THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Cloud Based Monitoring of a Renewable Energy System

ENGG492: Honours Research and Management Project

Final Report

October 2016

Jordan Crawford

Supervisor: Mark Apperley

This report is in partial fulfilment of the requirements for the degree of Bachelor of Engineering with Honours at The University of Waikato

Abstract

Renewable energy systems are becoming more popular as the price of solar panels and batteries decrease. A renewable energy system provides energy to a home using renewable methods of generation like solar PV or small wind turbines. There are three types of renewable energy systems; grid-tied, off-grid and hybrid. Grid-tied and hybrid systems retain a connection to the national electricity grid and can use this as a medium to sell excess energy. Off-grid and hybrid systems use batteries to store excess energy for times when generation is low. Off-grid systems use generators to provide a backup source of energy. Current methods of determining the state of a renewable energy system require the use of inaccurate readouts physically located on the equipment. Cloud based monitoring systems allow access to information from anywhere and can make use of the additional computing resources available in the cloud. In this project, a web based application has been developed that monitors a renewable energy system. The application consists of three components: the bridge software runs on a micro-computer on-site and sends readings to the backend; the backend runs on a cloud-server and processes sensor data; the frontend can be accessed from any internet connected device to show the state of the system. The application calculates the state of charge of the battery and provides information on energy flows within the system. Using this, the application calculates the future state of the system based on historical trends, allowing the next battery full or empty event to be determined. The application is highly configurable, allowing it to be used with any renewable energy system. It serves as a base for future work, including the development of functionality like generator automation and weather based predictions.

Table of Contents

Abstract ii

Table of Contents iii

Table of Figures vi

Glossary of Terms vii

1 Introduction 1

2 Background 4

2.1 Renewable Energy Systems 4

2.1.1 Types of Renewable Energy Systems 4

2.1.2 Common Methods of Renewable Generation 5

2.1.3 Batteries 6

2.1.4 Equipment used in Renewable Energy Systems 8

2.1.5 Reasons for being off-grid 10

2.2 Renewables Industry 11

2.2.1 Solarcity NZ 11

2.2.2 SolarCity US 11

2.2.3 Tesla Powerwall 12

2.2.4 Enphase Home Energy 12

2.3 The Electricity Grid 13

2.3.1 Smart Meters in New Zealand 13

2.3.2 Energy Monitoring for Grid Connected Consumers 14

2.3.3 Efforts to Change Consumption 15

2.3.4 Future of the New Zealand Grid 16

2.4 Energy Monitoring Systems 17

2.4.1 Kuenemann's Off-grid Monitoring System 17

2.4.2 Alahmari's Off-grid Monitoring System 18

2.4.3 SunFarmer Energy X 19

2.4.4 Enphase Enlighten 19

2.4.5 Automated Energy Management 20

2.5 Determining the State of Charge 21

2.5.1 Importance of State of Charge 21

2.5.2 Historical Methods 21

2.5.3 B3SOC Algorithm 23

3 Analysis 25

3.1 Objectives of the Application 25

3.2 Field Off-grid Setup 25
3.2.1 PentaMetric DC Monitor 26
3.2.2 Smart Circuit AC monitor 27
3.2.3 Intense PC Computer 27
3.2.4 Determining the wind turbine current 27
3.2.5 Laboratory Setup 27
3.3 Application Components 28
3.3.1 Bridge 28
3.3.2 Backend 29
3.3.3 Frontend 29
3.4 Conceptual Design 30
3.4.1 Getting data into the cloud 30
3.4.2 Displaying basic sensor data in the cloud 30
3.4.3 Processing readings and finding battery state of charge 30
3.4.4 Providing information about energy flows within the system 30
3.4.5 Future battery state estimation 31
3.4.6 Automatically powering on and off the generator 31
4 Implementation 32
4.1 Development Process 32
4.2 Architecture 33
4.2.1 Bridge 33
4.2.2 Backend 34
4.2.3 Frontend 35
4.3 Getting data into the cloud 36
4.3.1 Communication with sensor devices 36
4.3.2 Sending readings to the backend 38
4.4 Displaying basic sensor data in the cloud 38
4.4.1 Buildings, Bridges, Sensors and Readings 38
4.4.2 Bridge Authentication 38
4.4.3 People Authentication 39
4.4.4 Sensor Readings Explorer 39
4.4.5 CSV Export 40
4.5 Processing readings and finding battery state of charge 41
4.5.1 States and Processing data 41
4.5.2 The B42SOC Algorithm 43
4.5.3 Implementing B42SOC 44
4.5.4 Processing Parameters 45
4.6 Providing information about energy flows within the system 46

4.6.1 Energy Sources ... 46
4.6.2 Energy Flow Processing ... 47
4.7 Future battery state estimation ... 48
4.7.1 Energy Flow Predictions ... 49
4.7.2 Future States ... 51
5 Evaluation ... 53
5.1 System Implementation ... 53
5.2 Future Work ... 56
5.2.1 Data collection ... 56
5.2.2 Frontend administrative functionality ... 57
5.2.3 Data processing ... 57
5.2.4 Development environment ... 57
6 Conclusion ... 58
7 References ... 60
Appendix A – Backend Model Schema ... 64
Appendix B - Configurable Parameters ... 66
Appendix C - User Authorisation Rules ... 69

Table of Figures

Figure 1. Diagram of a typical renewable energy system. 4
Figure 2. A Tesla Powerwall installation (Tesla, n.d.) 12
Figure 3. Diagram showing how the Envoy connects to micro inverters and the internet (Enphase, 2016) 13
Figure 4 - A screenshot of Google PowerMete (Fehrenbacher, 2011) 14
Figure 5. The Powershop NZ app's electricity usage page (Powershop, n.d.). 15
Figure 6. The Choice app's usage recommendations (Flick Electric, n.d.). 16
Figure 7. The Choice app's generation split section (Flick Electric, n.d.). 16
Figure 8. Screenshot of the dashboard of Alahmari's monitoring system. 18
Figure 9. The Enphase Enlighten dashboard. 20
Figure 10. A diagram of the Pauaeke setup with its monitoring points. 26
Figure 11. The components of the system. 28
Figure 12. A screenshot of the AdminLTE template. 35
Figure 13. The login page for the frontend. 39
Figure 14. The reading explorer which supports filtering and ordering raw reading data. 40
Figure 15. The export page in the frontend. 40
Figure 16. An example of the format of the CSV export. 41
Figure 17. The "Data processing" section of the Configuration page. 42
Figure 18. The list and detail views of the Battery State page. 43
Figure 19. The graph of state of charge, charge level and battery capacity in the State page. 45
Figure 20. Battery level gauge on the Summary page. 45
Figure 21. Part of the Parameters section on the frontend's Configuration page. 46
Figure 22. The Daily Energy Sources graph showing renewable and non-renewable generation. 47
Figure 23. The Daily Total Energy graph showing consumption and generation. 48
Figure 24. The Energy Flow graph shown on the Summary page. 48
Figure 25.The solar panel generation prediction graph shown on the Prediction Patterns page. 50
Figure 26. The consumption prediction graph shown on the Prediction Patterns page. 50
Figure 27. The table of FutureState's on the Future State page. 52
Figure 28. An example of the charge capacity being pushed up significantly. 53
Figure 29. An example of the charge efficiency being re-calculated after a low battery voltage event in the night. 54
Figure 30. An example where the total daily consumption is less than the total generation. 55
Figure 31. The system in use on a mobile device. 59

Glossary of Terms

Voltage (V):

The difference in charge between two points. The unit is Volt (V) (SparkFun, n.d.).

Current (A):

The rate of flow of charge. The unit is Amp (A) (SparkFun, n.d.).

Energy (J):

Energy is the ability of an object to do work on another. Energy is measured in Joules (J) (SparkFun, n.d.).

Power (W):

Power is a measure of energy over a set amount of time. The unit is a Watt (W), which represents a joule of energy per second (SparkFun, n.d.).

Watt-hour (Wh):

A Watt hour is a unit of energy. It is equal to the amount of energy as 1W over the period of an hour (i.e.: 3600J). Battery capacities and solar array sizes are commonly measured in kilowatt-hours (kWh) (Rouse, 2005).

Alternating Current (AC):

Alternating current changes direction periodically. Electricity outlets are almost all AC. The grid is based on AC because transporting AC over long distances results in less loss in the power line due to resistance (SparkFun, n.d.).

Direct Current (DC):

Direct current flows in one direction. DC is used by batteries and sources of generation like solar or wind (SparkFun, n.d.).

State of Charge (SoC):

An indication of the level of charge in a battery relative to its full charge capacity.

1 Introduction

A renewable energy system provides energy to a home using methods of renewable generation, typically; solar PV, small wind turbines, or micro-hydro. There are three main types of renewable home energy systems; grid-tied, off-grid and hybrid. This work primarily focuses on off-grid systems, but some features of the system developed are applicable to hybrid systems and grid-tied systems. An off-grid system is completely disconnected from the national electricity grid and uses batteries to store excess energy for use when generation is low. Off-grid systems use a petrol or diesel generator to power the home during times of low generation or high consumption. Hybrid systems use renewable generation and a battery. However, instead of a generator, these systems use a connection to the electricity grid as a backup energy source. Hybrid systems are likely to become increasingly more popular as the price of solar PV panels and batteries decrease.

Electricity is an essential utility for today's modern conveniences. Occupants of off-grid homes are responsible for ensuring they have enough electricity to power the appliances they depend on. In order to keep their energy supply working, they must decide when to turn on or off their generator or decrease their consumption. In addition, generator use should be minimised as fossil fuels such as petrol or diesel are costly and their use is bad for the environment. Presently, to make these decisions, the occupant of an off-grid home must use readouts from equipment in the renewable energy system.

Using the readouts from equipment in the renewable energy system, the occupant wants to determine the state of charge (SoC) of the batteries. This is an indication of how much energy is available in the battery relative to the batteries capacity. The indications of state of charge available from these readouts is typically inaccurate, mostly because they assume the system is under a state of rest (i.e.: no current flowing in or out). However, in reality, an off-grid system is almost never in a state of rest with generation occurring at all times of the day and a constant electricity load from the home.

As the readouts are physically located on the equipment, they require users to walk to where the equipment is located. This makes it a hassle to check the state of the system often. These readouts also provide no indication of the previous state of the system or its potential future state. Because the equipment cannot provide a view of the future state of the system, a user must have a full understanding of their typical consumption and generation patterns to determine what the system will do in the future.

The aim of this project was to develop a cloud based application that can monitor a renewable energy system. A cloud based monitoring application gathers data from the system and provides access to this information from anywhere through the internet. This allows the use of a smartphone or computer as a more convenient and powerful way to display information than readouts on the equipment. In comparison to a monitoring system solely based on-site, cloud based applications use minimal computing resources, reducing the power consumption and cost. The cloud also provides additional reliability and allows the application to utilise the near infinite storage and computing resources available. This means the application can perform additional data processing to make it more useful. None of the existing cloud based systems that monitor a renewable energy system provide the state of charge of the battery, neither do they provide an estimation of the future state of the system.

The application should provide users with the current and historical state of charge of the renewable energy system. This lets users determine whether they need to turn the generator on or if there is enough energy in the batteries to run a high power appliance. The state of charge should be determined without requiring the battery to be at rest, without relying on the chemistry of the battery, and in a way that allows for ageing of the battery and other factors that affect its performance.

To help users discover how to improve the way they use their renewable energy system, the application should provide information about the energy flows in and out of the system. For example, this may allow a user to determine if they should invest in more solar panels or energy storage to improve their system. The application should also maintain information about typical consumption and generation patterns. Using these patterns, the application can determine the future state of the system so the user knows when their battery is likely to become empty or full.

To make day to day usage of an off-grid system easier, the application should automatically power on and off the generator. This should be at a time where the energy from the generator is maximised, such as while energy consumption is highest. Un-necessary generator use should be avoided by taking predicted energy generation into account. This lets the application ensure the home will never loose electricity.

The monitoring application was developed over a seven-month period, with functionality being tested at a real off-grid home. The application achieves all of its original aims, except automatically powering the generator on and off. This feature could not be implemented due to time limitations of the project. The application helped with the development of the B42SOC (Apperley M. , 2016) algorithm and allowed the limitations of this algorithm to be determined. The application serves as a base for future work while providing many helpful features.

This report outlines the research and design process of the project. Chapter 2 explores the background to this work including: an overview of renewable energy systems; the renewables industry; the electricity grid; energy monitoring systems; and methods of determining the state of charge of the battery. Chapter 3 covers the analysis of the requirements of the application and the approach to development. Chapter 4 covers the implementation of the application, including its general architecture and each stage of the development process. Chapter 5 evaluates the application’s implementation and outlines potential areas of enhancements.

2 Background

This section covers the research performed as background to the implementation of the monitoring application. Renewable energy systems and their typical components are introduced. The renewables industry, electricity grid and existing energy monitoring systems are explored. This section also presents the existing methods of determining the state of charge of a battery.

2.1 Renewable Energy Systems

A home renewable energy system is a set of hardware components used to provide energy to a home. These use methods of renewable generation, such as solar PV, wind, or micro-hydro. Depending on the type of renewable energy system, these may retain a connection to the national electricity grid. In this case, the grid is used as an additional energy source or as a medium to sell excess energy. Off-grid and hybrid renewable energy systems use batteries. There are several types of batteries, each with factors that affect how they perform. This section also explores the equipment typically used in a renewable energy system and the reasons for a home to utilise an off-grid renewable energy system. A diagram of a typical renewable energy system is shown in Figure 1.

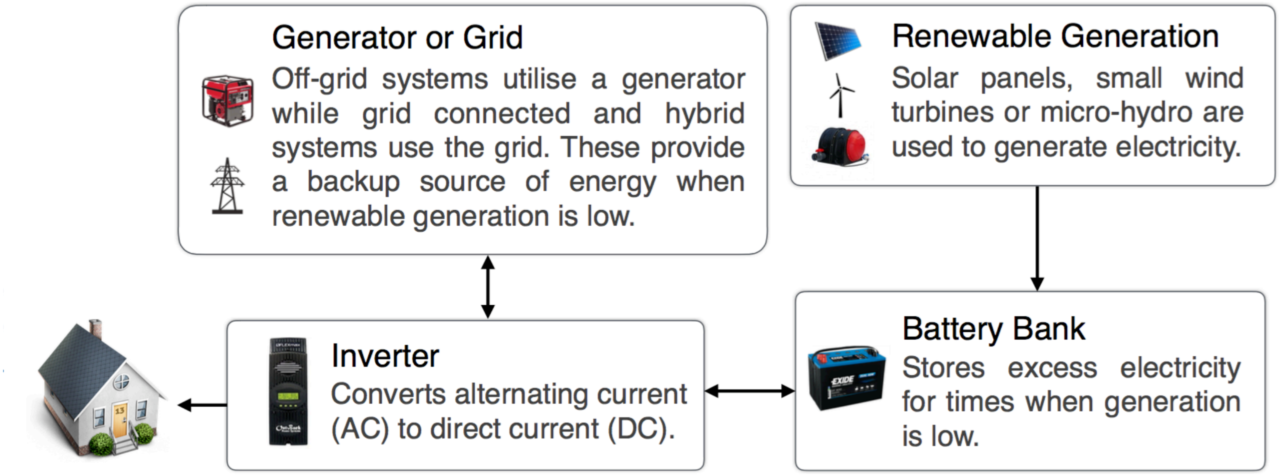


Figure 1. Diagram of a typical renewable energy system.

2.1.1 Types of Renewable Energy Systems

There are three main types of renewable energy systems used in homes. These are:

- Grid-tied.
- Off-grid.
- Hybrid.

2.1.1.1 Grid-tied systems

Grid-tied renewable energy systems utilise renewable energy sources in addition to the grid in order to reduce costs. During times of peak generation, a grid-tied system may supply all of the home's electricity. Excess energy can be sold back to the energy grid through feed-in tariff schemes (Maehlum, 2013).

This type of system reduces cost and requires minimal equipment while still benefiting from the reliability of the grid. However, the buyback rate for energy sold back to the grid has been decreasing. This is likely due to solar generation mostly occurring during off-peak hours and the buyback rate being significantly higher than wholesale electricity prices (Radio New Zealand, 2014).

