

Mark Nikora

Wintec

Mark.Nikora@wintec.ac.nz

This paper describes the development of a system for taking measurements of electrical power consumption in a domestic situation and using a web service to store the measurements in a cloud hosted database. This is part of a larger project to evaluate the potential for using power monitoring as a proxy for wellness as part of the assisted living projects at Wintec.

1. INTRODUCTION

The authors of this paper propose an alternative system of power monitoring and data storage, as a possible means of monitoring the activities of daily living of a person living alone, in a very unobtrusive way. The system described has recently become feasible for use in New Zealand due to advances in the availability of home automation products for the Australia/NZ market.

2. POWER MONITORING

Three Aeotec Smart Switches are used to monitor power outlets which provide power to the kettle, television and washing machine.

The meters all use the Z-Wave radio protocol to send power measurements to the controller.

3. Z-WAVE

Wireless (radio) communication was the preferred method for the system to transfer data from the power monitors to the controller, avoiding the installation of cables around the house.

The Z-Wave radio protocol (Z-WaveAlliance, 2013) was chosen for its interoperability with a wide range of home automation sensors from several manufacturers, and the availability of a controller board compatible with the Raspberry Pi miniature computer.

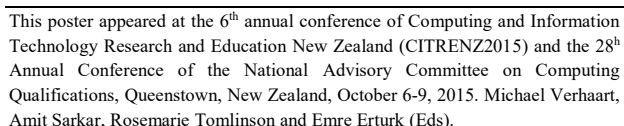
It is envisaged that the system could be extended to include other types of Z-Wave compatible sensors, including infrared movement sensors.

4. THE CONTROLLER

A Raspberry Pi miniature single board computer (Raspberry Pi Foundation, 2015) is used as the controller. It is fitted with a Razberry Z-Wave transceiver daughter board (Razberry, 2015). The use of a non-proprietary controller allows for the flexibility to communicate with a web service provided by the authors.

The Raspberry Pi runs a variant of the Linux operating system. The software supplied to work with the Razberry board, known as “Z-Way”, is implemented as a web server. This makes it easy to use a web browser to monitor and control Z-Wave devices from the Raspberry Pi or from anywhere on the local network.

The authors have developed a web client application which runs on the Raspberry Pi and asks the Z-Way server to query each of the power monitors every 15 seconds. The power measurements returned from the sensors are aggregated and then forwarded once per minute to a web service. The data is packaged as a JSON string and the HTTP POST method is used to transfer the data (ECMA International, 2013). The



application was written in C# and runs on the Mono framework installed on the Raspberry Pi (raspberrypi-sharp.org, 2015).

5. THE WEB SERVICE

A REST web service (Fielding, 2000) that runs on a virtual Windows Server™ in the Microsoft Azure™ cloud was developed. It follows the Model-View-Controller pattern, implemented with C#, to create a Web Application which runs on a Microsoft IIS™ web server (Microsoft, 2015). The data is stored in a table on an instance of Microsoft SQL Server™, also running on the same virtual server.

PowerReading objects hold the values for each power measurement. The properties of a powerReading object are Location, which is a code which identifies the home being monitored, Time, which is the date and time that the measurements were taken, and TotalPower, Socket1, Socket2 and Socket3 which are power measurements. An example of a JSON formatted string which represents these objects is "[{"Time":1425812406,"TotalPower":782.092,"Socket1":0.0,"Socket2":1.222,"Socket3":2.0,"Location":1597},{\"Time\":1425812421,\"TotalPower\":784.999,\"Socket1\":0.0,\"Socket2\":1.195,\"Socket3\":1.907,\"Location\":1597},...]\".

The HTTP POST method is used to send the JSON string to the web service.

An example of a URL to retrieve data to the web service is <http://assistedfreedom.co.nz:83/powermonitor/api/powerreadings?location=1597&startTime=1425812400&endTime=1425898800> where startTime and endTime represent the time period, in Unix time, over which the data is to be retrieved.

