System. Old and outdated appliances tend to have higher standby power consumption than the newer and more advanced appliances, especially those with Energy Star or other energy saving certification.

The four devices that are tested above did not include the power consumption of the Indigo Home Automation System itself. The system consumes about 3 W all the time during the 24-hour-period measured with energy meter, even when no appliance connected. This will result in additional 72 W·h power consumption on all mode. This is because the system needs power to keep the RaspberryPi computer and the WiFi adapter running in order to receive data from the user and record power usage data. Because of this system standby power consumption, the system cannot save energy if the connected appliance standby power is lower than 3 W as the system will cause the energy usage to be higher. If the system is used in no standby mode appliances like lamps and fans, it will not save the energy at all. It will only provide the monitoring, diagnosing and controlling capability, making the user comfortable controlling their appliances remotely.

With the system power consumption included in the equation, the amount of electricity that can be saved in the typical living room using the Indigo Scheduling Function is about 17%. If the user always immediately turn off the appliances with Indigo after use, the user can save about 59% of energy usage.

5.3 User Evaluation

To do the user evaluation, questionnaire is used for gathering the experience information. Questionnaire has been done with thirteen questions related to the users’ satisfaction to the Indigo Home Automation System. These questionnaires had been shared to mostly people in IT division and get 20 responses.

All respondents learn less than ten minutes. Most respondents (95%) find it either easy or very easy to control the appliance and only 5% of the respondents said that it is very hard to do so. All respondents also said that the monitoring and reporting function is either informative or very informative. Other than that, all respondents also said that the diagnose function is either helpful or very helpful. The system is easy to navigate as no respondent has no difficulty navigating it and all respondents are satisfied with the user interface design.

However, not every respondents are satisfied with the overall system responsiveness although most of them are (90%). Most of the respondents (90%) are still believing that Indigo Home Automation System reduces vampire energy efficiently. All of the respondents still want to use the system to reduce the vampire energy if the system is already in the market because the system is easy to use (95%) and has controlling ability to set the state of the appliance (85%). On the other hand, 30% of the respondents said that the system is too expensive and 10% said that it is not so useful to reduce vampire energy. All of the respondents are still recommending Indigo Home Automation System to others and 5% of the respondents state that the Indigo Home Automation System must be released to the market.

6. Conclusion

From the results and discussion, Indigo Home Automation System is able to save the power consumption up to 59% if the user uses the controlling function on the system efficiently. Even when the user is not always interacting with the system, with the scheduling function configured properly, the system can automatically turn off unused devices during off-peak hours and manage to save about 17% of energy. In addition, Indigo Home Automation System successfully makes controlling, monitoring and diagnosing easy for its user with the intuitive web user interface based on the user analysis.

References

1. Christopher Murray, C. S. United States Patent No. 7092772B2; 2006
2. Geumpana, T. A., Koentjoro, J., & Widjaja, N. D. Developing Cloud Energy Consumption Monitoring System for Home Electronic Appliance: Indonesian Case; 2012
3. OpenEnergyMonitor. Installing Arduino Libraries | OpenEnergyMonitor. Retrieved from http://openenergymonitor.org/emon/buildingblocks/installing-arduino-libraries; 2013
4. Qomariyah, N., & Politik, U. Perilaku Penggunaan Internet pada Kalangan Remaja di Perkotaan; 2009
5. Raj, P. A., Sudhakaran, M., & Raj, P. P. Estimation of Standby Power Consumption for Typical Appliances. Journal of Engineering Science & Technology Review, 2009; 2(1)