2.1.1.2 Off-grid systems

Off-grid systems are completely disconnected from the electricity grid. As a result they use batteries to store excess energy for use when generation is low. A backup generator should be installed to ensure electricity supply during times with low generation or high usage (Maehlum, 2013). Reasons for being off-grid are explored in section 2.1.5.

2.1.1.3 Hybrid systems

Hybrid systems combine the benefits of grid-tied and off-grid systems. These have methods of renewable generation, a grid connection, and batteries. Rather than selling energy back to the grid, generated energy is stored in the batteries for use during peak hours when the grid prices are more expensive. Hybrid systems can use their batteries for load shifting. This involves selling their energy back to the grid when the buyback rate is the highest and charging the batteries when the grid purchase rate is the lowest. Because these systems still have a grid connection, they do not require a backup generator (Maehlum, 2013).

2.1.2 Common Methods of Renewable Generation

Renewable generation methods convert natural energy into electricity. This provides free energy that does not deplete the worlds resources or cause pollution. Typical home renewable energy systems use either solar PV to harvest energy from the sun, small wind turbines to harvest energy from the wind, or micro-hydro to harvest energy from the flow of a stream.

2.1.2.1 Solar Photovoltaic

Solar Photovoltaic (PV) cells convert sunlight into electricity. The intensity of the sunlight determines how much energy is generated. Solar PV cells are joined together to form a PV panel. A solar PV panel has a direct current (DC) output. An installation of multiple panels is called a solar PV array and are often installed on roofs (EECA Energywise, 2016). Most solar panel specification sheets define the amount of power they can produce (often called P_{max}). This is under ideal, standard test conditions, and in reality the output will be less (Dunphy, 2014). The cost of solar panels is falling exponentially due to their increasing popularity. The "soft-costs" of solar installation must also be considered; this includes the cost of installation and maintenance of a solar array, which are also gradually decreasing (Mosaic, n.d.).

2.1.2.2 Small wind turbines

Wind turbines generate electricity when the wind turns the rotor blades of a turbine. This turns a shaft that is connected to a generator. Wind turbines are often mounted on towers so they are exposed to the most and highest speed wind. Wind turbines tend to work best in rural areas with less obstructions like buildings and trees. The capacity of a wind turbine is measured by the amount of power at a particular wind speed. For example, the specification may state the output as 1 kW in a 15 meters per second wind (EECA Energywise, 2015).

2.1.2.3 Micro-hydro

Micro-hydro systems harvest energy from the flow of a stream. These can provide 24 hours of energy and are unaffected by sun or wind conditions, however their output can be affected by the seasons. Micro-hydro systems produce either alternating current (AC) or direct current (DC) energy. DC based micro-hydro requires the turbine to be located close to the energy system to avoid losses in transmission. Location is important as some positions in the stream may have quicker flow than others. A solution is to use a micro-hydro system that produces an unrectified AC output as this reduces line loss and the energy can be rectified to DC at the energy system. This means the micro-hydro system can be positioned further away to maximise its output (Lance, 2009).

2.1.3 Batteries

Batteries are used in off-grid and hybrid renewable energy systems. These store energy for use when renewable generation is low. There are various types of batteries and a number of factors that impact their performance.

2.1.3.1 Common types of batteries

There are two main types of batteries used in home renewable energy systems, lead acid and lithium-ion.

Lead-acid

Lead-acid batteries are the most common type of battery for off-grid systems (Sindelar, 2011). Flooded lead-acid batteries must be refilled regularly because their electrolyte evaporates during charging. These also need ventilation because they expel hydrogen gas. Absorbed glass mat (AGM) and gel based lead acid batteries do not require any maintenance (Zipp, 2015).

Charing lead-acid batteries

A lead-acid battery charges in three phases. The bulk of charge is applied during the first half of the total charge time with a constant current. Then, until the battery is fully charged, a lower charge current is used. Beyond this point, a float charge is used to compensate for loss by self-discharge (Battery University, n.d.).

Lithium-ion

Lithium-ion batteries are common in portable devices like smartphones and laptops. These batteries are also commonly used in electric vehicles. Lithium-ion batteries are an emerging technology and their cost is quickly decreasing, although lead-acid batteries remain the cheapest option. The current cost of a lithium-ion battery for an electric vehicle is around $400 USD / kWh, but estimates predict it to be around $125 USD / kWh by 2022 (Shahan, 2016). Lithium-ion batteries require a battery management system to monitor the voltage and temperature to prevent excessive charging or discharging. These batteries have several benefits over lead acid, such as a higher number of cycles throughout their lifetime, less self-discharge and they are lighter. Section 0 covers the Tesla Powerwall, a lithium-ion battery designed for home solar panel installations.

2.1.3.2 Factors that affect battery performance

The performance of a battery is affected by many different variables (Bas, 2010):

- Cyclic life. Batteries have a limited number of cycles they can handle. The performance decreases as the number of cycles increase until the battery stops working.
- Depth of discharge (explored below).
- Temperature. At cold temperatures, batteries have lower capacities. At hot temperatures, the cyclic life is reduced.
- Charge or discharge rate. A high charge or discharge rate can lead to unwanted chemical reactions that reduce the battery capacity (Woodbank Communications Ltd, n.d.)

Impact of depth of discharge on battery life

There is a roughly logarithmic relationship between the average number of cycles a battery is capable of and the depth of discharge of the battery (Woodbank Communications Ltd, n.d.). An owner of an off-grid electricity system must appropriately balance the capacity of the system and the depth of discharge used. If a system is rated for 15,000 cycles at 5% depth of discharge or 1,000 cycles at 55% depth of discharge, then a 5% depth of discharge is capable of 750 cycles of the full capacity, while 55% depth of discharge results in 550 cycles of the full battery capacity. While a smaller depth of discharge extends the life of the battery, a much larger battery may be required to fulfil the household requirements, increasing the overall cost (Murphy, 2012).

2.1.3.3 Typically battery bank voltages

Battery banks used in renewable energy systems are generally either 12V, 24V or 48V. Traditionally, 12V systems were popular and 24V or 48V inverters and charge controllers were uncommon. However, now 24V is more common as this voltage works better with larger systems (Solar Homestead, 2015).

2.1.4 Equipment used in Renewable Energy Systems

Additional equipment is required make Renewable Energy Systems work. Solar charge controllers are essential when solar panels are used with battery storage and inverters convert direct current from the energy system to alternating current for use in a building.

2.1.4.1 Solar Charge Controller

A solar charge controller regulates the energy charge being delivered to a battery by limiting the rate of current. This prevents batteries from overcharging and keeps the battery healthy (Maehlum, 2013). Solar charge controllers are only required when using solar panels with battery storage.

Maximum Power Point Tracking Solar Controllers

The solar panel voltage varies depending on the temperature, with high temperatures resulting in lower voltages. As a result, the voltage of a solar array may not always match the voltage of the battery. This leads to energy losses when the solar panel voltage is greater than the battery voltage. A maximum power point tracking (MPPT) solar controller converts the solar panel's DC input to a DC output that best matches the battery voltage. This allows the maximum output from the solar panels (Northen Arizona Wind & Sun, n.d.).

2.1.4.2 Inverters

All equipment in a typical renewable energy system uses direct current (DC). This is because batteries and many sources of renewable generation are DC based. The electricity grid uses alternating current (AC) because this results in less energy loss when transmitted over long distances; as a result, all appliances in homes use AC. Inverters convert energy from direct current (DC) to alternating current (AC) to allow energy from the renewable energy system to be used in a house. There are many types of inverters, each with different features and sizes. Inverters may also support low voltage disconnect or charging. Inverters are designed for different purposes, with some suited to grid-tied installations and others for off-grid installations.

Low Voltage Disconnect / Shutdown

Some inverters include low voltage disconnect (LVD) functionality. When the battery voltage drops below a low point, the inverter output is automatically switched off to prevent damage to the battery (as mentioned in section 2.1.3.2, the depth of discharge has impact on battery performance). Once the battery voltage reaches a high point that represents charging, the inverter output is turned on again. For example, for a nominal 12V system it may cut off at 11.5V and reconnect at 12.5V (REUK.co.uk, n.d.). This can also be called a low voltage shutdown (LVSD) (Apperley & Alahmari, 2013).

Charging

Inverters with charging functionality use the AC input from a generator or the grid to charge the house batteries. While the charger is on, the house loads are powered from the AC input allowing the use of high power equipment (Xantrex, n.d.).

Grid-tied Inverters

A grid-tied inverter synchronises its output phase and frequency to match the grid. These are only required when selling energy back to the electricity grid (Maehlum, 2013).

Off-grid Inverters

Off-grid inverters convert energy from DC to AC without needing to synchronise it to the grid supply (Maehlum, 2013). In the context of solar-panels, these are called string inverters because a single inverters converts the energy from several solar panels.

Micro-Inverters

Some solar PV based systems use micro-inverters on each panel. This results in more efficient generation because a panel in shade does not decrease the efficiency of the entire solar array (What Power Crisis, n.d.). Each solar panel can be individually monitored to identify the most efficient position to optimise the amount of electricity generated. This results in a more modular system as adding more solar panels does not require the purchase of a larger inverter.

2.1.5 Reasons for being off-grid

The two main reasons for a household to install an off-grid system are the high cost or impracticality of setting up a grid connection and the potential for reduced electricity costs.

2.1.5.1 Cost of a setting up a grid connection

This appears to be the most common reason to go off-grid. In some remote areas, obtaining a connection to the grid can be very costly or impractical, sometimes with quotes of $30,000 NZD or higher (Alahmari, Off-grid Energy Monitoring, 2013). In a survey of twelve off-grid households in Canada, eight required an off-grid system due to the cost of bringing utilities to the site (Canada Mortgage and Housing Corporation, 2001).

2.1.5.2 Reduced electricity costs

Renewables are becoming a way of reducing cost for its users, rather than just as method of becoming environmentally responsible (Banerjee, Rollins, & Moran, 2011). Of twelve off-grid households in Canada, four said their decision to go off-grid was influenced by the opportunity of long term cost savings (Canada Mortgage and Housing Corporation, 2001).

2.2 Renewables Industry

The renewables industry consists of solar providers with unique solar financing offers that encourages solar installations in homes. Battery systems like the Tesla Powerwall (Tesla, n.d.) and the Enphase AC Battery (Enphase, 2016) are aimed at the mass market of potential hybrid systems. Enphase's battery is part of the Enphase Home Energy system, a range of renewable energy products.

2.2.1 Solarcity NZ

Solarcity is a solar provider in New Zealand. Solarcity started with the traditional model of selling solar panel installations but in 2015 began offering their solarZero pricing model. In this model, home owners get solar panels and maintenance for no up-front costs but instead pay Solarcity for electricity on a per unit basis with a fixed price over a 20-year contract. Solarcity retains ownership of the systems installed through the solarZero model (McClure, 2015). Monitoring is powered by Enphase's MyEnlighten app (Solarcity, n.d.).

2.2.2 SolarCity US

SolarCity is a solar provider in the United States (SolarCity, n.d.). They have a similar business model as Solarcity in New Zealand but despite the same name, are not associated. The MySolarCity app allows users to monitor their energy generation and consumption. SolarCity customers have the option of linking their system with batteries to go from a grid-tied to a hybrid system. SolarCity has over 250,000 customers as of July 2015 (Wesoff, 2015).

2.2.3 Tesla Powerwall

The Tesla Powerwall is a wall mountable lithium-ion battery pack (Figure 2 shows an example of this installed on a house). The Powerwall is designed to be used with home solar installations. The Powerwall has a capacity of 6.4kWh and sells for $3,000 USD (Tesla, n.d.). The Powerwall is one of the few lithium-ion based batteries aimed at home installations. The Powerwall has received significant media coverage and demand due to Tesla's popularity in the electric vehicle space. Vector Limited, a New Zealand electricity and gas distribution company, has a partnership with Tesla Energy to sell the Powerwall in New Zealand (Vector Limited, n.d.).

Figure 2. A Tesla Powerwall installation (Tesla, n.d.)

2.2.4 Enphase Home Energy

Enphase Energy is a California based energy technology company. Enphase has several products that form the AC-coupled Enphase Home Energy Solution. In comparison to traditional DC-coupled off-grid systems that use a single inverter, an AC-coupled system uses AC transmission between all its components. Enphase states that AC-coupled systems have equivalent efficiency to DC-coupled systems but with increased safety and scalability. To achieve this, Enphase systems use solar panels with micro-inverters so the solar generation is AC based (Enphase, n.d.).

The Enphase AC Battery has a lithium iron phosphate chemistry, a 1.2kWh capacity and a built-in inverter. The specification sheet lists a 95% depth of discharge and 96% roundtrip efficiency. The warranty covers 80% capacity for the soonest of 10 years or 7,300 cycles. Each battery is small but the system is designed to be very modular, allowing a user to expand their total capacity by adding more batteries at a later point. The Enphase AC Battery is launching in the United States in late 2016 (Enphase, 2016).

Enphase Envoy is a device that connects to Enphase's other products to send data to the internet for monitoring through Enlighten. To communicate with micro-inverters and the AC Battery, Envoy uses Power Line Communication through the existing powerlines in the building. Envoy connects to the internet through Wi-Fi or Ethernet (Enphase, 2016). Figure 3 shows how Envoy communicates with other equipment in the home and with the internet.

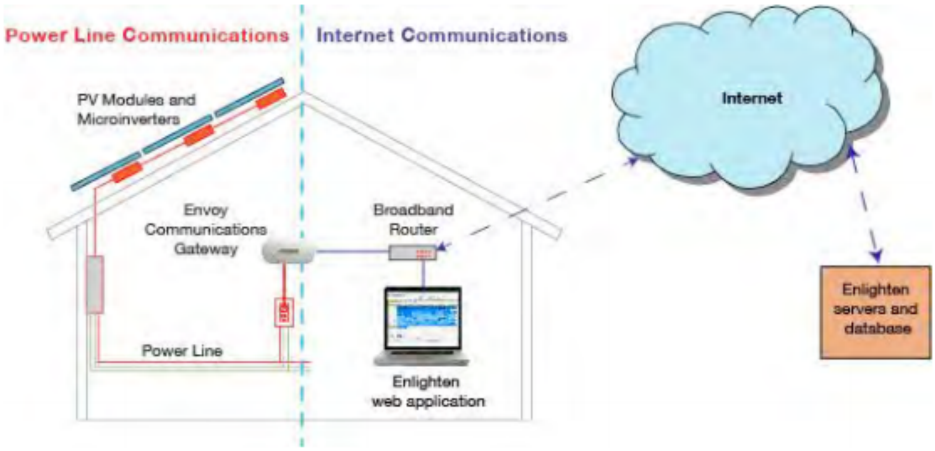


Figure 3. Diagram showing how the Envoy connects to micro inverters and the internet (Enphase, 2016)

2.3 The Electricity Grid

The New Zealand electricity grid is getting smarter with the installation of smart meters and monitoring systems for grid connected consumers. Some electricity providers encourage consumers to change how they use electricity to help smooth the grid and make the most of renewables. In the future, the grid is expected to have a smoother load profile and higher reliability through distributed storage.

2.3.1 Smart Meters in New Zealand

Smart meters allow electricity providers to receive electricity usage data for a building at regular intervals (typically half-hourly) (Electricity Authority, 2016). This means they can provide graphs and other information to end users that can help manage their energy usage. This is in contrast to an analogue reader which is typically read only every two months. Smart meters broadcast data in a similar method to a mobile phone or through a mesh network between meters.

There are 1.25 million smart meters installed in households across New Zealand as of March 2016 (Bridges, 2016). This can help occupants of grid connected homes observe patterns in their usage and reduce their consumption.

2.3.2 Energy Monitoring for Grid Connected Consumers

As smart meters are becoming more common, there are more ways for grid connected consumers to monitor their energy usage. This can help users realise how they use energy and reduce consumption in order to save money.

