Power Consumption and Generation Logging
3rd Year Project
ELEC3330

John Pearce
10504036
Supervisor: Associate Professor Thomas Bräunl

26/05/09

Abstract

An energy monitoring system combining data from the REV Project and IDEAL House at the University of Western Australia. The system logs the amount of energy consumed by the REV Project in keeping an electric vehicle charged and the amount of energy generated by solar panels atop the IDEAL House. The intention is that the energy generated by the IDEAL House will be used to offset the energy used by the REV Project. The results of the net energy gained or lost, graphs, and summary display will be present on the REV Project's website.

Available directly at `http://www.therevproject.com/powerlogging/`.

Contents

List of Figures

Nomenclature

EC3000	Voltcraft Energy Control 3000
IDEAL House	Intelligently Designed Engineering for Advanced Living House
Java	A programming language developed by Sun Microsystems
MySQL	Popular Open Source Database
NSSM	Non-Sucking Service Manager
PHP	PHP Hypertext Processor
PNG	Portable Network Graphic
REV	Renewable Energy Vehicle
RS232	A serial communications electrical standard
SNAP	Student Network Access Project
SQLite	A light-weight embedded database
USB	Universal Serial Bus
VPN	Virtual Private Network
WLAN	Wireless Local Area Network

Part I
Introduction

1 Background

The REV Project at the University of Western Australia is an ongoing set of student projects focused on creating a vehicle running entirely on renewable energy. Currently the vehicle is charged through a standard grid-connected 240V wall outlet. As of late 2008, solar panels have been installed on the roof of the IDEAL House, another UWA School of EE&CE project. The solar panels do not store power in batteries for the IDEAL House but instead feed the power generated back into the electricity grid. In 2008 student Carl Beyer undertook a project to monitor the energy usage of the electric vehicles and to offset this with the monitored energy generated by the IDEAL House solar panels. These results would then be presented on the REV Project's website. Due to timing issues the solar panel energy monitoring was not completed. However both the REV and IDEAL House solar panels are both equipped with the hardware to log power usage and generation.

2 Project Motivations and Objectives

The objective of this project is to produce a dynamic webpage to integrate into the current REV Project website which will present the energy usage of the REV and energy generated by the IDEAL House solar panels. Allowing the viewer to see how much of the REVs power is offset by the power generated by the solar panels. The power data will be automaically collected from both the REV Project and IDEAL House and forwarded to the website at regular intervals.

2.1 Project Scope

The aims of this project were

1. To evaluate the already existing source code and system.
2. To make any required changes to the already existing system.
3. To design and implement a power logging system for the IDEAL House.
4. To present the energy generation and consumption results on a webpage.

Part II
Evaluation

As previous students have worked on relevant projects in both the REV Project and IDEAL House an evaluation of previous contributions was carried out.

3 REV Project

3.1 Hardware

A Voltcraft Energy Control 3000 was already connected to the charger of the REV. The EC3000 consists of two components

- A wireless power sensor
- USB logging receiver

The EC3000 sends, via USB to a computer in the REV lab, the power being used in 5 minute intervals.

3.2 Software

The EC3000 comes packaged EnergyProfessional, a program able to read the data from the device. While EnergyProfessional is running it constantly reads data from the EC3000 and stores it in a Microsoft Access database on the local computer.

A service helper, NSSM, is used to run the EnergyProfessional program as a Windows service. This ensures that the program will run in the background and automatically restart itself if ever needed.

An Apache webserver was used on the REV Lab computer to host the PHP scripts which generated the webpage.

3.2.1 PHP Webpage

The already existing webpage consisted of 3 major components

- A scheduled script to process the data and generate the graphs : `gengraph.php`
- A flat-file to store energy totals : `data.txt`
- A script a display the webpage : `display.php`

The script `gengraph.php` is scheduled to run daily to read the EnergyProfessional database, calculate various sums of the energy used over different periods and generate graphs for that day. The graphs are generated using PHP library *JpGraph* and the energy totals are stored in `data.txt`. The script `display.php` is executed each time a client loads it using a web-browser. This script opens th file `data.txt` and displays the summary energy results along with the previously generated graphs.

