Implementation of an automatic appliance control and energy monitoring system

Rula Sharqi, Muhammad Hassaan Ayyub

Abstract— The concept of home automation has recently sparked consumer interest due to rapid innovation, development of new technologies, growing trend towards sustainability and efficiency, to serve this issue, a ‘stepping-stone’ needs to be developed, not necessarily providing a comprehensive automation solution, but instead aiming to ease the transition to such a solution by minimizing the investment required by the consumer, along with installation and maintenance overheads. In this paper, such a product is developed by improving a low-cost device that already exists in most homes and has the potential to be useful for automation purposes – the power strip. The developed product is a power strip that allows users to remotely control and automate appliances, and track the power consumption of each appliance using a web interface connected to the Internet. The developed system serves as a simple and low-cost entry into the world of automation and will aid in presenting this concept to consumers who are not ready to invest the substantial amount required by existing solutions.

Keywords— home automation; wireless control; energy consumption; power strip; smart homes

I. INTRODUCTION

The aim of this paper is to develop a power strip (an electrical power distribution system for home appliances) that is capable of being controlled and monitored remotely using a web interface. Through the interface, users could manually toggle appliances on or off, or automate them by setting timers and date and time schedules. The power strip also measured the power consumption of each appliance connected to it, and displayed this information as a graph in the web interface [5, 6]. The power strip was implemented using a Raspberry Pi 2 microcontroller, an Arduino Uno microcontroller, a relay module, and current and voltage measurement circuits. An Apache web server was installed on the Raspberry Pi that hosted the web interface. Any actions the user performed in the interface were interpreted by the web server and sent to a Python program. The Python program evaluated the actions and sent control signals to the relay module to switch appliances on or off via the GPIO pins of the Raspberry Pi.

The Arduino microcontroller functioned as a data acquisition system, taking measurements from the current and voltage measurement circuits, and sending them to the Raspberry Pi via a USB interface. The Python program in the Raspberry Pi calculated active power consumption and other electrical quantities from these measurements and stored them in a database. The web server then accessed this information from the database and displayed it in the user interface upon request [1].

The voltage measurement circuit was implemented as a mains transformer connected to a signal conditioning circuit, which scaled and offset the signal in order to be input into the Arduino microcontroller’s analogue input pins. The current drawn by each appliance was measured using current transformers which were also connected to signal conditioning circuits to output the signal to the Arduino microcontroller. The Arduino sent this data to the Raspberry Pi which generated waveforms of voltage and current and integrated their product over the sampling time to obtain the active power consumption [2].

Tests were carried out to obtain the primary-secondary voltage ratio of the mains transformer, and the output voltage characteristic of the current transformers with respect to input current. Thereafter, the scaling and offsetting components of the signal conditioning circuits were tested in order to determine the scaling factors required to construct the voltage and current waveforms in the Raspberry Pi. Finally, the overall voltage and current measurements taken by the Arduino were compared with measurements from multimeters and figures for the accuracy of the power consumption measurement were obtained. The average error in the power consumption measurement was found to be less than 1% for load currents greater than 1A, and approximately 10% for currents in the 100mA range [3] [7].

II. DESIGN AND IMPLEMENTATION

A user specification was developed based on the tasks required to be accomplished by the system, and the system was developed based on that. The main tasks of the system were controlling appliances, both manually and automatically, and monitoring the power consumption of these appliances. Therefore, two modules were required that performed these tasks. In addition, a user interface was required so that the user could interact with the power strip, and a processing module was required that would link the user interface with the controlling and monitoring modules, and perform any

The authors are with the School of engineering and physical sciences, Heriot-Watt University – Dubai Campus, Dubai, United Arab Emirates. r.sharqi@hw.ac.uk, ma1004@hw.ac.uk

calculations required. A data storage system was also required to support the processing module. The overall architecture of the developed system is shown in Fig. 1.