2.3.2.1 Google PowerMeter

Google PowerMeter is a discontinued product by Google.org. This was a web application that allowed occupants of grid-connected homes with smart meters to access details about their electricity usage (Kopytoff, 2009). The product was available from October 2009 until it was retired in September 2011 due to having too few users. A possible reason for its lack of growth is that it required users to opt-in to the service and may have been seen as a threat to electricity utility companies (Fehrenbacher, 2011). A screenshot of Google PowerMeter is shown in Figure 4.

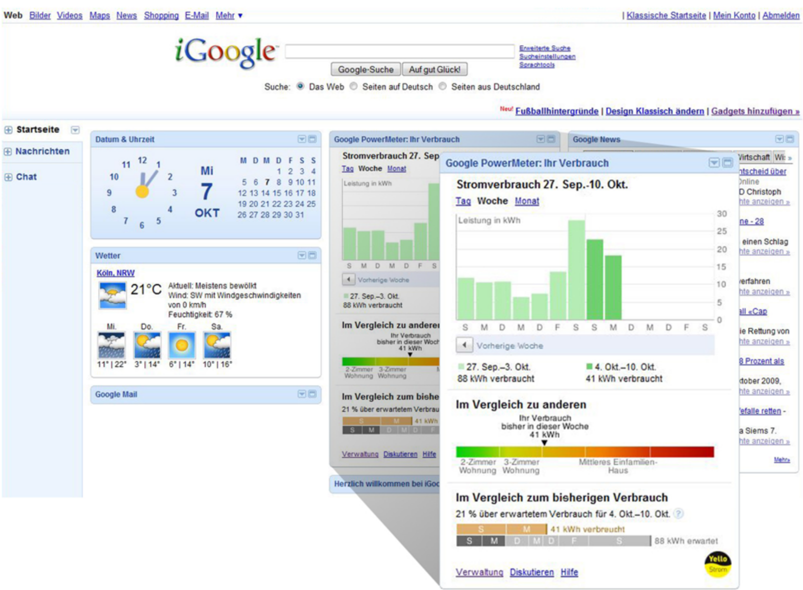


Figure 4 - A screenshot of Google PowerMete (Fehrenbacher, 2011)

2.3.2.2 Powershop

Powershop is a New Zealand electricity retailer. They provide functionality for customers to track their electricity usage and cost through the Powershop website and mobile app. If a user does not have a smart meter they can manually enter a meter reading through the app or website so they may still utilise the monitoring functionality (Powershop, n.d.). A screenshot of the Powershop app's electricity usage page is shown in Figure 5.

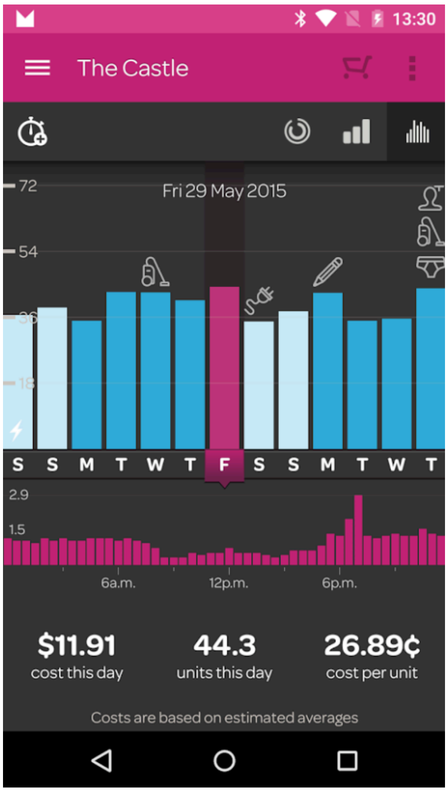


Figure 5. The Powershop NZ app's electricity usage page (Powershop, n.d.).

2.3.3 Efforts to Change Consumption

Several New Zealand electricity providers encourage their customers to change the way they use electricity to save money, flatten the electricity grid and boost renewable usage. Electric Kiwi offers customers a free hour of power during an off-peak time and Flick Electric has an app that provides a recommendation about the most environmentally friendly time to use electricity.

2.3.3.1 'Free hour of power' by Electric Kiwi

Electric Kiwi Limited is an electricity provider that allows customers to set a 60-minute period each day where they will not be charged for electricity. This is possible because residential smart meters provide regular consumption data to electricity providers, so usage within a single hour can be determined. This period must be throughout off-peak times; these are between 9am and 5pm or 9pm to 7am. Electric Kiwi suggest doing tasks like running clothes driers, running dishwashers or charging electric cars during this time. This helps to smooth the electricity grid by shifting large loads to off-peak times. This reduces the need for standby electricity generation plants during peak overload times such as during cold snaps in the winter. Standby electricity generation plants are costly to start and maintain so less frequent use could reduce the cost of electricity (Electric Kiwi, n.d.).

2.3.3.2 Choice by Flick Electric

Electricity retailer, Flick Electric provides a smartphone app called 'Choice' which provides information about New Zealand's current sources of electricity generation. An on-screen meter shows whether the emissions of New Zealand's current generation mix are considered high or low. This also provides a recommendation of whether tasks like running appliances or having showers is recommended (Figure 6). The app also shows the amount of the country's energy provided by each energy source and the impact each source has on the total emissions (Figure 7). This has potential to affect when New Zealanders use power, which helps flatten the grid and shifts the majority of consumption to the times when generation is the most environmentally friendly. This could help further the adoption of renewable electricity generation in New Zealand (Flick Electric, n.d.).

Figure 6. The Choice app's usage recommendations (Flick Electric, n.d.).

Figure 7. The Choice app's generation split section (Flick Electric, n.d.).

2.3.4 Future of the New Zealand Grid

Transpower (2016) provides an insight into the future of the New Zealand electricity system. The electricity grid in New Zealand is reasonably clean, with 80% being supplied by renewable energy sources. Strengthened climate change policies will shift some industries, such as transport, to use electricity rather than fossil fuels. To smooth out daily peaks, more distributed storage will be added near points of use and a smarter grid moves peak usage to off-peak times.

Transpower still expects the grid to be relevant in the future due to high costs involved with off-grid setups and the already highly renewable nature of the New Zealand electricity grid. Even if there is a significant uptake of distributed generation with solar, the grid would still be needed during winter. In a the most extreme scenario for Transpower, they note that distributed generation may reduce the generation requirement for the grid but still require the national grid for electricity transmission between points of generation and storage.

From the year 2020 onwards, electric vehicles are expected to be more prominent and trends like appliance automation will lead to a smoother demand profile. After 2040, the grid will contain a significant amount of distributed storage. By this time, the peak electricity demand would flatten significantly due to smarter consumption. Energy transfers between distributed storage around the country could be scheduled. Distributed storage would also lessen the impact of system failures by powering cities from storage until the fault is fixed.

2.4 Energy Monitoring Systems

The type of information provided by existing computer-based monitoring systems varies significantly. In general, many existing systems cater to one or more of the following purposes:

- To assist with maintenance.
- To help reduce energy consumption and cost.
- To optimise electricity generation.
- To assist in everyday usage.

Below, examples of existing monitoring systems are explored.

2.4.1 Kuenemann's Off-grid Monitoring System

Kuenemann's system was built to monitor their home's hybrid renewable energy system. This monitors the system at several points and uses a Raspberry Pi computer to store this information in a database. Data can be viewed from a website hosted on the Raspberry Pi and through an Android application. Through custom hardware, the system automatically switches to grid power if the battery level is too low to continue use (Kuenemann, 2014).

This system shows:

- Current energy production, consumption, and battery discharge rate in Watts.
- Total production and consumption in Watt-hours.
- Battery current and voltage.
- Battery temperature.

2.4.2 Alahmari's Off-grid Monitoring System

Alahmari (2013), produced a web based off-grid monitoring system. This collected data from a Bogart DC battery monitor, three Smart Circuit 20 devices, and six current cost EnviR monitors. A web server ran on an Intense PC in the equipment cupboard of an off-grid system allowing web clients on the local network to view the latest data from the system. A screenshot of the system is shown in Figure 8.

The system displays:

- Total daily consumption.
- Total daily generation.
- Battery voltage.
- Graphs of historic data for the above quantities.

Figure 8. Screenshot of the dashboard of Alahmari's monitoring system.

2.4.3 SunFarmer Energy X

SunFarmer is a non-profit organisation providing solar based off-grid systems in Nepal. Their monitoring system, Energy X, was created in response to prior attempts at off-grid installations in Nepal that became cost ineffective due to poor maintenance (SunFarmer, 2015). Energy X notifies engineers and building occupants of failures and allows engineers to change settings remotely, saving time and money. Energy X devices connect to CDMA or GSM phone networks to transmit data rather than requiring a constant internet connection.

The system is capable of monitoring:

- Energy production.
- Energy consumption.
- Battery temperature.
- Overall battery health.

2.4.4 Enphase Enlighten

Enphase's Enlighten is a cloud-based monitoring system for Enphase's range of products (Enphase, n.d.). As mentioned in section 2.2.4, in the Enphase Home Energy system, data from the electricity system is sent using an Enphase Envoy device. Enlighten also provides tools for management of a fleet of solar based systems or for solar installers. System owners can use a web based interface or the MyEnlighten smartphone app. Figure 9 shows a screenshot of the MyEnlighten web interface.

The Enlighten system shows:

- The overall health status of the system.
- Graphs and totals of energy generated over a period.
- A birds-eye-view of the solar installation with information about the performance of each panel.

Figure 9. The Enphase Enlighten dashboard.

This helps users optimise their electricity generation with panel performance information and reduce their energy consumption with usage totals. While there is no information available on the Enphase website, it is assumed Enlighten would be capable of monitoring the state of an Enphase AC Battery system upon this product's release.

2.4.5 Automated Energy Management

The system by Banerjee, Rollins, & Moran (2011) provides home automation techniques for better utilisation of generated energy. This stems from the realisation that reactive techniques are not enough to prevent critical battery situations, and that a user of an off-grid home should be constantly aware of how they can reduce consumption. Technologies like Google PowerMeter give a user access to raw data but do not help the user make proactive energy use decisions.

Typical energy use patterns that apply to grid attached systems like running appliances throughout off-peak times in the night are often not suited for off-grid systems. Instead, appliances should be used throughout the day to make use of incoming electricity and high battery levels. A fixed strategy for energy management also cannot work due to variation between seasons, weeks and days.

This system can provide early warnings when the battery level is predicted to be low, allowing users to reduce energy consumption. It suggests task rescheduling of appliances at times of peak generation or when the battery will be full to best utilise harvested energy. The system can also provide energy conservation recommendations. This is based on the users 'risk' level and their preference of maximising energy usage to make the most of generation, or to minimise usage to sell more energy back to the grid in a grid-tied home. A user's 'risk' level is the range of adjustment to their energy usage that they are comfortable with making.

Monitoring was performed by capturing data from the inverter every minute and using networked WattsUp meters to monitor appliances. The system consists of a 'profiler' which aggregates data collected while a 'scheduler' determines recommendations. A mobile application displays these recommendations and allows control of some devices. The system was installed in an off-grid home, but the researchers hoped the resulting recommendations and techniques would be useful for grid-tied or standard grid homes too. This work found some times where rescheduling usage would have helped conserve energy.

2.5 Determining the State of Charge

The state of charge of a battery is very important to make full use of a battery and maintain its health. Methods of state of charge determination have improved over time, becoming more advanced and accurate. The B3SOC algorithm (Apperley M. , 2016) provides an estimate of the state of charge of a battery in continuous use and without relying on its chemistry.

2.5.1 Importance of State of Charge

A state of charge (SoC) system with poor reliability may result in users recharging a battery more frequently than necessary. This will wear-out the battery quicker, so having an accurate method of determining the SoC can extend its lifetime. This also means batteries do not need to be over-engineered and smaller batteries may be able to be used if the user can accurately determine the state. When SoC is used to control charging, a system with poor reliability can result in undercharging or overcharging the battery, leading to earlier wear-out (Pop, Bergveld, Notten, & Regtien, 2005).

2.5.2 Historical Methods

Methods of battery state of charge determination have improved over time. Many of the initial approaches were based on the battery voltage, followed by open circuit voltage based methods. Coulomb counting methods in combination with prior approaches can give an accurate estimate and allow for factors that affect performance like battery ageing.

2.5.2.1 Voltage Based Methods

Many of the early methods of SoC determination were very basic. An example is the battery ‘fuel’ level gauges by Curtis Instruments in 1963. An example of a method of SoC determination used at this time is “predicting the remaining capacity of a battery by measuring the elapsed time period since the loaded voltage dropped below a certain value” (Pop, Bergveld, Notten, & Regtien, 2005). York et al’s 1974 method uses two voltage levels stored in the system. The system is determined to be in one of three states; greater than the first voltage, between the voltages, or lower than the second voltage. A threshold circuit monitors the magnitude and duration of the voltage reduction and this is used to produce an indication of the SoC (Pop, Bergveld, Notten, & Regtien, 2005).

2.5.2.2 Open Circuit Voltage Based Methods

In 1975, Christianson et al created an open-circuit voltage (OCV) based method of SoC determination. OCV is defined as: $OCV = V_{term} + I\ R$. Here, V_{term} is the battery terminal voltage, I is the actual battery current, and R is the internal resistance. When the battery is at equilibrium (no charging or discharging) and after allowing the battery to relax, $OCV = V_{term}$. OCV is directly proportional to the battery SoC.

In 1984, Peled developed a method for determining the SoC of lithium-ion batteries. This requires measuring the temperature of the battery, applying a current load on the battery, then measuring the OCV after a short rest period. The method determines the residual state of charge based on the temperature and OCV using a look-up table (United States of America Patent No. 4725784 A, 1984).

2.5.2.3 Coulomb Counting

Coulomb counting is a technique that measures the current flowing to or from the battery from external circuits. Current is measured in small intervals and is integrated to accumulate the total energy flowing in and out of the battery over a period of time (Woodbank Communications Ltd, n.d.). Aylor (1992) provides a method to determine the SoC of lead-acid batteries using a combination of the OCV method and coulomb counting. The error that accumulates from coulomb-counting is corrected with an OCV reading when the battery is rested. A method by Seyfang (1988) uses coulomb counting and makes use of temperature, charge efficiency, self-discharge and aging. When the battery is fully discharged or charged, parameters like the conversion efficiency of the battery are learned to allow for aging. A similar approach is used by Berrgveld et al in 2000 for a lithium battery. This method considers overpotential during discharge which depends on several factors including the discharge current, SoC, age and temperature (Pop, Bergveld, Notten, & Regtien, 2005).

2.5.3 B3SOC Algorithm

Apperley & Alahmari (2013) developed the Black Box Battery State of Charge (B3SOC) algorithm. B3SOC does not rely on inaccurate measures of state of charge, like the battery voltage or methods which rely on the chemistry of the battery. Instead, B3SOC uses coulomb counting methods coupled with known events for re-calibration to produce an estimate. The algorithm is designed for continuous use situations, like those in renewable energy systems.

The algorithm treats a battery as a container that can store some upper limit of energy, and applies charge efficiency when energy is stored in the battery (consuming electricity is assumed to have no energy loss). The voltage and current of the battery are used to determine the amount of energy added or removed from the battery since the last reading. The state of charge is the ratio of the amount of energy in the battery and its capacity.

The state of charge is affected by the charge efficiency and the battery capacity. The algorithm is dependent on having appropriate techniques for recalibrating these values relatively frequently.

2.5.3.1 Charge efficiency calibration

A low voltage disconnect (LVD) is initiated by an inverter to prevent the battery level getting low enough that it could damage the battery. A LVD occurs when the battery voltage is below a defined limit for a period of time. An LVD provides an opportunity to re-assess the zero charge level, and recalibrate those parameters which determine charge level - particularly the charge efficiency. If the state of charge is greater than 5% and an LVD event occurs, the charge efficiency is too high and should be re-calculated. If the state of charge becomes less than -5% but no LVD event occurs, the charge efficiency is too low and should be re-calculated.

Charge efficiency is calculated as the ratio of the amount of energy flowing into the battery in the last seven days and the amount of energy out over this time (taking into account the amount of energy already in the battery seven days ago).