The advantage of this method is that all of the resource intensive processing is done only once per day and so the webpage loads quickly. However as this processing only stores summary historical energy results accessing the data for previous dates is not possible. Another disadvantage is that no data is displayed for a date until the end of the day when processing occurs.

4 IDEAL House

4.1 Hardware

As of late 2008 solar panels were installed on the roof of the IDEAL House. These panels are connected through a *Fronius IG30* inverter to the power grid. The inverter has a power logging card installed with an RS232 connection. However no cable had been run to connect to it. The power logging card stores the power being generated at 15 minute intervals.

The IDEAL House also contains a Linux and Microsoft Windows 2003 servers. These are housed in a seperate room to the inverter. Both computers have WLAN built-in and use external antennas. With the antennas placed correctly both computers are able to connect to SNAP wireless, however with a low signal strength.

4.2 Software

Fronius *Solar.access* comes packaged with the inverter. This program is used to setup and administer the inverter as well as to read the data from the power logging card. Each time the data is read from the power logging card it is stored in a Microsoft SQL Server 2005 database. The program can be configured to perform an automatic download according to a schedule.

Part III
Implementation

5 System Overview

The system can be broken down into 3 main catagories

- REV Project
- IDEAL House
- Web Server

The data is collected from the two sources and submitted to the Web Server for storage, processing, and display.

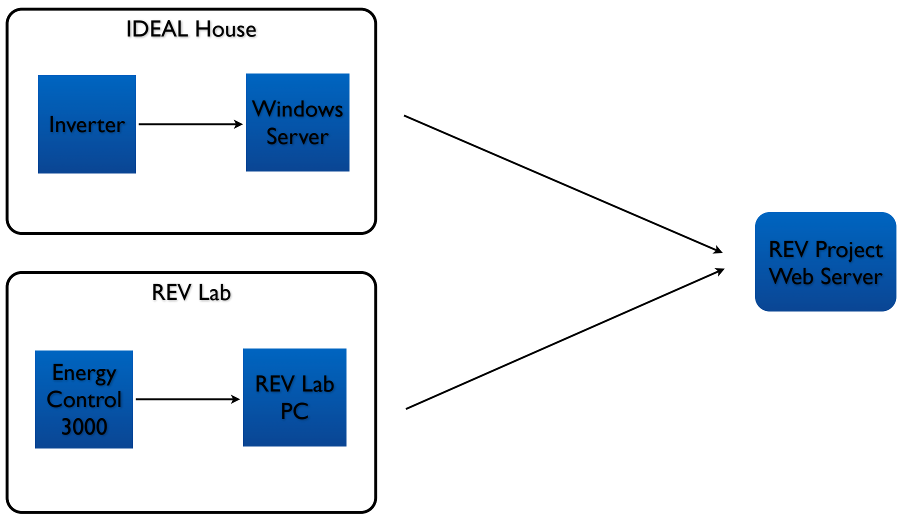


Figure 1: System Overview

6 Collecting the Data

6.1 REV Project

A Microsoft Windows XP computer in the REV Lab is connected to the EC3000 via USB. EnergyProfessional is always running in the background as a Windows service. While EnergyProfessional is running it logs the current power being used to a Microsoft Access database stored on the local computer.

The EC3000 sends the current power being used to the computer every 5 minutes. A value of 0W is sent if no power is being used, or if there is an error communicating with the server.

Every hour the PowerUploader service checks the Microsoft Access database for any new records since the last sucsessfuly sumbitted record. Any new records and then submitted to the PHP Form hosted offsite.