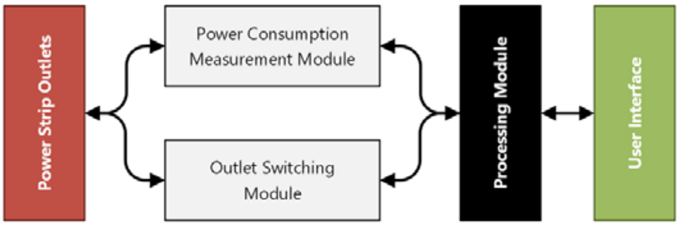


Fig. 1. Overall architecture of developed system

A. *Electrical Outlets*

These were electrical sockets to which consumer appliances could be connected. Each socket was connected to power strip electrical plug using three wires: Live, Neutral and Earth. The outlet switching modules and power consumption measurement modules were directly connected to each of these outlets in order to control and monitor them respectively.

B. *User Interface*

This was the front-end of the power strip. It allowed the user to control both the power strip outlets and monitor the power consumption of the outlets. Overall, it presented controls for the following:

☐ Toggling power strip outlets individually.

☐ Setting timers for each outlet.

☐ Setting custom date and time schedules for each outlet

☐ Viewing power consumption history for each outlet.

The user interface was implemented as a website running on an Apache 2 web server in the processing module. Table 1 shows the different pages of the user interface along with their functions, and Fig. 2 (a, b, and c) shows the layout of the interface pages.

Table 1. Pages of the user interface along with their functions.

Page	Purpose
Homepage	• Viewing the current status of all the outlets of the power strip. • Toggling any outlet into on or off state by pressing a button. • Navigating to each outlet's timers, routines or power consumption page.
Timers Page	• Setting a timer for the selected outlet o Setting a duration of the timer. o Setting the required state for the outlet throughout the timer.
Routines Page	• Creating a date and time schedule for the selected outlet. o Setting a start date and time for the schedule o Setting an end date and time for the schedule o Setting the required state for the outlet throughout the schedule
Power Consumption Page	• Viewing power consumption information o Mains voltage. o Current drawn by the appliance o Active power, reactive power, apparent power. o Power factor. o Graph of active power consumption history.

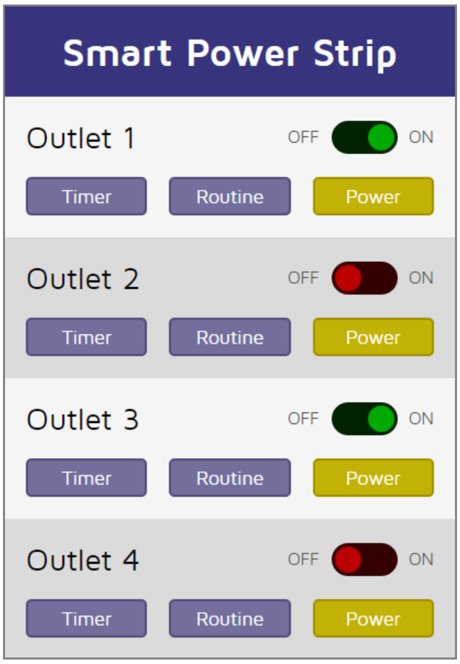


Fig.2 (a) User interface homepage.

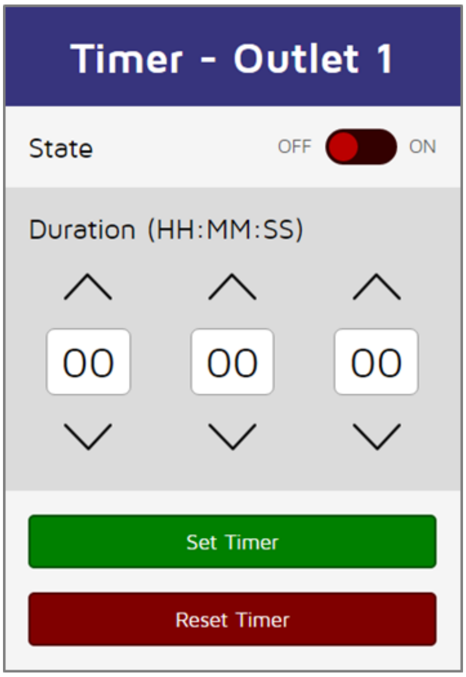
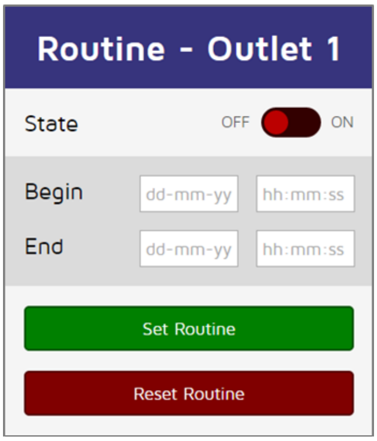