2.5.3.2 Battery capacity calibration

If the amount of energy in the battery exceeds its apparent capacity (i.e: SoC > 100%), the battery capacity value will be recalibrated upwards to match its current amount of energy. However, downwards recalibration is more difficult, since the only indication of the battery becoming fully charged is a charger back-off event, when the charger thinks the battery is nearing full capacity. These events are unreliable, so instead, the capacity value is decremented by 2% daily allowing recalibrations upwards to occur if the capacity is too low.

3 Analysis

This section covers the objectives of the application and introduces the off-grid system used to evaluate the developed application. The components of the application and its key features are identified.

3.1 Objectives of the Application

The project sets out to fulfil the following requirements:

- The application should be compatible with as many renewable energy systems as possible. This ensures the application can be useful to others after the completion of the project.
- The application should make use of cloud computing resources. This ensures the application and data processing is not limited by the computing resources available on-site.
- The application should make use of the B3SOC state of charge algorithm due to its high accuracy and independence from the chemistry of the batteries.
- The application should be able to be accessed through the internet to allow access and control from anywhere.
- The application should be secure. All data transport should be encrypted and authentication should be required to access data and control physical hardware.

3.2 Field Off-grid Setup

The application was deployed in a real off-grid house throughout development. The house, known as Pauaeke, has an off-grid system consisting of:

- A six panel solar PV array.
- A small wind turbine.
- A 230 Volt AC petrol generator.
- A 24 Volt deep-cycle battery set.
- An inverter/charger with support for low voltage disconnect (LVD).
- A solar controller.

Pauaeke was used in the work by Alahmari in 2013 as described in section 2.4.2. As a result, the monitoring equipment was still avaliable for use in this project. Figure 10 shows how the components connect in Pauaeke and the points of monitoring.

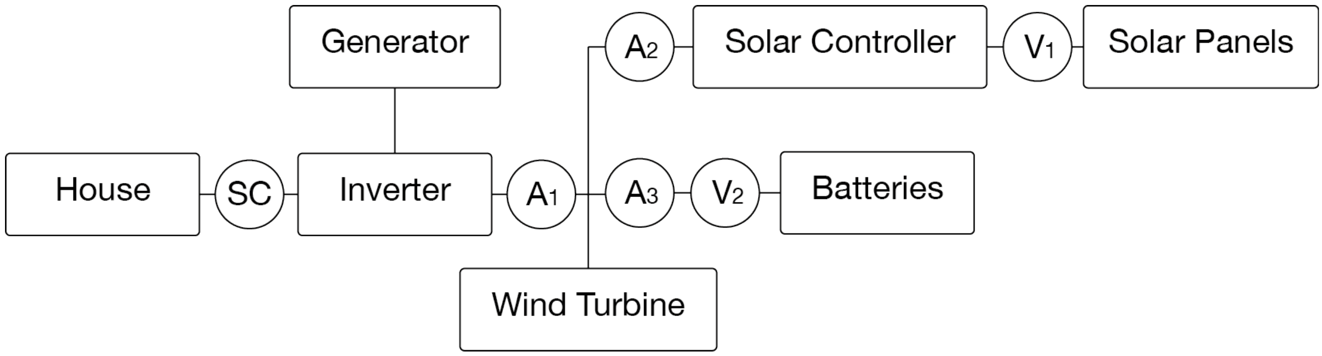


Figure 10. A diagram of the Pauaeke setup with its monitoring points.

3.2.1 PentaMetric DC Monitor

The Bogart Engineering PentaMetric PM-5000-U is capable of monitoring two DC voltages and three DC currents. The device can be interfaced with a computer through USB (altE Store, n.d.)

In the Puaeke setup (Figure 10), the PentaMetric device is connected as follows:

- **System load current** (A_1)
  The current flow from the system to the inverter. This is negative when the system is providing energy to the inverter but positive if the inverter is charging the system.
- **Solar current** (A_2)
  The current flow from the solar controller to the battery. This is positive or zero. Any negative values should be ignored as sensor error.
- **Solar voltage** (V_1)
  The voltage of the input to the solar controller. This is not used by the monitoring application.
- **Battery current** (A_3)
  The current flow to the battery. This is a positive value if the battery is charging.
- **Battery voltage** (V_2)
  The voltage across the terminals of the battery. This can be used as an indication of a battery empty event.

3.2.2 Smart Circuit AC monitor

The WattsUp Smart Circuit 20 (labelled as SC in Figure 10) can monitor the AC electricity usage of a load connected to it. Among other readings, this unit can determine (WattsUp Meters, n.d.):

- Power.
- Current.
- Voltage.
- Power factor.
- Frequency.

In the Puaeke setup, the Smart Circuit device is connected to the house load from the inverter. This allows it to determine the electricity consumption of the building. If the power to the building is off (e.g.: if an LVD event has occurred), the Smart Circuit is off so the computer will get no response from the device.

3.2.3 Intense PC Computer

The computer used is a FitPC Intense PC[1] which is connected to the internet via Ethernet. Because the off-grid site may lose power, this computer is setup with a FitUptime uninterruptable power supply[2] to ensure data collection continues. This data must be stored until the internet connection is restored with the house power.

3.2.4 Determining the wind turbine current

As above, the PentaMetric monitor only supports three currents and two voltages. As a result, the wind current does not have a dedicated sensor reading so must be determined through other means. Wind current can be determined as below (Apperley & Alahmari, 2013):

Wind current = Battery current - Load current - Solar current

3.2.5 Laboratory Setup

A laboratory setup remains from the work by Alahmari in 2013. This contains the same sensors as the field setup and an Intense PC computer. This was used during the development of the Bridge software to interact with the same type of sensors as in the real setup. Because the field Intense PC had been pre-configured to work in the laboratory, setting up the bridge in the field was simple. The

[1] FitPC Intense PC computer: http://www.fit-pc.com/web/products/intense-pc/
[2] FitUptime uninterruptable power supply: http://www.tinygreenpc.com/fit-uptime.html

laboratory Intense PC was also used to test experimental versions of the bridge software to keep the field setup gathering data without interruption.

3.3 Application Components

As outlined by the objectives in section 3.1, the application should use cloud computing and allow access to data from anywhere. It should also be compatible with as many renewable energy systems as possible, so should use the least equipment possible. By performing data processing in the cloud, minimal computing resources are required on-site, reducing power usage and making the system cheap and easy to install.

For this reason, from an early stage of the project it was clear the solution would be split amongst three pieces of software:

- The Bridge.
- The Backend.
- The Frontend.

Figure 11 shows a diagram of the components of the application.

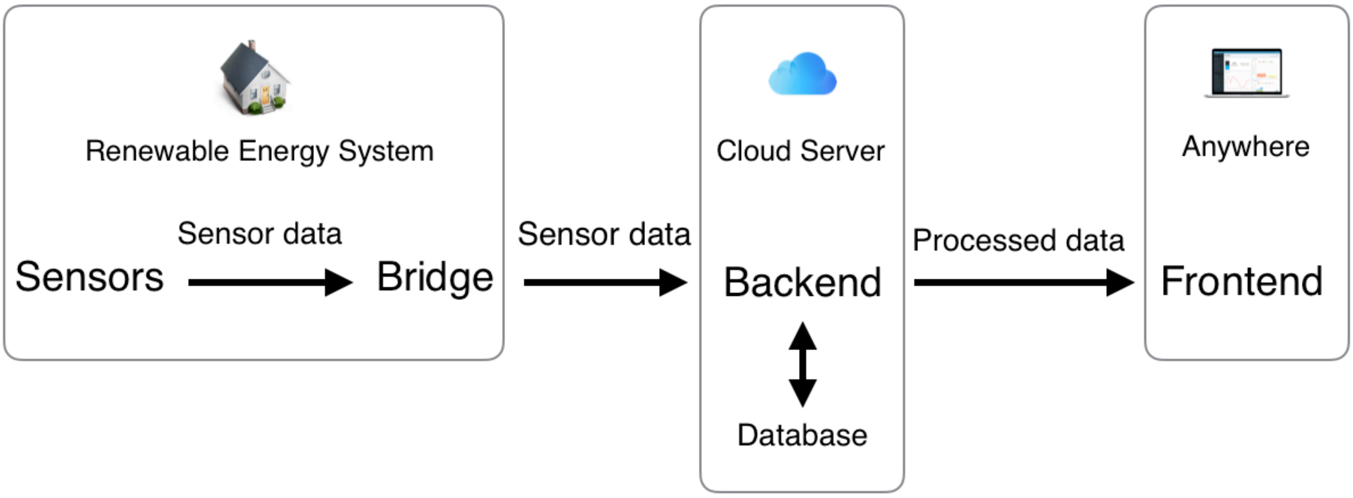


Figure 11. The components of the system.

3.3.1 Bridge

The bridge is software that collects data from the monitoring equipment and transmits it to the backend. The bridge software runs on a computer at the renewable energy system where it 'bridges' the gap between the sensors and the backend. This runs on a computer at the renewable energy system.

This software is responsible for:

- Communicating with sensor devices.
- Getting the latest data for each sensor at regular intervals.
- Sending batches of data to the backend at regular intervals.
- Handling connection or backend failure without losing data and re-sending data when possible.

3.3.2 Backend

The backend is the software that receives, stores and processes sensor data. The backend is located on a cloud server and is accessible through the internet. It makes these readings and other processed information available via an API.

This software is responsible for:

- Receiving data from bridge clients.
- Processing sensor data.
- Providing authenticated access to information.
- Storing data and exported files.

3.3.3 Frontend

The frontend is the user interface to display the status of the renewable energy system. This is accessed from a user’s internet connected device, such as a smartphone or laptop.

This software is responsible for:

- Providing an interface for the user to change settings for a building.
- Displaying the current state of the renewable energy system.
- Allowing the user to query for raw data stored in the backend.
- Supporting mobile and desktop screen sizes.

3.4 Conceptual Design

The application consists of six key features:

- Getting sensor data into the cloud.
- Displaying basic sensor data in the cloud.
- Processing sensor readings and finding battery state of charge.
- Providing information about energy flows within the system.
- Making estimates about the future state of the system.
- Automatically powering on and off the generator.

Each of these features is explained in more detail below.

3.4.1 Getting data into the cloud

As sensor data is processed in the cloud, the first required feature is to transmit data to the backend. The bridge software should support fetching data from sensor devices at a regular interval and sending it to the backend.

3.4.2 Displaying basic sensor data in the cloud

A user should be able to securely login to the application from anywhere with their internet connected device. They should be able to access the raw readings and sort or filter this information. The application should support exporting data to a CSV file to allow for manual data processing.

3.4.3 Processing readings and finding battery state of charge

The application should determine the state of charge of the battery using uploaded reading data. The user should be able to access the current state of charge and a graph of the historical state of charge of the battery.

3.4.4 Providing information about energy flows within the system

The application should calculate the total hourly/daily consumption and generation. This is used to display graphs of the consumption and generation throughout the day. The application should show an instantaneous view of the current energy flow in and out of the system.

3.4.5 Future battery state estimation

The application should determine a prediction pattern of each energy source and the building's consumption. This is used to determine the average consumption for a particular hour. The application can use this to calculate the state of the system over the next 24 hours and determine the next battery full or empty event.

3.4.6 Automatically powering on and off the generator

The bridge should support controlling the generator. This would automatically turn the generator on when the battery voltage becomes sufficiently low and use default rules to turn the generator off. The backend would use the future state to determine the best time to turn the generator on and off, then send a command to the bridge to control the generator. The frontend would show when the generator is predicted to turn on and off, and allow manual generator control through the internet.

4 Implementation

The application's architecture is based on three parts (as shown in Figure 11); the bridge, the backend and the frontend. The first stage of development involved getting data into the cloud. The bridge communicates with sensor devices and sends it to the backend at regular intervals. The next stage was to show sensor data stored in the cloud through the web-based frontend. This allows filtering raw data and exporting data to a CSV file. The B42SOC algorithm (described in section 4.5.2) was implemented to provide the state of charge of the system. This is applied to data as it is uploaded, or by re-processing the full set of readings. To provide information about the energy flows in the system, energy sources are identified through current sensors for the source, the charger or the 'other' energy source. The hourly and daily energy flow is totalled for each energy source and the house consumption. Using this data, the application displays graphs of the instantaneous or daily energy flow. The application estimates future energy flows using prediction patterns and prediction multipliers, allowing the next battery event to be determined and displayed in the frontend.

4.1 Development Process

The system was deployed at Pauaeke throughout the development of the application. This enabled the occupants of this house to provide continuous feedback on the usefulness and accuracy of the application. The development was planned so that new functionality could be deployed in the field as quick as possible to allow this feedback to impact future development.

However, to allow this to be a reasonable evaluation of the application, the software used in the field needed to be as stable as possible. To achieve this, new versions of the bridge were tested at the laboratory setup and a local development environment was used for the backend rather than the actual server. The local development environment used a copy of the real database to allow realistic data processing.

BitBucket[3] was used to provide Git source control throughout the project. New versions of the system were deployed by logging into the server and running a script that pulls the latest code, re-builds the frontend and restarts the backend.

[3] BitBucket: http://bitbucket.com/

4.2 Architecture

As shown in Figure 11, the application consists of three software components; the bridge, the backend, and the frontend.

4.2.1 Bridge

The bridge is software that collects data from the monitoring equipment transmits it to the backend.

4.2.1.1 Execution Environment

The bridge is designed to be run on a computer at the renewable energy system. This computer is always on and does not need to be connected to a screen or interacted with physically in any way. The computer is plugged into sensor equipment so is located with the other equipment that runs the system.

The computer (in both the development and field setups) ran the Linux based Ubuntu Server[4] operating system. This was used as it is a free and lightweight operating system. Because it is completely command line based, all functions of the computer can be controlled via remote secure shell (SSH) access. Because the field setup used a residential internet connection, it is assumed that the external IP address is dynamic and that port-forwarding for remote access may not by allowed by the internet service provider. As a work around to this, a reverse SSH tunnel is setup and maintained using the autossh[5] program. This is an outgoing connection from the field computer to the server so the field computer does not need to accept incoming connections from the internet. Remote access of the bridge is obtained by logging into the server and connecting to the port specified for the remote SSH tunnel.

4.2.1.2 Software Environment

The Bridge is written in Node.js, a JavaScript runtime environment (Node.js, n.d.). This makes the software quick and easy to write and matches the JavaScript language used for all the other software components of the project. In JavaScript, it is very easy to do tasks asynchronously without the programmer needing to manage threads themselves. This is essential for the bridge, which constantly performs asynchronous tasks like waiting until the next sensor reading, getting sensor readings, or sending data to the API.

---

[4] Ubuntu Server. Version 14.04 was used: http://www.ubuntu.com/download/server
[5] Using autossh: https://www.everythingcli.org/ssh-tunnelling-for-fun-and-profit-autossh/

A software tool called npm (Node Package Manager)[6] is used to manage the software dependencies of the Bridge. The pm2[7] software is used to ensure the Bridge software runs when the computer starts and is restarted if it crashes.

4.2.2 Backend

The backend is the software that receives, stores and processes sensor data. It makes these readings and other processed information available via an API.

4.2.2.1 Execution Environment

The backend must be run on a server that is publically accessible from the internet and has a fast and reliable internet connection. This is because the server must be accessible by the Bridge and Frontend clients. Unlike the Bridge, the server should be a reasonably high-spec computer in order to handle data processing and have sufficient storage for the database.

The server can be physically located anywhere that meets these requirements. Throughout the duration of the project, a server was provided by the University of Waikato's Faculty of Computing and Mathematical Science.

4.2.2.2 Software Environment

The backend is written in NodeJS using the Loopback framework[8]. The Loopback framework saves development time because once models and relationships are defined, Loopback handles the entire database interaction layer and creates all standard API endpoints. Models in Loopback can then be extended to add additional API endpoints and custom handlers for events like data queries or saves. Loopback provides an abstraction layer for the database which means that changing the database vendor at a later point is trivial. The system currently uses MongoDB, a NoSQL based database.

The same as for the bridge, the pm2 software is used to ensure the Backend runs when the server starts and is restarted if it crashes. The npm software is used to manage dependences for the backend.