6.1.1 Maximum Energy

The EC3000 occasionally returns erroneous data in the form of extremely large power consumption figures. This may be caused by a wireless communications failure or potentially the behaviour in pluging and unplugging the sensor.

A maximum energy value allowable to be submitted to the PHP Form can be set in the Java configuration file. A default value of 2400W is used as this corresponds to a common 10A circuit.

Listing 1: Configuration for a maximum energy of 2400W

```
#Maximum power value to upload
#10A * 240V = 2400W, anything bigger than this is an error
maxenergy=2400
```

6.2 IDEAL House

The Microsoft Windows 2003 server in the IDEAL house has been connected via RS232 to the data logging card in the inverter. This allows the Fronius Solar.access software to communicate with the inverter. Solar.access has been configured to automatically download data from the data logger every hour and store this data in a local Microsoft SQL Server 2005 Express database.

The data logging card in the inverter stores the power being generated at 15 minute intervals. When no power is being generated the inverter is switched off and the data logging card records no data.

Every hour the PowerUploader service checks the SQL Server 2005 database for any new records since the last sucsessfuly sumbitted record. Any new records and then submitted to the PHP Form hosted offsite.

6.2.1 SNAP Access

As the data from the inverter must be submitted to a host outside of the UWA network a SNAP account was created. Currently this account is restricted to internal UWA traffic only. Once it is changed to an unrestricted account data will begin to be uploaded offsite. As there are concerns of excess traffic being used by other services on the computer the firewall GhostWall has been installed and configured to block outgoing access to anything expect the UWA internal network and therevproject.com.

```
SNAP User:  ideal
SNAP Password:  fuN7imes
```

6.2.2 RS232 Cable

To connect the inverter's data logging card to the Windows 2003 server an ~15m long RS232 cable was installed by the Electronics Workshop. This cable runs from the inverter in the left room of the IDEAL house to the Windows server in the right room. The cable is a standard 9-pin to 9-pin Null Modem cable.

6.3 PowerUploader

PowerUploader is a Java Windows service constantly running to check for new power data. Every hour PowerUploader checks the local database it is configured for. If any new records have been added it sends an HTTP POST to the PHP Form for each new record.

6.3.1 PHP Form

The PHP form `insert.php` takes an authenticated HTTP POST request and stores the record offsite in the SQLite database ready for processing. Before inserting the record into the SQLite database the PHP form first checks that the user is authenticated to submit to the datebase. It then checks that a valid timestamp is being sent and that the energy, period and table values are not blank. If all the submitted values are acceptable the record is inserted into the database and confirmation is sent back to the client, else an error is returned.

If needed a webpage for manual user entry is available at manualsubmit.html. It is anticipated that manual submit will only be used for testing or for temporary use in the case of a failed sensor.

```
User: dbuser
Password: thaswu4e
```

7 Processing the Data

Processing of data occurs at the same time as the data is displayed. To allow access to live and historical data nothing is cached and instead queried directly from the database.

7.1 Storage

Initially the design called for the data logged to be stored in a MySQL database. This was initially selected due to it being the default database available at the hosting for therevproject.com. This required a MySQL server to be available where ever the program was installed, limiting its portability and increasing setup complexity.

Later in the development of the project it was decided that there was potentially a need to be able to install the program at different locations with unknown environments. So the program was transitioned to use an embedded database, SQLite. SQLite is a default extension to PHP5 and runs as part of the same processes. This means all the is required to be distributed is an SQLite database file with write-access file permissions.

7.2 Sumary Data

Summary energy data is generated using a number of SQL queries each time the page is displayed. These queries calculate the sums and averages of energy used over a number of periods from the power data. To generate the summary data 36 individual SQL queries are performed. This shifts most of the processing away from the PHP script and onto the database engine. However with a large dataset, such as a period of years, this process can take several seconds on a modern server. Algorithm 1 shows the SQL statement used to calculate the energy generated on a selected date.