6. FAULT TOLERANCE

Fault tolerance is important because each part of the system has to work correctly for the power measurement data to be successfully transferred to the database and stored. Power supply and network failures could impact on the reliability of the system.

Several processes were set up to mitigate failure at various points in the system. If the connection to the internet or the web service fails, the power measurements will be stored by the Raspberry Pi, and sent later when the web service becomes available again.

A separate “Connection Checker” application running on the Windows Server gets the latest power measurements stored in the database every 60 seconds. If the stored values have a time value which is more than 60 seconds in the past, which is the interval that the Raspberry Pi normally sends the data, an email is sent to the authors, stating when the service failed. In addition, a daily email is sent by the application at 9.00am every morning informing the authors of the state of the service. Failure to receive an email has occurred only once over several months of operation, and necessitated a reboot of the virtual server.

In the course of development the Z-Wave software (controlling the Raspberry daughter board) periodically froze, resulting in identical power measurements being sent from the Raspberry Pi. The Connection Checker application monitors the presence of duplicate successive power values in the database. If three of these are detected a soft reset message is sent in the

acknowledgement to the next set of power values from the Raspberry Pi. When the soft reset message is received by the Raspberry Pi, it resets the Z-Wave software process. The Connection Checker also sends an email message to the authors informing them of the event.

7. CONCLUSION

The authors have shown that it is possible to build a reliable, relatively high resolution domestic power monitoring system using available home automation products and a Cloud Service. The authors used the C# programming language throughout the project. The cost of the hardware components is about \$600, with half of that used for the monitoring of individual power outlets, which may not be needed in an assisted living scenario.

The data collected is accessible via the internet from any location, and potentially from any platform, removing the need for some sort of display panel. Further analysis of the data, for the purpose of disaggregation – allowing individual appliances and lights to be recognised, would be possible by developing an additional application which could run on the cloud server.

If the system were to be used for assisted living, there is scope for further research into the automatic recognition of an abnormal pattern of power usage, indicating possible ill health of the person being monitored, using a machine learning algorithm.

8. REFERENCES

- Aeon Labs LLC. (2015, March 26). *Home Automation Products*. Retrieved from Aeotec: <http://aeotec.com/homeautomation>
- ECMA International. (2013, October). *ECMA-404 Standard The JSON Data Interchange Format*. Retrieved March 16, 2015, from ECMA International: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>
- Fielding, R. T. (2000). *Fielding Dissertation: Chapter 5: Representational State Transfer (REST)*. Retrieved March 16, 2015, from [www.ics.uci.edu: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)
- Hunt, T. D. (2013). *Utilising home electricity usage as a low intrusive wellbeing monitor*. Hamilton, NZL.
- Microsoft. (2015, April 22). *ASP.Net MVC*. Retrieved March 18, 2015, from ASP.NET: www.asp.net/mvc
- Raspberry Pi Foundation. (2015, March 30). *Raspberry Pi*. Retrieved March 16, 2015, from [RaspberryPi.org: http://www.raspberrypi.org/](http://www.raspberrypi.org/)
- raspberrypi-sharp.org. (2015, April 22). *Raspberry#*. Retrieved March 16, 2015, from [raspberrypi.org: http://www.raspberrypi-sharp.org/](http://www.raspberrypi-sharp.org/)
- Razberry. (2015, March 16). Retrieved March 16, 2015, from [razberry.z-wave.me: http://razberry.z-wave.me/](http://razberry.z-wave.me/)
- Z-Wave.Me. (2015). *Razberry Project*. Retrieved March 16, 2015, from Razberry Project: <http://razberry.z-wave.me/>
- Z-WaveAlliance. (2013). *About Z-Wave*. Retrieved November 6, 2013, from [www.z-wavealliance.org: http://www.z-wavealliance.org/technology](http://www.z-wavealliance.org/technology)