---

[6] npm: https://www.npmjs.com/
[7] pm2: https://github.com/Unitech/pm2
[8] Loopback Framework: https://loopback.io/

4.2.3 Frontend

The frontend is the user interface that displays the status of the system.

4.2.3.1 Execution Environment

The frontend is a set of HTML, JavaScript and CSS files. These are hosted on the server that runs the backend but these are rendered and executed on the user's computer. As a result, code should be written with performance in mind for devices with low computational power such as smartphones, and data usage should be kept low for users on slow or limited internet connections.

4.2.3.2 Software Environment

The frontend is a web-app written in AngularJS. AngularJS is a model-view-controller framework which manages data bindings between the HTML template and variables in JavaScript (Google, n.d.). An example of the power of Angular is that JavaScript variables can be included in the HTML template and automatically updated on the page when their value changes.

The AdminLTE website template was used as a base for the user interface. AdminLTE extends the Bootstrap[9] user interface framework (Almsaeed, n.d.) which provides a grid system for responsive design and many useful HTML components (Chouhan, 2013). Figure 12 shows a screenshot of the AdminLTE template. All charts are displayed with Chart.js[10] and using the angular-chart.js[11] plugin for easier use with AngularJS.

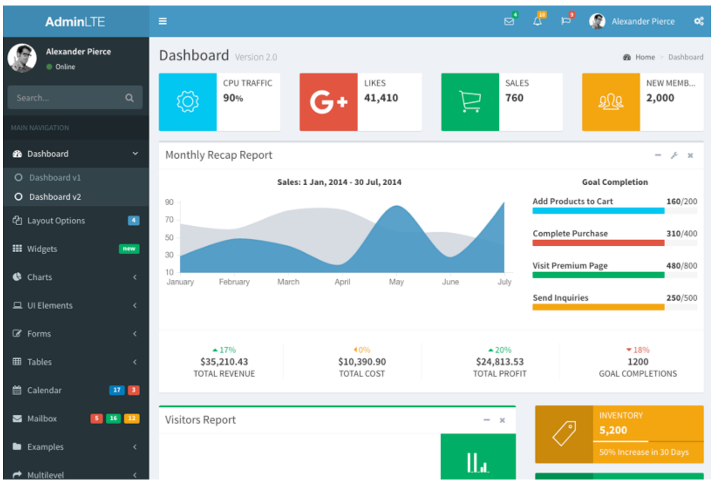


Figure 12. A screenshot of the AdminLTE template.

---

[9] Bootstrap: http://getbootstrap.com/
[10] Chart.js: http://www.chartjs.org/
[11] angular-chart.js: https://github.com/jtblin/angular-chart.js

The Loopback framework includes an AngularJS SDK[12] which generates all API interaction code needed in the frontend. The Grunt task runner[13] is used to run the frontend in the development environment or to build the frontend for production use on the web server. This combines and minifies CSS and JavaScript resources to save bandwidth for end users.

4.3 Getting data into the cloud

Getting data into the cloud involved development of the bridge software to communicate with sensor devices. The bridge collects sensor data and sends it to the backend at a regular interval.

4.3.1 Communication with sensor devices

The bridge communicates with sensor devices to get sensor data. The bridge is designed to be expandable, so a device module is written for each sensor device. These consist of a JavaScript file that implements a device API and another that implements the lower level drivers for each device. This standard format means new sensor drivers can be written in the future without needing to change the rest of the bridge. In the bridge, a sensor represents a single point of measurement while a sensor device is a piece of hardware that provides sensor readings. Sensor devices only exist in the bridge and have no representation in the backend.

4.3.1.1 Device Configuration

Each device has an entry in the bridge configuration file. This defines:

- The name to use for the device, mainly for logging purposes, e.g.: “DC monitor”.
- The type of device. This defines the device module to use.
- The device path in the file system. This is used to target specific USB ports through `/dev/serial/by-path/[port identifier]`.
- An array of sensors with their backend ID and other fields that help the fetch task identify the sensor.

4.3.1.2 Device APIs

The device API is constructed with the device configuration and a log context (so the location of log messages can be identified). The API contains a single fetch function which returns a Promise object that will resolve with an array of sensor IDs and values. A Promise object is a reference to a

---

[12] The Loopback AngularJS SDK: https://docs.strongloop.com/display/public/LB/AngularJS+JavaScript+SDK
[13] Grunt task runner: http://gruntjs.com/

pending task and the `Promise.then` function is used to provide a callback function upon the success or failure of the task.

Code from Alahmari's work in 2013 was a useful resource throughout the development of the fetch tasks as this included the sequences of sensor commands to run to get the desired data.

PentaMetric DC Monitor:

The configuration defines the type of sensor (voltage or current) and the sensor number. To request a sensor reading from the PentaMetric device, a command is sent which retrieves the particular sensor value. The fetch function requests a value from each configured sensor and returns the combined Promise.

Smart Circuit AC Monitor:

The configuration defines the type of reading for each sensor, for example, "voltage" or "power". The Smart Circuit is capable of storing values to an internal memory where it can be bulk downloaded. However, for the bridge this needs to be an instantaneous reading. To achieve this, the Smart Circuit fetch task clears the sensors memory, waits until it is populated with a value, then reads the sensors memory to get a near instant reading.

4.3.1.3 Low level sensor drivers

Sensor drivers were written to provide a high level, Promise based API to access physical hardware. This allows fetch tasks and test code to be written without concern of the device state.

Drivers were written for the PentaMetric and Smart Circuit devices. Both:

- Provide access to hardware commands with functions like `getMemory` or `getVoltageReading`. These handle all response parsing so provide numerical or object based results.
- Internally queue commands so the device has only one request to process at a time and the responses are handled by the correct functions.
- Handle opening and closing device connections.

Code from Alahmari's work in 2013 was a useful resource throughout the development of the sensor drivers as this contained parsing code for sensor responses.

4.3.2 Sending readings to the backend

The bridge requests readings from all sensors at a regular interval defined in the bridge configuration file (throughout the project, ten seconds was used). Sensor data is structured in batches based on the time interval they were retrieved during. A reading consists of the timestamp and a key-value object of sensor IDs and their retrieved sensor value. If a fetch task for a sensor device fails, it is ignored and re-attempted at the next fetch interval.

Readings are sent to the backend at a regular interval defined in the backend configuration file (throughout the project, a minute was used). If the readings failed to send then they are re-added to the queue to be re-sent at the next interval. A maximum of 50 readings will be sent at a time to prevent overloading the backend.

4.4 Displaying basic sensor data in the cloud

Displaying basic sensor data in the cloud required development of the backend to support key structures like buildings, bridges, sensors and readings. Bridge and people authentication was setup in the backend to ensure privacy is maintained. The ability to explore sensor readings and export readings to a CSV file was added to the frontend.

4.4.1 Buildings, Bridges, Sensors and Readings

The backend was built with future expansion in mind. As a result, the backend supports multiple buildings. Buildings can support multiple bridges to allow for future bridge types, for example, sensor hardware that can directly connect to the internet without a connection to a computer. Bridges have an associated set of Sensors. Sensors have names to help identify them such as "Solar Current". Sensors also have a type, such as "acvoltage" which can affect how the sensor is displayed to the user. Readings sent to the backend are associated with a bridge. When a reading is received, the sensor IDs provided are checked to ensure they are valid sensors for the bridge.

4.4.2 Bridge Authentication

As part of the application's objective of being secure, Bridges should be authenticated to prevent malicious users uploading readings. Bridges do not authenticate with a person's username and password. This is because users should be able to change their password without stopping the bridge from working and malicious users could obtain access to the bridge configuration file containing the credentials. Instead, a bridge secret is randomly generated upon the creation of a bridge in the backend. This secret is added to the bridge configuration and is sent as part of the URL when the bridge uploads readings. The combination of the bridge ID and bridge secret authenticate and authorise requests from bridges. This ensures they can only upload readings for their bridge ID. If a bridge client becomes compromised, the bridge's secret can be changed to prevent unauthorised access. People authentication methods cannot be used to upload readings to a bridge.

4.4.3 People Authentication

Authentication for people is performed using an email address and password. Figure 13 shows a screenshot of the system's login page. The Loopback framework generates a user authentication token upon login which must be included as a query string parameter or header to make authenticated requests. All API methods (except login and bridge related methods) require token authentication. People can be assigned as owners of a building and buildings can have multiple owners. In addition to token based authentication, a user can only access content related to buildings that they own. See Appendix C - User Authorisation Rules for a full listing of building access rules. An example is that users can only access readings recorded against the bridge of a building that they own. These rules ensure the privacy of users is retained.

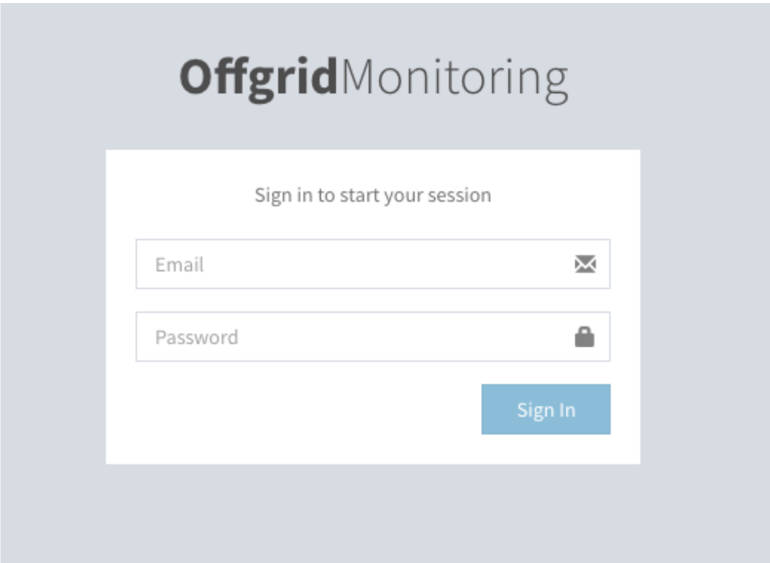
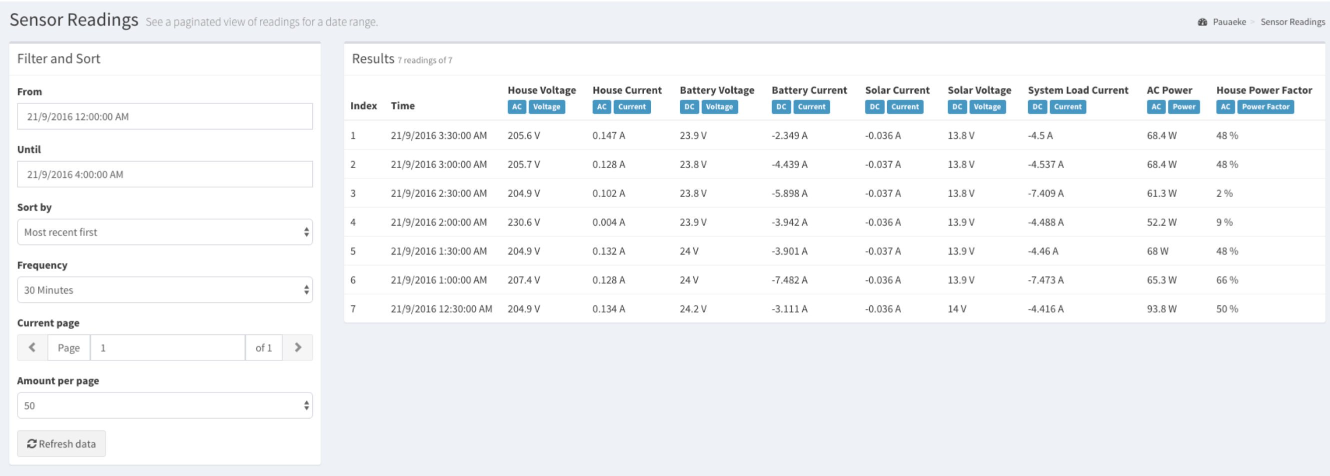


Figure 13. The login page for the frontend.

4.4.4 Sensor Readings Explorer

The system supports filtering and ordering the raw sensor readings in a paginated view. Figure 14 shows the Sensor Readings page in the system. Support for this on the backend is provided through filtering and ordering options provided with the Loopback framework. This allows:

- Filtering data to within a time range.
- Sorting the time order to 'Most recent first' or 'Oldest first'.
- Setting the frequency of readings.

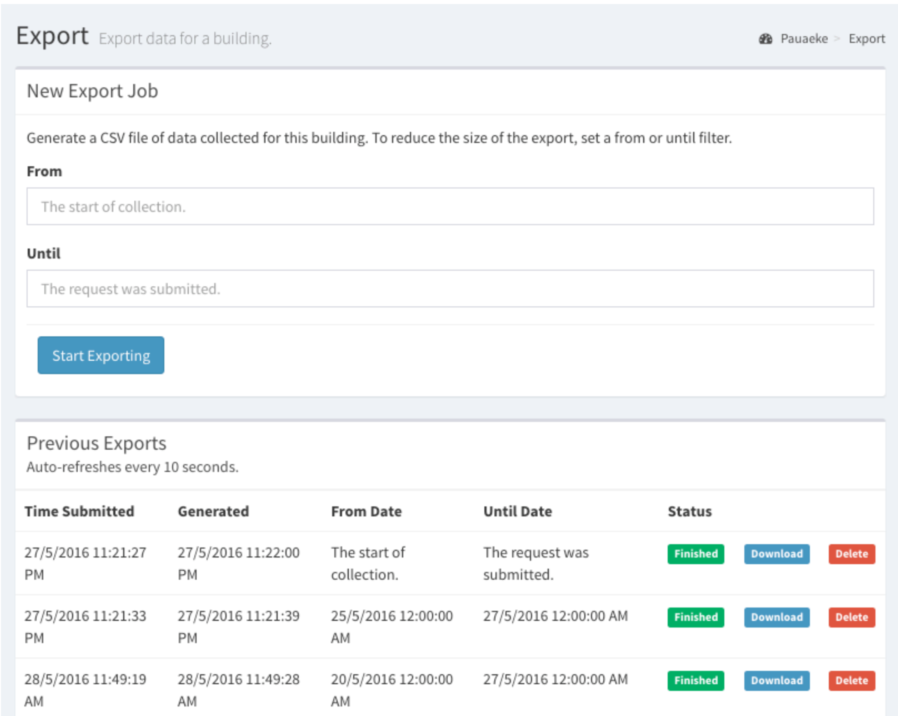


Index	Time	House Voltage (AC Voltage)	House Current (AC Current)	Battery Voltage (DC Voltage)	Battery Current (DC Current)	Solar Current (DC Current)	Solar Voltage (DC Voltage)	System Load Current (DC Current)	AC Power (AC Power)	House Power Factor (AC Power Factor)
1	21/9/2016 3:30:00 AM	205.6 V	0.147 A	23.9 V	-2.349 A	-0.036 A	13.8 V	-4.5 A	68.4 W	48 %
2	21/9/2016 3:00:00 AM	205.7 V	0.128 A	23.8 V	-4.439 A	-0.037 A	13.8 V	-4.537 A	68.4 W	48 %
3	21/9/2016 2:30:00 AM	204.9 V	0.102 A	23.8 V	-5.898 A	-0.037 A	13.8 V	-7.409 A	61.3 W	2 %
4	21/9/2016 2:00:00 AM	230.6 V	0.004 A	23.9 V	-3.942 A	-0.036 A	13.9 V	-4.488 A	52.2 W	9 %
5	21/9/2016 1:30:00 AM	204.9 V	0.132 A	24 V	-3.901 A	-0.037 A	13.9 V	-4.46 A	68 W	48 %
6	21/9/2016 1:00:00 AM	207.4 V	0.128 A	24 V	-7.482 A	-0.036 A	13.9 V	-7.473 A	65.3 W	66 %
7	21/9/2016 12:30:00 AM	204.9 V	0.134 A	24.2 V	-3.111 A	-0.036 A	14 V	-4.416 A	93.8 W	50 %

Figure 14. The reading explorer which supports filtering and ordering raw reading data.