Algorithm 1 SELECT SUM(energy*period/60) FROM generated WHERE date(time) = date($selectedDate)

7.2.1 Power Calculation

The power logging devices do not record the amount of energy used in a particular time period, but rather an instantaneous power value. To convert this reading into an energy value we assume it remains the average for the entire period. A shorter recording period will ensure more accurate energy values.

$Energy = Power * Period$

7.3 Graphs

Four graphs are generated as the page is displayed using the JpGraph PHP library. A single SQL query is used for each graph to select the data for a particular date from the datebase. The data is then loaded into an array for each axis and passed to the JpGraph library. JpGraph then produces a PNG image to return to the client. It was decided to return a plain image to the client to minimise client requirements. Alternatively the arrays of data could have been sent back to the client and graphs generated with Flash or JavaScript. While this may have allowed for some degree of interactivity client-side with the graphs, it was decided against to instead keep client requirements minimal.

Listing 2: PHP code to format a graph

```
$graph = new Graph(360,250);
$graph->SetScale( 'datlin',0,0,mktime(0),mktime(23));
$graph->xaxis->scale->SetDateFormat('H:i');
$graph->SetMargin(52,21, 35, 35);
$graph->SetMarginColor("black");
$graph->yaxis->SetLabelFormatString("%d W");
$graph->yaxis->SetTickSide(SIDE_LEFT);
$graph->xaxis->SetTickSide(SIDE_BOTTOM);
$graph->title->Set("Power Generated");
$graph->title->SetColor("gray");
$graph->yaxis->SetColor("gray");
$graph->xaxis->SetColor("gray");
$graph->ygrid->SetColor("gray");
$graph->xaxis->SetTextLabelInterval(3);
```

7.4 Grid Losses

The energy generated by the solar panels is being used to offset the energy used by the REV. As they are not directly connected and instead connect via the power grid there will be grid losses. Power is transferred into the power grid from the solar panels and out of the grid to the REV. Some estimates for other countries grid losses are ~7% from generation to consumer. To calculate the actual amount of energy received by the REV from the solar panels a grid loss percentage can be set in the PHP configuration file.

Listing 3: Configuration for a grid loss of 7.0%

```
$gridloss = 7.0;
```

8 Displaying the Data

Everytime a client loads the webpage the data is first processed before it is displayed. This webpage display was largely already completed by a previous student. It has been modified to allow the user to view energy data and graphs for previous dates. Two display webpages are available

display.php With the REV Project header and style

disply2.php With only the REV Project style and included files locally.

The first line of the page displays the date range of energy data available in the database. The second indicates the selected date for the daily summary and graphs. The daily summary is shown directly below this followed by the four graphs, this can be seen in Figure 3. Graphs are displayed for power being used and generated throughout the day aswell as for the cummulative energy being generated and used throughout the day as seen below in Figure 2. At the bottom of the page a summary of the energy used is presented showing

- Total EnergyGenerated
- Total Energy Used
- Total Energy Gained
- Days
- Average Generated Daily
- Average Used Daily
- Average Gained Daily

This summary data is grouped into the date ranges of

- Yesterday
- This Week
- Last Week
- This Month
- Last Month
- This Year
- Last Year
- All Time

With the Days row showing the number of days data is available for energy generated and energy used.

```
$daysGenerated.−.$daysUsed
```