Fig.2(b) User interface timer page.



Routine - Outlet 1

State OFF ON

Begin dd-mm-yy hh:mm:ss

End dd-mm-yy hh:mm:ss

Fig.2 (c) User interface routine page

III. OUTLET SWITCHING MODULE

The function of this module was to switch the electrical outlets on or off based on signals received from the processing module according to user selection in the interface. This module was implemented as a 4-channel relay module which received control signals from the processing module and switched the electrical outlets accordingly.

IV. POWER CONSUMPTION MEASUREMENT MODULE

The function of this module was to acquire voltage and current measurements from all outlets of the power strip and send these measurements to the processing module to calculate power consumption data. The module consisted of five sensors: four current sensors to measure the current drawn by each outlet, and one voltage sensor to measure the mains voltage supplied to the outlets. The current sensors were implemented as current transformers with burden resistors. The voltage sensor consisted of a transformer along with a voltage divider. Both types of sensors were connected to signal conditioning circuits before being input to a data acquisition module.

The data acquisition module consisted of an Arduino Uno microcontroller with an internal analogue-to-digital converter. The Arduino Uno read the measurements from all five sensors and transferred the data to the processing module for power consumption to be calculated [4].

V. PROCESSING MODULE

This module was the link between the user interface and the other modules of the power strip. It was also where power consumption calculations and outlet timer and routine management occurred. Overall, the following functions were performed by this module:

- Reading user input from the user interface, deciphering it and performing the appropriate actions, which could be displaying data in the interface, or sending control signals to outlet switching module.
- Calculating power consumption information from measurements sent by the power consumption measurement module and storing the calculated information in the data storage system.

This module was implemented as a Raspberry Pi 2 microcontroller running the Raspbian Jessie operating system. The different functions were implemented in the Raspberry Pi as Python programs running on the microcontroller. An Apache web server was also configured to host the user interface website. The Raspberry Pi communicated over a USB interface with the data acquisition module in the power consumption measurement module, and via GPIO pins with the relays in the outlet switching module.

VI. SYSTEM-LEVEL OVERVIEW

Fig. 3 shows a detailed block diagram of the system. The user interface sent commands to the processing module according to user input, and also received power consumption data from it to be displayed to the user. The processing module, supported by the data storage system parsed the commands from the user interface and sent appropriate control signals to the outlet switching module. It also queried the power consumption module and calculated power consumption information, which it displayed in the user interface. The outlet switching module and the power consumption measurement module both accessed the electrical outlets directly, controlling and monitoring them respectively [7].