4.4.5 CSV Export

The system allows users to export raw reading data as a CSV file. Because of the high volume of data these can contain, the user requests a CSV for a particular time range and the CSV is created and saved asynchronously. The user can see the status of the job and once it finishes they can download the file. The export page is shown in Figure 15.



Time Submitted	Generated	From Date	Until Date	Status		
27/5/2016 11:21:27 PM	27/5/2016 11:22:00 PM	The start of collection.	The request was submitted.	Finished	Download	Delete
27/5/2016 11:21:33 PM	27/5/2016 11:21:39 PM	25/5/2016 12:00:00 AM	27/5/2016 12:00:00 AM	Finished	Download	Delete
28/5/2016 11:49:19 AM	28/5/2016 11:49:28 AM	20/5/2016 12:00:00 AM	27/5/2016 12:00:00 AM	Finished	Download	Delete

Figure 15. The export page in the frontend.

A user can delete the generated CSV or keep it in the system where it can be re-downloaded. CSV's are generated using the csv-write-stream[14] package. The CSV's contain a column for each recorded sensor and a formatted timestamp (example data from a CSV export is shown in Figure 16). The CSV export functionality allows users perform their own processing on the data in addition to the processing performed by the system. This was used to provide data for the B42SOC algorithm development by Apperley (2016).

Date	Bridge ID	House Voltage (type: acvoltage, ID: 572023553916fd9b1ac7ba4d)	House Current (type: accurrent, ID: 5720235d3916fd9b1ac7ba4e)
27/04/16 14:50	572022983916fd9b1ac7ba39	230.1	2.57
27/04/16 14:51	572022983916fd9b1ac7ba39	229.9	2.553
27/04/16 14:51	572022983916fd9b1ac7ba39	229.3	2.537
27/04/16 14:51	572022983916fd9b1ac7ba39	229.2	2.507
27/04/16 14:51	572022983916fd9b1ac7ba39	229	2.503
27/04/16 14:51	572022983916fd9b1ac7ba39	228.8	2.508
27/04/16 14:52	572022983916fd9b1ac7ba39	228.8	2.641
27/04/16 14:51	572022983916fd9b1ac7ba39	228.4	2.507
27/04/16 14:52	572022983916fd9b1ac7ba39	229	2.637
27/04/16 14:52	572022983916fd9b1ac7ba39	228.9	2.673
27/04/16 14:52	572022983916fd9b1ac7ba39	228.5	2.624
27/04/16 14:52	572022983916fd9b1ac7ba39	228.8	2.627

Figure 16. An example of the format of the CSV export.

4.5 Processing readings and finding battery state of charge

The backend processes readings and determines the battery state of charge. This helps users determine how much charge is remaining in the battery and can provide additional insights, such as the effective capacity of the battery as a result of performance affecting factors. The state of the system is processed as readings are uploaded or all readings can be re-processed after a configuration or implementation change. The B42SOC algorithm (Better Black Box Battery Simplified State-of-Charge) offers improvements over B3SOC and is implemented in the system.

4.5.1 States and Processing data

In the backend, a state represents the current state of a building at a point in time. The timestamp of a state corresponds with the timestamp of the last reading processed. A state contains all variables required for state of charge calculation and any hourly/daily totals required to display energy flows (covered in section 4.5). The function `processReadingsSerially` takes a building, an array of readings, the previous reading and the start state. This processes each reading in series, calculating the state of charge, energy flows and future state as appropriate.

After a group of readings are uploaded and saved, they are run through the `processReadingsSerially` function to ensure the states are up to date. When the implementation or configuration parameters change, the change only makes a difference to future readings. To update the historical data after making a change, the system allows users to re-process

[14] csv-write-stream: https://www.npmjs.com/package/csv-write-stream

existing data. This clears all states and processes all readings in batches of 10,000. The "Data processing" section on the Configuration page allows a user to trigger a full re-process from the frontend (Figure 17). This shows the status of the last re-process job including the time taken, the number of successful readings, and the number of failed readings. While re-processing, the interface shows a loading spinner and refreshes the status until the job is complete. The data included in re-process jobs can be limited with the "Process data after" and "Process data until" configuration options.

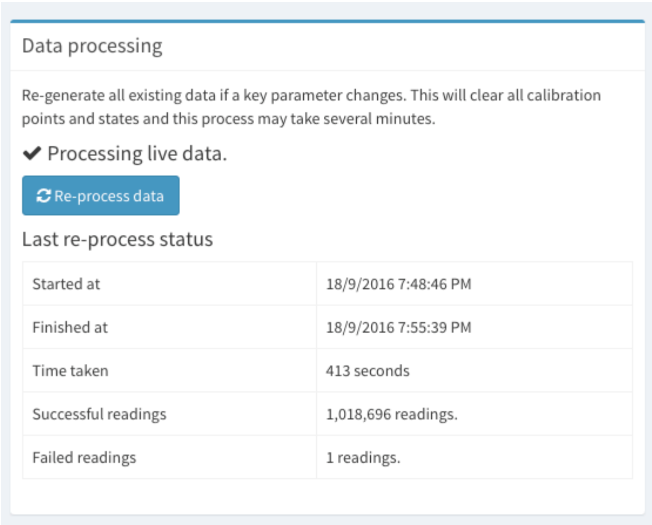
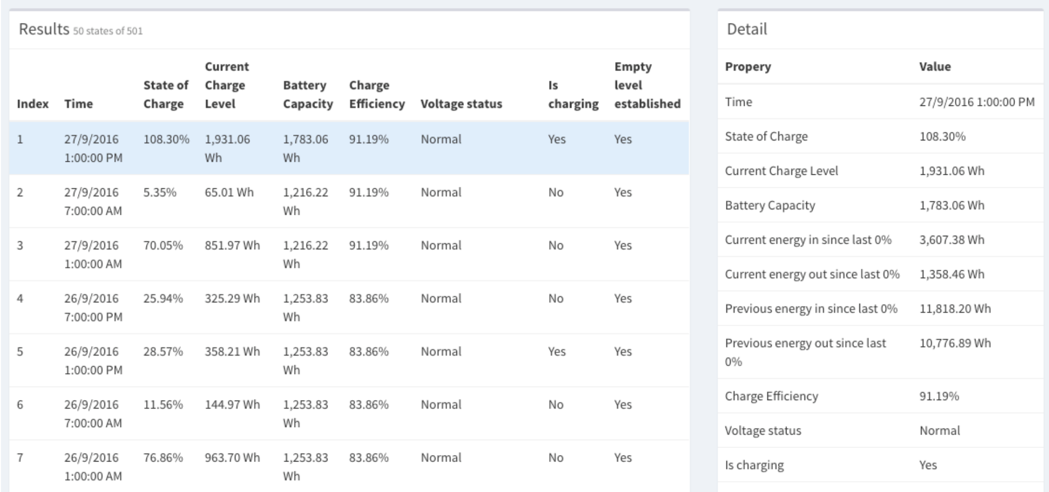


Figure 17. The "Data processing" section of the Configuration page.

A state entry is recorded at 10 minute intervals based on the reading timestamp. The current state is stored and is overwritten. A building stores the ID of its current state entry, allowing for quick retrieval of the current state of the system. The Battery State page in the frontend lets users sort and filter states in the same way as the Reading Explorer. Because there are many properties in a state entry, only a subset of parameters are shown in the list. Other parameters are be shown in the detail view to the right by hovering on a state or clicking to lock the detail view to a particular state. Figure 18 shows the list and detail views on the Battery State page.


Results 50 states of 501

Index	Time	State of Charge	Current Charge Level	Battery Capacity	Charge Efficiency	Voltage status	Is charging	Empty level established
1	27/9/2016 1:00:00 PM	108.30%	1,931.06 Wh	1,783.06 Wh	91.19%	Normal	Yes	Yes
2	27/9/2016 7:00:00 AM	5.35%	65.01 Wh	1,216.22 Wh	91.19%	Normal	No	Yes
3	27/9/2016 1:00:00 AM	70.05%	851.97 Wh	1,216.22 Wh	91.19%	Normal	No	Yes
4	26/9/2016 7:00:00 PM	25.94%	325.29 Wh	1,253.83 Wh	83.86%	Normal	No	Yes
5	26/9/2016 1:00:00 PM	28.57%	358.21 Wh	1,253.83 Wh	83.86%	Normal	Yes	Yes
6	26/9/2016 7:00:00 AM	11.56%	144.97 Wh	1,253.83 Wh	83.86%	Normal	No	Yes
7	26/9/2016 1:00:00 AM	76.86%	963.70 Wh	1,253.83 Wh	83.86%	Normal	No	Yes

Detail

Propery	Value
Time	27/9/2016 1:00:00 PM
State of Charge	108.30%
Current Charge Level	1,931.06 Wh
Battery Capacity	1,783.06 Wh
Current energy in since last 0%	3,607.38 Wh
Current energy out since last 0%	1,358.46 Wh
Previous energy in since last 0%	11,818.20 Wh
Previous energy out since last 0%	10,776.89 Wh
Charge Efficiency	91.19%
Voltage status	Normal
Is charging	Yes

Figure 18. The list and detail views of the Battery State page.

4.5.2 The B42SOC Algorithm

The Better Black-Box Battery Simplified State of Charge (B42SOC) algorithm is based on the B3SOC algorithm. This was implemented by Apperley (2016) and tested with data collected throughout the development of the application.

Relative to B3SOC, the key changes in B42SOC are:

- A preliminary phase occurs until the battery is first fully depleted. No charge efficiency re-calibrations are performed throughout this phase. If the charge level drops below zero, the charge capacity is increased as this represents capacity that was previously unseen by the algorithm's arbitrary starting point. The maximum charge level throughout the preliminary phase is recorded.
- When the battery is depleted the system leaves the preliminary phase. When this happens, the charge capacity is set to the maximum charge level seen throughout the preliminary phase.
- In the operational phase, the total energy in (before it is multiplied by charge efficiency) and out of the battery since the last charge efficiency calculation is stored. When the battery is expected to be empty but no LVD occurs or an LVD occurs when it is not expected, the charge efficiency is calculated. This is calculated as the energy out since the last calculation, divided by the energy in since the last calculation. However, each of these figures must be greater than some multiple of the charge capacity. Throughout development, this was set to eight times the charge capacity to ensure this covers about a week worth of usage. If the values since the last calculation are not large enough, the energy in and out since the second to last calculation is used. This prevents charge efficiency values from being too extreme if

they do not cover enough history. This can happen if the algorithm detects several low battery events within a few hours.

The battery is defined as being empty when:

- The inverter output is off, as this means LVD event occurred.
- Or, the battery is below the critical low voltage level (23.1V used throughout development) and the load power is less than the high power threshold (500W used throughout development) for the low voltage trigger time (50s used throughout development).

To prevent calibration events from being performed too often, events based on the state of charge allow for error with the tolerance percentage. Throughout development, a tolerance percentage of 10% was used. This means the charge capacity will only increase at 110% SoC and the battery is expected to be empty between -10% and 10% SoC.

4.5.3 Implementing B42SOC

The algorithm was manually applied on a dataset using an Excel spreadsheet. The code implementation of the algorithm was run on this same dataset to ensure it is implemented correctly. Configuration parameters were added for users to choose which sensors should be used for the sensor roles defined in the algorithm. These sensor roles are:

- The battery current sensor.
- The battery voltage sensor.
- The building power sensor.
- The load current sensor.

For a reading to be processed, each of the above sensors must have a reading except the building power sensor as this is optional. This is optional because the Smart Circuit device may have no power if the house loses power, and high load detection is not applicable in these situations.

In the frontend, the State page shows a graph of the state of charge, the current charge level, and the battery capacity (Figure 19). This helps to visualise the adjustments made by the algorithm. The Summary page shows a state of charge graph that is limited to the bounds of 0% to 100% for ease of readability. The Summary page also shows a battery gauge which displays blue if the battery is discharging, green if the battery is charging or red if the battery is below 20% (Figure 20).

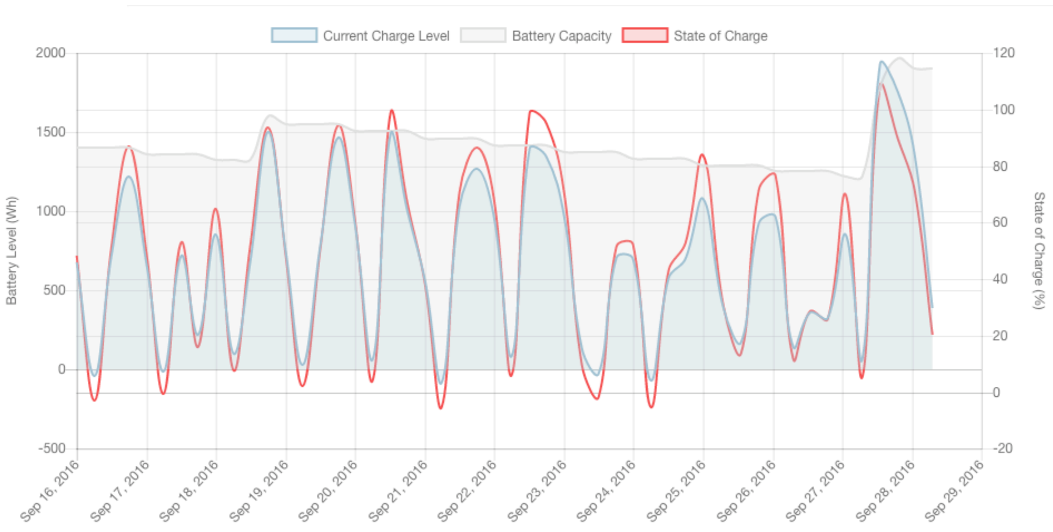


Figure 19. The graph of state of charge, charge level and battery capacity in the State page.

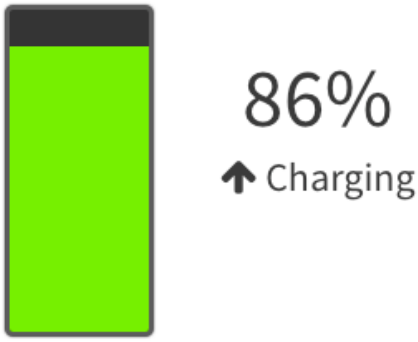


Figure 20. Battery level gauge on the Summary page.

4.5.4 Processing Parameters

The key state of charge algorithm parameters are configurable to allow the system to fit other Renewable Energy Systems and to experiment with the range of suitable parameters for the algorithm. These parameters are stored as part of the building and can be changed in the frontend through the Configuration page (Figure 21). A full listing of configurable options is shown in Appendix B - Configurable Parameters.

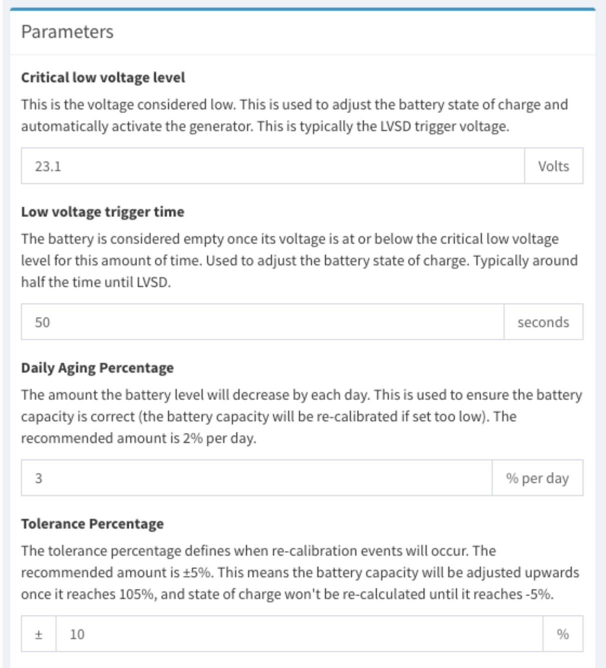


Figure 21. Part of the Parameters section on the frontend's Configuration page.