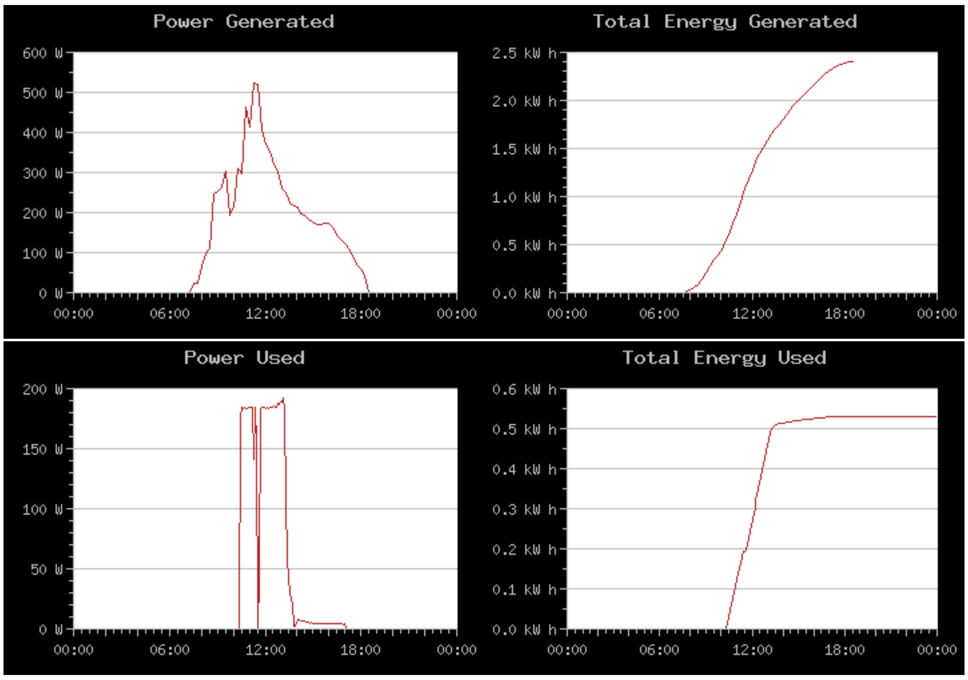
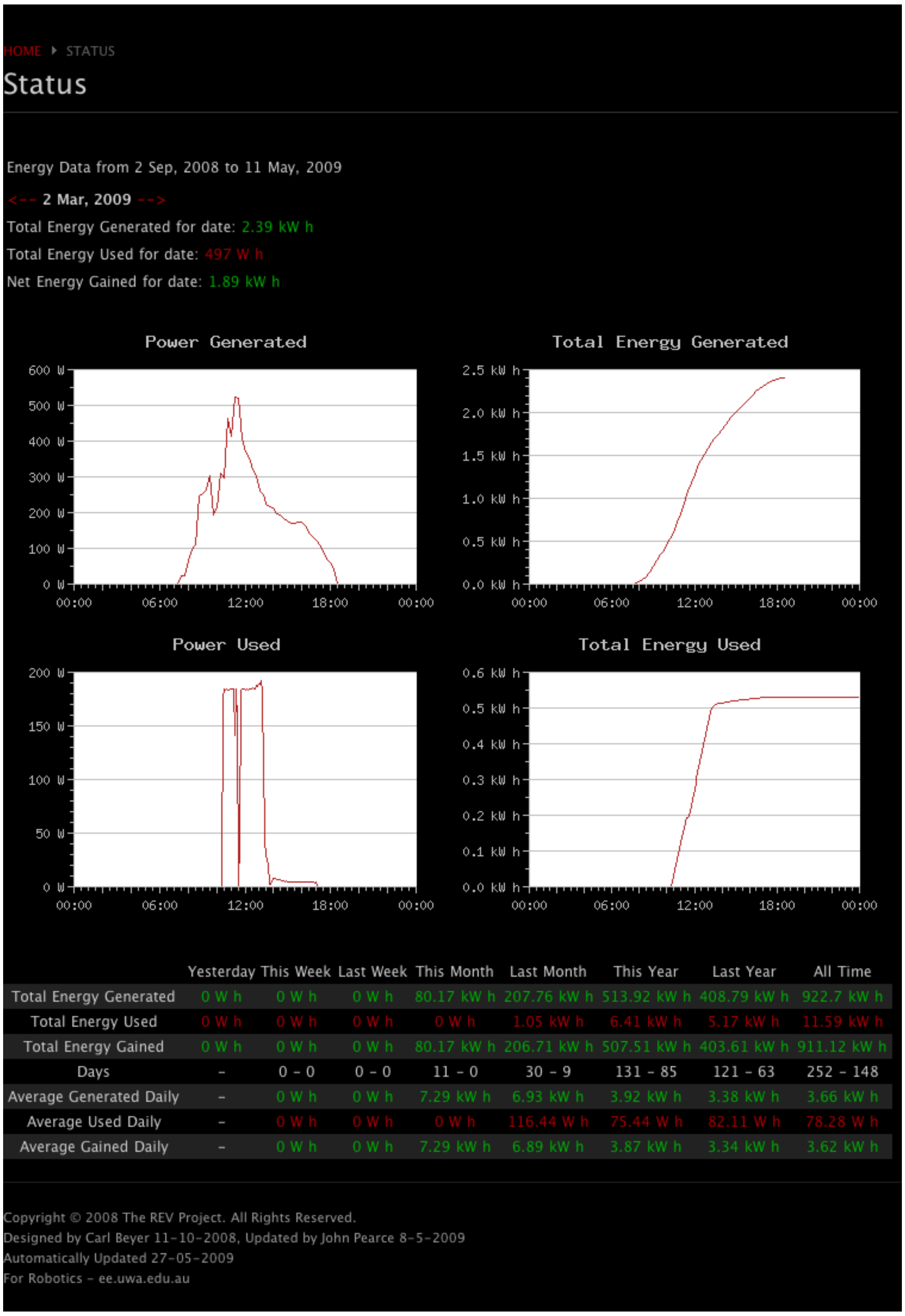