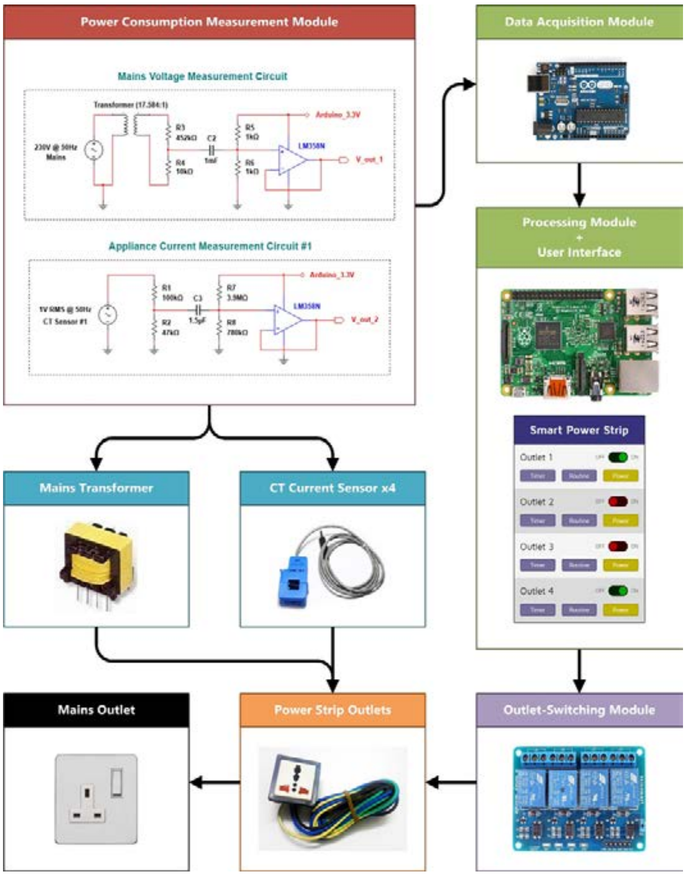


Fig. 3. Block diagram of the system.

VII. RESULTS AND ANALYSIS

It was concluded from testing (refer to Table 2) that the average error in the power consumption measurement for a load current of greater than 0.5A was of an acceptable level – only rising above 1% for one case. The maximum error in this range was also reasonable, only rising above 2.5% for one case.

However, at the 100mA range the error was too high and the results were not consistent. This was due to the fact that at 100mA, the output voltage from the signal conditioning circuit was only 10mV, and an error of even a single ADC step in the Arduino created an error of around 10% in the measured value. The average error in that range was 9.4% – a single step of the ADC, whereas the average maximum error was 17.2% – around two steps of the ADC. When evaluated from the ADC's perspective, the error is very reasonable. Using an ADC with greater resolution will allow for better evaluation.

Table 2. Error in voltage, current and power consumption at different load currents at mains voltage of 230V.

Current (A)	Voltage Measurement Error (%)	Current Measurement Error (%)	Error in Power Consumption (%)
0.101	0.28	9.42	9.7
0.499		0.66	2.9
0.734		0.76	1.0
1.001		0.83	1.1
4.00		0.46	0.7

VIII.3. CONCLUSION

The power strip was developed successfully, and all of the controlling and monitoring functionality operated as expected. The accuracy of the power consumption measurement was excellent at higher currents, with an average error of less than 1% above 1A of load current. Although the error increased to around 10% at currents in the range of 100mA, it can still be considered to be reasonable taking into account the fact that the power strip would be used in homes and the measurement would still give a good enough estimate of the actual power consumed for this application.

References

[1] Shajahan and A. Anand, "Data acquisition and control using Arduino-Android platform: Smart plug", 2013 International Conference on Energy Efficient Technologies for Sustainability, 2013.

[2] Product datasheet, "Raspberry Pi 2 Model B", Raspberry Pi, 2017. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/ [Last Accessed: 20- Jun- 2017].

[3] Product datasheet "Arduino - ArduinoBoardUno", Arduino.cc, 2017. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno [Last Accessed: 20- Jun- 2017].

[4] Shajahan and A. Anand, "Data acquisition and control using Arduino-Android platform: Smart plug", 2013 International Conference on Energy Efficient Technologies for Sustainability, 2013.

[5] R. Rane, S. Patel, R. Phutane and S. Halhalli, "New Android Application for Smart Home System",International Journal of Advance Foundation and Research in Computer (IJAFRC), vol. 2, no. 2015, pp. 499-504, 2015.

[6] K. Baraka, M. Ghobril, S. Malek, R. Kanj and A. Kayssi, "Low Cost Arduino/Android-Based Energy-Efficient Home Automation System with Smart Task Scheduling", 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks, 2013.

[7] DEWA "DEWA launched Power Factor campaign targeting Commercial and Industrial sectors" Dubai electricity and water authority newsletter, 2016. [Online]. Available: https://e-services.dewa.gov.ae/newshist/eventsdetails.aspx?id=02022273000000000000000002022273 . [Last Accessed: 31- Mar-2016].