4.6 Providing information about energy flows within the system

The energy sources of the system are identified using dedicated current sensors, the charger and the ‘other’ source. Energy flow data is processed to determine hourly and daily totals for energy sources or consumption. This allows for plots of the daily energy flow which can provide additional insights about how the system is used.

4.6.1 Energy Sources

An energy source provides energy to the system. Typical energy sources are wind turbines, solar panels or generators. As part of the systems objective of working with as many Renewable Energy Systems as possible, energy sources are implemented generically. To help with recognisability, an energy source has a name, sort index, and chart colour.

Energy sources have a property that defines where they get their data from. This is either an actual current sensor (e.g.: solar current), the charger or the ‘other’ source. The charger is considered active when the inverter provides energy to the battery. This is detected when the load current is positive. When the charger is on, the inverter powers the house through the charging source. As a result, the charger power consists of the load current power plus the house consumption power. As mentioned in section 3.2.4, in Pauaeke the wind turbine has no dedicated current sensor but can be derived using the existing sensors. This is implemented generically to identify the current of all energy sources without a defined sensor as follows:

Other current = Battery current - Load current - Currents of all energy sources with sensors

Pauaeke uses the following energy source configuration:

- **Name:** Solar
  Source Current: Solar Current Sensor
- **Name:** Wind
  Source Current: Other (remaining energy generation)
- **Name:** Petrol
  Source Current: Charger (when load current is positive)

4.6.2 Energy Flow Processing

The total hourly and daily energy flow is recorded for each energy source and the house consumption. For simplicity, these totals are reset at the reading immediately after the start of a new hour or day. For example, the total hourly house consumption between 07:00:00 and 08:00:00 is shown on the state at 08:00:00 and is reset before processing the state at 08:00:10. This information allows plots of the daily energy flow to be shown. The Daily Energy Sources graph on the Battery State page shows which energy sources are used throughout the day (Figure 22). This distinguishes between renewable and non-renewable sources, showing renewable sources on the top and non-renewable sources on the bottom. This helps a user identify how often they use non-renewable sources and how much these contribute to a day's energy supply. The Daily Total Energy graph shows all energy sources and the house consumption (Figure 23). This helps identify the role of the battery when the total generation and consumption differs.

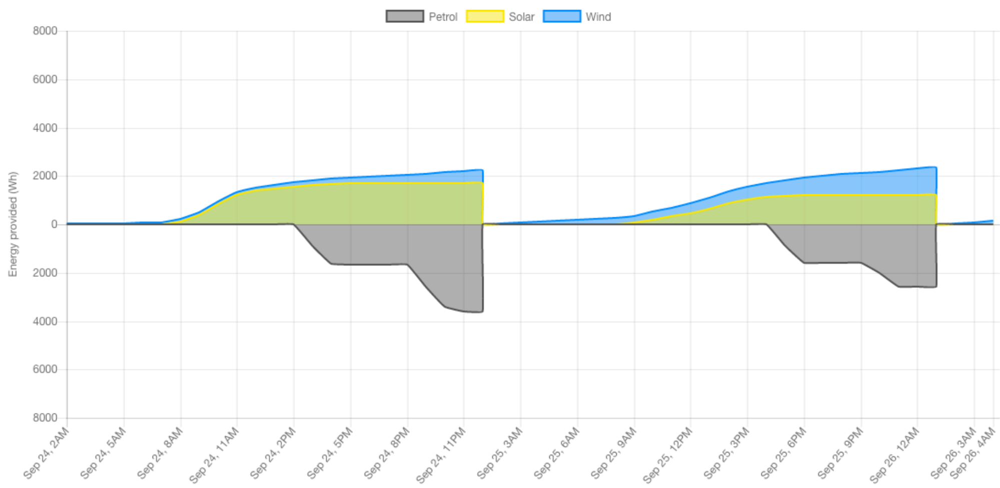


Figure 22. The Daily Energy Sources graph showing renewable and non-renewable generation.

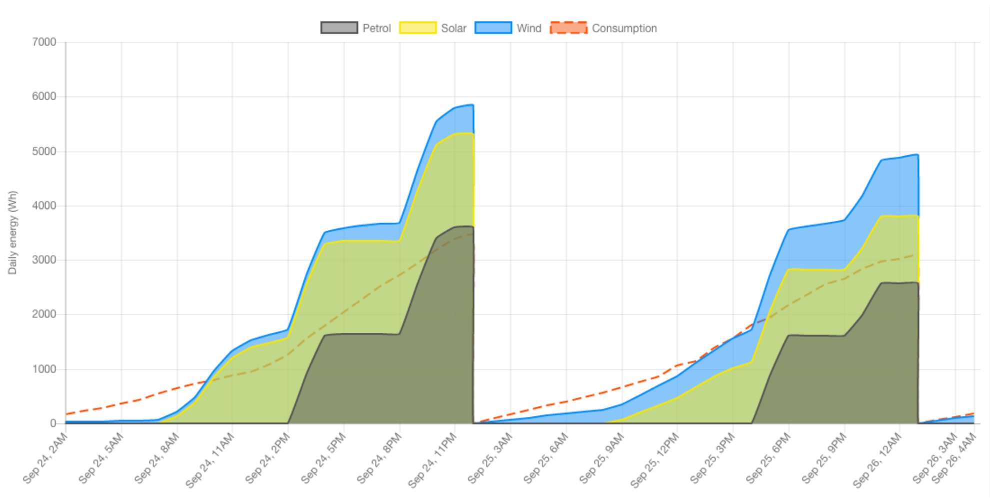


Figure 23. The Daily Total Energy graph showing consumption and generation.

An exponential average of the power of each energy source and the house consumption is calculated. This provides an indication of the present energy flow while smoothing variance between readings for a more representative view. A chart of the present energy flow in and out of the battery is shown on the Summary page (Figure 24).

Figure 24. The Energy Flow graph shown on the Summary page.

4.7 Future battery state estimation

Estimations of the future state of the system allows users to make changes to how they use the system. An example is that the user might see that the battery will be full soon so they should use additional appliances to make the most of the excess energy. Another use case is when the battery will be empty soon and they should reduce consumption and/or turn on the generator. This requires estimates of the future consumption or generation of the system through prediction patterns.

4.7.1 Energy Flow Predictions

A prediction represents the predicted hourly energy flow (in Wh) of an energy source or the house consumption. The prediction consists of a prediction pattern and a prediction multiplier. Consistent with energy sources, predictions patterns are implemented generically so an energy source or the house consumption are not tied to a particular type of prediction pattern. This allows for experimentation and future types of energy source.

4.7.1.1 Processing and use of predictions

Prediction patterns and prediction multipliers are updated when a reading that represents the end of an hour is processed. Getting a prediction for a future hour requires the index of the day of the week (0 to 6) and of the hour (0 to 23).

4.7.1.2 Prediction Pattern Types

The prediction pattern types are none, hourly, daily and weekly.

None:

The none prediction type is used to predict consumption/generation sources that have no consistent pattern so are assumed to have no energy flow in the future. This is appropriate for generators as these are directly controlled by the user and as such, are unpredictable.

Hourly:

The hourly prediction type is used to predict the energy flow of consumption/generation sources that can change within a few hours, for example, the energy generation from a wind turbine. The prediction is calculated using an exponential average of previous hours, meaning more recent hours have a greater impact on the prediction. This pattern assumes the current prediction continues for every hour in the future.

Daily:

The daily prediction type is used to predict the energy flow of consumption/generation sources that occur the same every day. This uses a rolling average with a window of 14 days to determine the pattern. This is used with energy sources like solar panels, allowing predictions to adjust with the seasons. A table and graph of the daily prediction patterns are shown on the Prediction Patterns page in the frontend. Figure 25 shows a graph of Pauaeke's predicted solar generation.

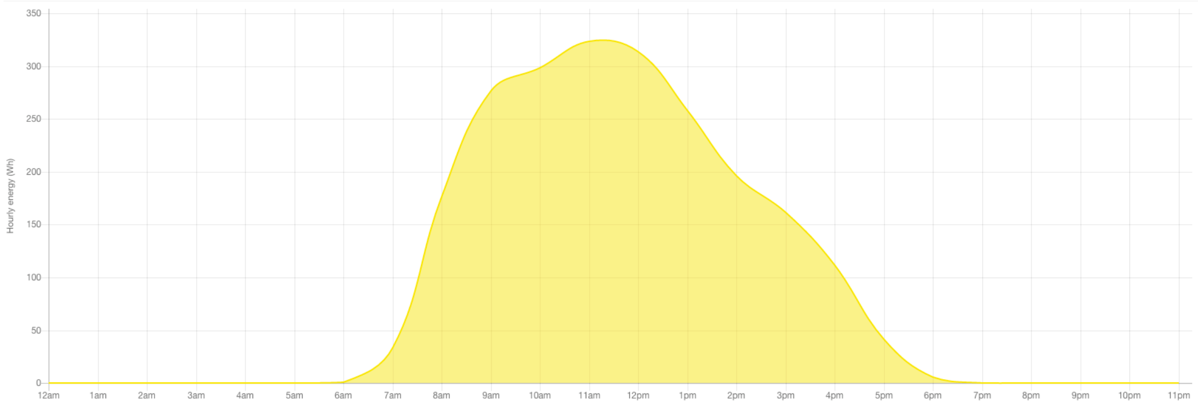


Figure 25.The solar panel generation prediction graph shown on the Prediction Patterns page.

Weekly:

The weekly prediction type is for trends that occur the same every week but not necessarily every day. The primary use case for this is a house's consumption, as this varies within a day (e.g.: higher consumption in the evening) but also on different days of the week (e.g.: weekends might consume more than weekdays). This uses a rolling average with a window of three weeks. This allows the prediction to adjust as the building occupants routines change but also allow for minor exceptions in routine. A table and graph of the weekly prediction patterns are shown on the Prediction Patterns page in the frontend. Figure 26 shows the prediction pattern of Pauaeke's consumption.

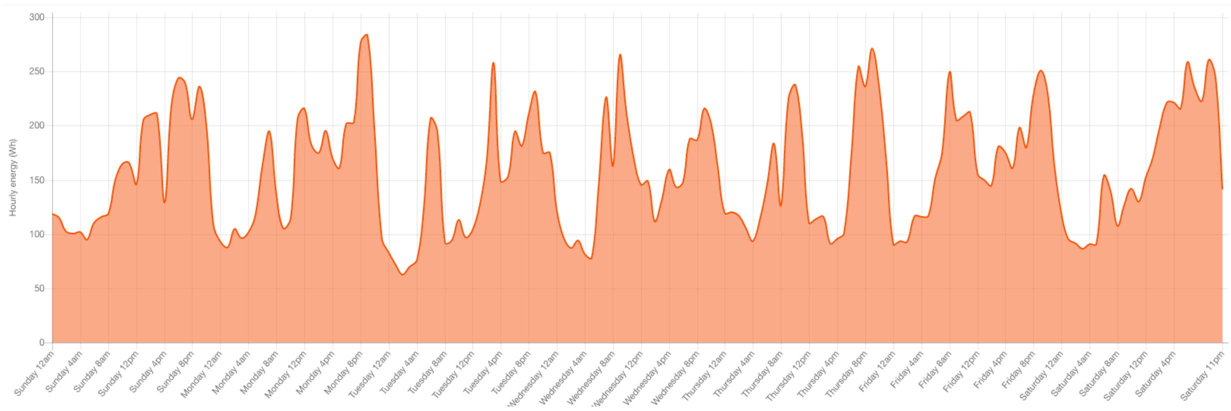


Figure 26. The consumption prediction graph shown on the Prediction Patterns page.

4.7.1.3 Use of rolling averages

Rolling averages require the full set of data over the rolling period to use for the calculation. However, if possible, the additional database queries to fetch this should be avoided to increase performance. The solution is to estimate a rolling average with the below formula:

$$newRollingAverage = \frac{(rollingPeriod - 1)\ currentAverage + newTotal}{rollingPeriod}$$

4.7.1.4 Prediction Multiplier

While the energy flow pattern can be identified, the magnitude of the actual totals may vary. An example is that the solar panel output would follow a similar pattern every day but some days may have above or below average generation. Prediction multipliers are calculated for the house consumption and each energy source. This is equal to the actual energy flow in the hour, divided by the original prediction for the hour. This is the value the previous prediction would need to be multiplied by to get the actual energy flow. The system uses this multiplier with the prediction of future hours for a more accurate estimate. To prevent the prediction multiplier from having a significant effect in extreme cases, the multiplier is capped between 0.5x and 1.5x.

4.7.2 Future States

The approach to calculating the future state is to apply the predicted generation and consumption over the next 24 hour period and determine the state of the battery at each point. The future state is re-calculated after every half hour of readings, allowing the estimate of the next battery event to be more granular. The hourly predictions are assumed to be a constant energy flow throughout the hour, so are halved for use with the half-hourly states. If the energy into the battery pushes it beyond 100% SoC, the amount of energy in from each energy source is reduced to represent generation that is wasted.

As the future states are processed, the system checks if the battery level drops below 10% or above 90% for empty or fully charged events respectively. If an event is found, its type and timestamp are recorded as part of the current state of the system. Event checking stops after one event is discovered. The Summary page shows the details of the next event, for example, “Empty in 2 hours”.

FutureState’s are objects which represent the state of the system at some time in the future. These consist of a timestamp, charge level and hourly/daily totals for consumption and each energy source. These are cleared and re-added every time the future state is calculated. The Future States page on the frontend shows a graph of the future state of charge, energy flows, and a table of FutureStates (Figure 27).

State Detail

Time	State of Charge	Current Charge Level	Daily consumption	Hourly consumption	Daily Solar charge	Daily Wind charge	Daily Petrol charge	Hourly Solar charge	Hourly Wind charge	Hourly Petrol charge
30/9/2016 2:00:00 PM	100.00%	1,969.44 Wh	436.72 Wh	131.68 Wh	1,341.22 Wh	380.30 Wh	1,876.15 Wh	-198.69 Wh	-9.46 Wh	0.00 Wh
30/9/2016 2:30:00 PM	100.00%	1,969.44 Wh	472.98 Wh	36.27 Wh	1,341.22 Wh	380.30 Wh	1,876.15 Wh	0.00 Wh	0.00 Wh	0.00 Wh
30/9/2016 3:00:00 PM	100.00%	1,969.44 Wh	509.25 Wh	72.54 Wh	1,341.22 Wh	380.30 Wh	1,876.15 Wh	0.00 Wh	0.00 Wh	0.00 Wh
30/9/2016 3:30:00 PM	100.00%	1,969.44 Wh	554.38 Wh	45.13 Wh	1,341.22 Wh	380.30 Wh	1,876.15 Wh	0.00 Wh	0.00 Wh	0.00 Wh
30/9/2016 4:00:00 PM	100.00%	1,969.44 Wh	599.51 Wh	90.26 Wh	1,341.22 Wh	380.30 Wh	1,876.15 Wh	0.00 Wh	0.00 Wh	0.00 Wh

Figure 27. The table of FutureState's on the Future State page.

4.8.1 Automatically powering on and off the generator

Automatically powering on and off the generator was unable to be implemented due to the restricted time available for the project. Although this has not yet been implemented, it would heavily depend on the accuracy of the state of charge algorithm and the predictions of the future state of the system. As a result, further improvements should be implemented in these areas before attempting automation.

5 Evaluation

Overall, the application was implemented successfully and has worked reliably. The development process has revealed the limitations of the B42SOC algorithm (Apperley M. , 2016) and identified future improvements that can be made to make the application more useful.

5.1 System Implementation

The application delivers on its goal and provides a novel way of monitoring a renewable energy system. The application has worked reliably throughout its development, collecting over 1.3 million readings from the Pauaeke system over a five month period.

The Bridge software has worked reliably and does not result in loss of data if the backend goes offline. The B42SOC algorithm was implemented in the cloud successfully. However, continued use of the application has revealed issues with this algorithm. These issues cause either the battery capacity or the charge efficiency to be recalculated too frequently, leading to inconsistent results.