Figure 2: A sample of the graphs displayed

	Last Month	This Year	Last Year	All Time
Total Energy Generated	207.76 kW h	513.92 kW h	408.79 kW h	922.7 kW h
Total Energy Used	1.05 kW h	6.41 kW h	5.17 kW h	11.59 kW h
Total Energy Gained	206.71 kW h	507.51 kW h	403.61 kW h	911.12 kW h
Days	30 - 9	131 - 85	121 - 63	252 - 148
Average Generated Daily	6.93 kW h	3.92 kW h	3.38 kW h	3.66 kW h
Average Used Daily	116.44 Wh	75.44 W h	82.11 W h	78.28 W h
Average Gained Daily	6.89 kW h	3.87 kW h	3.34 kW h	3.62 kW h

Table 1: An abbreviated sample of the summary energy display



Status

Energy Data from 2 Sep, 2008 to 11 May, 2009

<-- 2 Mar, 2009 -->

Total Energy Generated for date: 2.39 kW h

Total Energy Used for date: 497 W h

Net Energy Gained for date: 1.89 kW h

	Yesterday	This Week	Last Week	This Month	Last Month	This Year	Last Year	All Time
Total Energy Generated	0 W h	0 W h	0 W h	80.17 kW h	207.76 kW h	513.92 kW h	408.79 kW h	922.7 kW h
Total Energy Used	0 W h	0 W h	0 W h	0 W h	1.05 kW h	6.41 kW h	5.17 kW h	11.59 kW h
Total Energy Gained	0 W h	0 W h	0 W h	80.17 kW h	206.71 kW h	507.51 kW h	403.61 kW h	911.12 kW h
Days	-	0 - 0	0 - 0	11 - 0	30 - 9	131 - 85	121 - 63	252 - 148
Average Generated Daily	-	0 W h	0 W h	7.29 kW h	6.93 kW h	3.92 kW h	3.38 kW h	3.66 kW h
Average Used Daily	-	0 W h	0 W h	0 W h	116.44 W h	75.44 W h	82.11 W h	78.28 W h
Average Gained Daily	-	0 W h	0 W h	7.29 kW h	6.89 kW h	3.87 kW h	3.34 kW h	3.62 kW h

Copyright © 2008 The REV Project. All Rights Reserved.
Designed by Carl Beyer 11-10-2008, Updated by John Pearce 8-5-2009
Automatically Updated 27-05-2009
For Robotics - ee.uwa.edu.au

Figure 3: A sample of the webpage

Part IV
Conclusions

9 Outcomes

By building upon the work and research of a previous student the development of the project went smoothly with no major problems encountered. As at first we were unsure of the best location to host the webpage developement took place under a standard environment of PHP and MySQL, later changing to PHP and SQLite. Under this standard environment such as is provided by`therevproject.com` the webpage operates successfully. Aside from the issues with transferring data outside of the UWA network the webpage and project in general have been a success and I have enjoyed the challenges faced.

10 Shortcommings

Due to the configuration of the internal UWA webserver `robotics.uwa.edu.au` the webpage has been hosted offsite at `therevproject.com`. This means that data must be sent outside of the internal UWA network and will be charged for. While the amount of data transferred is minimal this was still an issue I would have preferred to have solved. With further time to investigate the configuration of `robotics.uwa.edu.au` or by changing some of the configuration to support either SQLite or MySQL the problem could be resolved. At the time of this writing the SNAP account for the IDEAL house was setup with restricted access. Thus blocking the ability for data from the IDEAL house to be sent offsite by PowerUploader. If the SNAP account is changed to unrestricted access data should beging to flow from the IDEAL house to `therevproject.com` automatically.

11 Future Work

With a large record set the high number of SQL queries performed the webpage can take several seconds to load. The summary energy data is the most resource intensive to create and only changes daily. This data could be generated once per day and cached for all other users that access the page. The graphs that are generated could also be cached each day. Implementing caching could significantly reduce the load times and the processing resources required.

The wirless strength of the SNAP connection is quite low in the IDEAL house and the antenna must be postitioned at the window to make a connection. It is likely that future projects in the IDEAL house will also require a connection to either the UWA internal network or the Internet. Further investigation into a larger antenna or an alternative Internet connection such as 3G could be beneficial.

Part V
Appendix

A CDROM Contents

/Report Contains a copy of this report

/PowerUploader

- **/bin** PowerUploader binary and setup
- **/src** PowerUploader source code

/Webpage The files that must reside on a webserver

/DB Contains sample power logging SQLite databases

README.TXT Installation and configuration notes

B JAVA Configuration File

```
#PowerUploader Configuration

#Time/date of the last successful recording submitted
#Updated automatically
lastdate=2008-12-05 11\:35\:00

#Path to form
form=insert.php
#Host the form is running on
host=http\://localhost/
#Username an password to submit
user=dbuser
pass=thaswu4e
#Table to submit to
#generated or used
inserttable=used
#Period (minutes) of recorded power vales
period=5

#Maximum power value to upload
#10A * 240V = 2400W, anything bigger than this is an error
maxenergy=2400

#JDBC Driver to source database
jdbcdriver=sun.jdbc.odbc.JdbcOdbcDriver
#Connection string to source database
conn=jdbc:odbc:EnergyProf

#Table in source DB to read from
```

```
#dbo_LOG_DETAIL_VALUES or Sensor5
table=Sensor5

#Energy column in source DB to read from
#Value or Data_orig
energycol=Data_orig

#Time column in source DB to read from
#LOG_DATE or Data_orig
timecol=Time_rounded

#Any additional SQL statements required
#FK_CHANNEL = 3 AND
extrasql=
```

C PHP Configuration File

```
<?php
/**
 * Global Constants
 */

//Login for DB Insert
$user = "dbuser";
$pass = "thaswu4e";

//DB connection string
$db = "sqlite:powerlogging.sqlite";

//Timezone
$timezone = "Australia/Perth";

//The percentage of power lost through the grid
$gridloss = 0.0;

/**
 * Global Functions
 */

//Set the default timezone
date_default_timezone_set($timezone);

?>
```