The battery capacity can change often, even after several months of adjustments. An example is one case where the battery capacity increased by around 500 Wh when the battery charges as shown in Figure 28. The reason for this is that the algorithm essentially measures the battery capacity as the amount of capacity that is typically exercised in a cycle. However, the actual capacity of the battery is higher due to the small depth of discharge used to maintain the life of the battery. Varied rates of charge or discharge also affect the performance of the battery. A potential solution is to maintain information about the capacity over a longer term to better represent the full energy capacity.

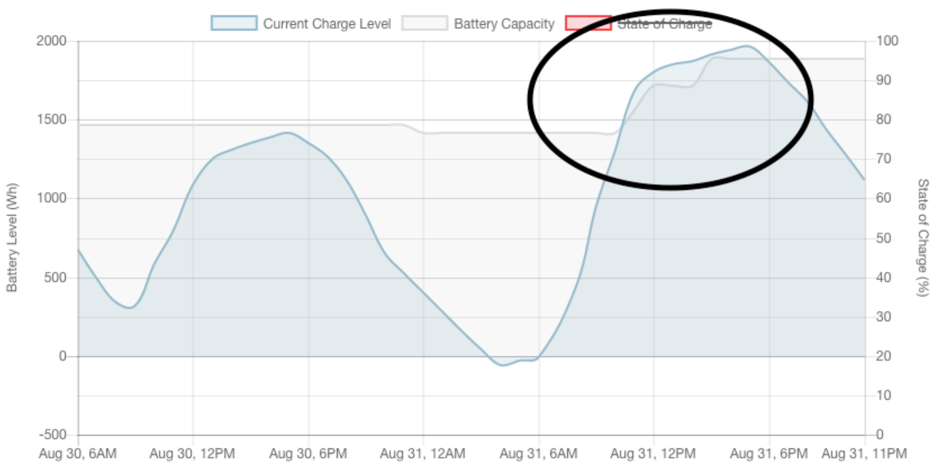


Figure 28. An example of the charge capacity being pushed up significantly.

The charge efficiency is calculated when a low battery event occurs when it is unexpected, or does not occur when it is expected. At the point of the calculation, the algorithm assumes the battery is empty so the charge level is set to zero. The charge efficiency is calculated based on the total energy in and out since a recent charge efficiency calculation. An observation of the application is that the charge efficiency calculations happen often, resulting in unreliability and inconsistency for the user. Figure 29 shows an example of this, where the state of charge drops from 30% to 0% within 10 minutes because a low battery event occurs. The reason for this is that when the battery discharges under a low load overnight, it takes longer for voltage to drop to a level that triggers a low battery event. This means that charge efficiency calculations do not all happen at consistent points, resulting in calculations that are inaccurate. A potential solution is to prevent charge efficiency calculations where the load is below average. This targets the cases of early morning usage that get the battery lower than normal, however the impact of this solution has not been analysed.

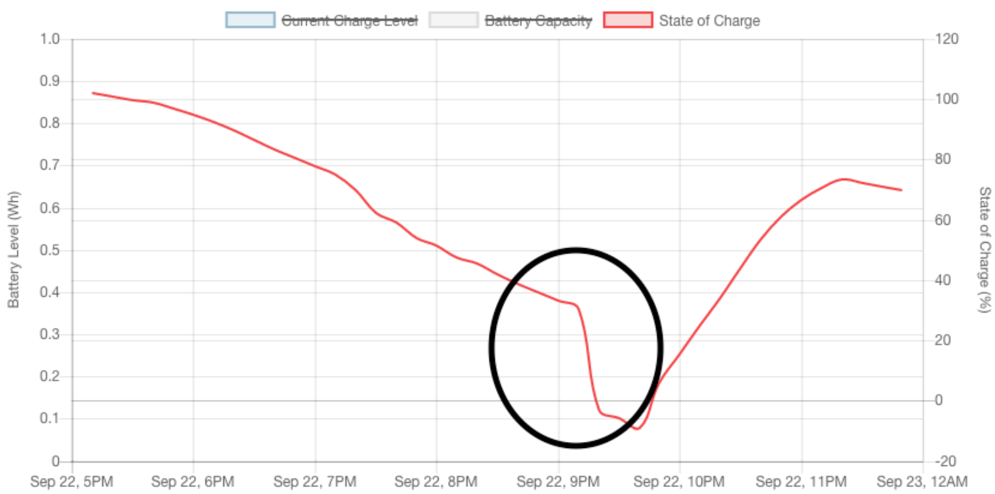


Figure 29. An example of the charge efficiency being re-calculated after a low battery voltage event in the night.

The application analyses energy flow to provide graphs of the total daily energy flows of the renewable energy system. However, the total daily consumption is often significantly less than the total daily generation (an example is shown in Figure 30). While some of this energy will remain in the battery between days, much of this energy is likely lost to the efficiency of the battery, the efficiency of the inverter and other equipment in the energy system. The application could be improved to detect the effect of more of these losses in order to fully account for all energy in and out of the system.

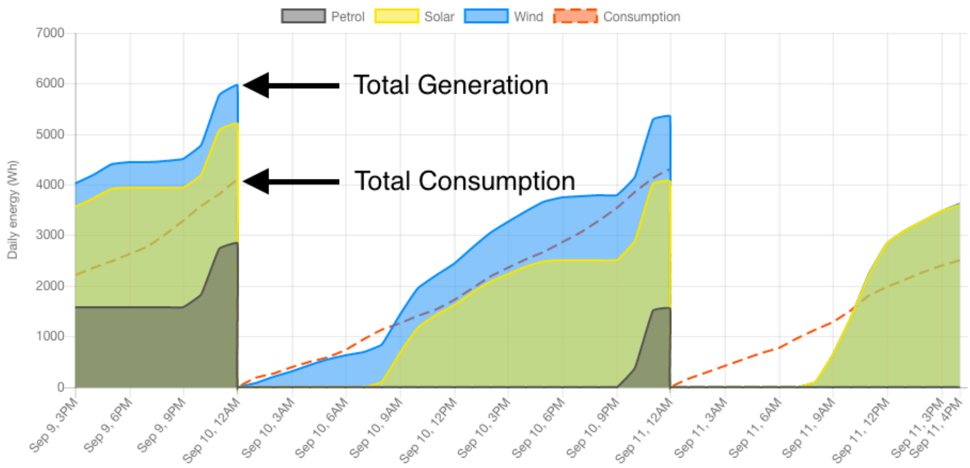


Figure 30. An example where the total daily consumption is less than the total generation.

The application estimates the future state of the battery based on predicted energy flows and determines the next significant event. However, the accuracy of these estimates has not been analysed. An observation of these is that the predicted 'next event' changes frequently throughout the day. The cause of this is most likely that the prediction multiplier causes predictions to be too extreme. The prediction multiplier is used to change the magnitude of predictions based on the comparison between the previous hour's actual and predicted energy flows. This is currently capped between 0.5x and 1.5x, but a solution may be to use a smaller allowed range for the prediction multiplier. These predictions are also affected by the recorded energy flows and state of charge, so improvements in these areas will likely improve the estimates given.

The application has been designed with future expansion in mind. The bridge allows expansion with new sensor device modules being easy to write and the backend supports upload from multiple bridge devices. Many of the data processing parameters are configurable, allowing the application to support other renewable energy systems. The energy sources and predictions have been implemented generically to allow for additional energy source types and configurations. The API in the backend means that new bridges or frontends could be developed. Overall, the data collection and processing structure provided by the monitoring system serves as a good base for future development to expand the application's capabilities.

Both the bridge and backend used the NodeJS environment, with the backend making use of the Loopback framework. The bridge software is highly asynchronous so benefitted from the use of NodeJS, however in the backend this led to additional complexity. The backend would benefit from a programming language that can block while waiting for an asynchronous result (like a database operation), rather than requiring the definition of a callback function. Debugging software written in

NodeJS can be difficult, leading to most debugging of the bridge and backend being through adding log statements and re-executing the code.

The use of AngularJS, the Loopback SDK for AngularJS and the AdminLTE template saved a significant amount of time while developing the frontend. This has resulted in a frontend that is responsive, visually appealing and can exercise many of the features of the backend.

5.2 Future Work

The scope for this project was large. Many of the planned features were developed but there are many additions that could be made in future work to improve the application. These include improvements to the way the application performs data collection, administrative functionality and data processing. To make future development easier, some improvements to the development environment should be made.

5.2.1 Data collection

The Intense PC computer used as the Bridge client is likely much more powerful than required for the purpose of data collection. Instead, a micro-computer such as the Raspberry Pi[15] could be used as a low-cost alternative. The Smart Circuit and Pentametric sensor devices were used to record readings at several points in the renewable energy system. These could be replaced with a single device capable of all monitoring. In addition, the Smart Circuit was likely more advanced than required as this device is also capable of recording readings in its internal storage. The Intense PC was setup on a uninterrupted power supply to allow data collection to continue after an LVD event occurs. However, if this computer does lose power (for example, if the uninterrupted power supply battery runs out), then any readings not uploaded to the backend will be lost. The bridge should be modified to support saving readings to a local database to prevent data loss.

The backend supports the generation of a CSV file as a method of exporting readings. This is generated with a single database query that iterates through all readings in the system. On large datasets, this can lead to excessive memory usage so should instead be implemented to process data in batches. The backend currently retains all readings. As the application is used for a longer period of time or with several buildings, this could quickly fill up the storage on the server it is hosted on. The backend should be setup to automatically clear old readings, or potentially store older readings at a lower frequency (e.g.: hourly readings instead of every 10 minutes).

[15] Raspberry Pi 3 Model B: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

5.2.2 Frontend administrative functionality

The following administrative functionality is missing on the frontend:

- The ability for users to register an account.
- The ability to perform basic user account actions, such as changing the name, email address or password.
- The ability to edit information about a building, bridge or sensor.

These tasks can currently only be performed manually through the API. To make the application easy to use, the frontend should guide users through the process of setting up a bridge client, possibly with a tool to generate a bridge configuration file.

5.2.3 Data processing

The backend data processing should be updated to support time zones and daylight savings time (DST). The application is presently hardcoded to work in the GMT+12 time zone with no support for DST, resulting in daily operations happening at the wrong time. There are many potential expansions to the data processing performed by the application. An example is that the application could determine recommendations for the user, such as whether more energy storage or generation is needed. The application could directly notify users when the battery is nearly fully charged or empty. The system should also gain support for generator automation. This would determine the most efficient time for the generator to be activated and save time for the occupants of the home. An integration with an online weather source could be used to improve future state predictions, especially for energy sources like small wind turbines.

5.2.4 Development environment

To improve reliability and development speed, automated tests should be written to ensure the application continues to work as expected. A continuous integration service should be used to automatically run tests and deploy stable versions of the system to the production environment.

6 Conclusion

Electricity is an essential utility for the conveniences of modern life. For the occupant of a home with a renewable energy system, having a method of determining the state of the system is vital to ensure a reliable energy supply. Currently, the only way to determine the state of a renewable energy system is through inaccurate readouts physically located on the equipment. A cloud based solution allows access to this information from anywhere and benefits from the additional computing resources available in the cloud to provide advanced data processing and historical data. There are no commercially available cloud systems that can provide the full state of a renewable energy system. Such a system is becoming increasingly important as the decreasing cost of solar panels and batteries increases the popularity of renewable energy systems.

The aim of this project was to develop a cloud based monitoring application for a renewable energy system. The application consists of three components; the bridge, backend and frontend. The bridge runs on a micro-computer located at the renewable energy system and sends data from the sensors to the backend. The backend is located on a cloud server and processes the data, determining the state of charge of the battery using the B42SOC algorithm. The frontend can be accessed from an internet connected device to show the state of the system. The energy flows within the system are identified, determining the total hourly and daily energy generated or consumed throughout the system. The application uses historical data to determine a prediction pattern for the house consumption and each energy source. This is used to predict state of the system for the next 24 hours, allowing the next battery event to be determined.

The application was originally planned to support automatically turning on and off the generator, however, due to time restrictions this was unable to be implemented. Continued use of the application has revealed the limitations of the B42SOC algorithm and some additional tweaks and functionality that would improve the application. Overall, the application delivers on its goal and gives users a method of monitoring their renewable energy system. The application allows access to this information from anywhere with an internet connected device (Figure 31 shows the system in use on a smartphone and Figure 32 of the system in use on a laptop). This allows users to make informed decisions about their energy consumption and appropriately decide when to run a generator to ensure a reliable energy supply. The application saves users time by having convenient and clear access to this information in comparison to integrated displays on the system's equipment. The application can be used with other renewable energy systems due to its high configurability. It serves as a base for future expansion with new bridges and frontends being able to be developed.

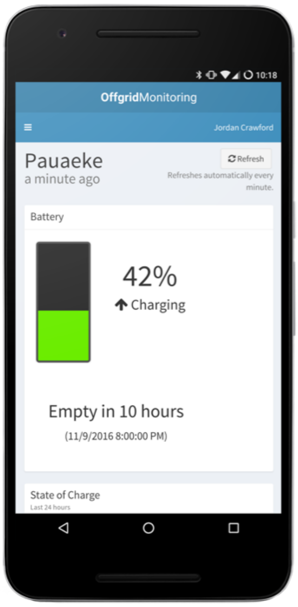


Figure 31. The system in use on a mobile device.

Figure 32. The system in use on a laptop.

7 References

Alahmari, M. M. (2013). Off-grid Energy Monitoring. Hamilton, New Zealand: University of Waikato.

Almsaeed, A. (n.d.). *Free Bootstrap Admin Template*. Retrieved from Almsaeed Studio: https://almsaeedstudio.com/

altE Store. (n.d.). *BOGART ENGINEERING PM-5000-U PENTAMETRIC INPUT UNIT*. Retrieved from altE Store: https://www.altestore.com/store/meters-communications-site-analysis/meters-battery-monitors/ammeters-voltmeters-battery-monitors/bogart-engineering-pm-5000-u-pentametric-input-unit-p253/

Apperley, M. (2016). *B42SOC: Better Black-Box Battery Simplified State-Of-Charge* .

Apperley, M. D., & Alahmari, M. M. (2013). Tracking Battery State-of-Charge in a Continuous Use Off-Grid Electricity System.

Banerjee, N., Rollins, S., & Moran, K. (2011). Automating Energy Management in Green Homes. *Proceedings of the 2nd ACM SIGCOMM workshop on Home networks.* ACM.

Bas, L. (2010, March 12). *Battery Life and Performance Factors*. Retrieved from Civic Solar: https://www.civicsolar.com/support/installer/articles/battery-life-and-performance-factors

Battery University. (n.d.). *Charging Lead Acid*. Retrieved from Battery University: http://batteryuniversity.com/learn/article/charging_the_lead_acid_battery

Bridges, S. (2016, March 24). *7 out of 10 households now have smart meters*. Retrieved from New Zealand Beehive: https://www.beehive.govt.nz/release/7-out-10-households-now-have-smart-meters

Canada Mortgage and Housing Corporation. (2001). *Energy Use Patterns in Off-Grid Houses.* Canada Mortgage and Housing Corporation.

Chouhan, H. (2013, June 4). *6 Reasons to Choose the Bootstrap CSS Framework*. Retrieved from OSTraining: https://www.ostraining.com/blog/coding/bootstrap/

Dunphy, C. (2014, September 6). *Understanding Solar Panel Specifications*. Retrieved from Technomadia: http://www.technomadia.com/2014/09/understanding-solar-panel-specifications/

EECA Energywise. (2015, September 24). *Small wind turbines*. Retrieved from EECA Energywise: https://www.energywise.govt.nz/at-home/generating-energy/small-wind-turbines/

EECA Energywise. (2016, June 1). *Solar PV systems*. Retrieved from EECA Energywise: https://www.energywise.govt.nz/at-home/generating-energy/solar-pv-systems/

Electric Kiwi. (n.d.). *Hour of power*. Retrieved from Electric Kiwi: https://www.electrickiwi.co.nz/hour-of-power

Electricity Authority. (2016, May). *What are smart meters?* Retrieved from Electricity Authority: http://www.ea.govt.nz/consumers/what-are-electricity-meters/

Enphase. (2016). *Enphase AC Battery.* Retrieved from Enphase: https://enphase.com/sites/default/files/downloads/support/AC-Battery-DS-EN-US.pdf

Enphase. (2016). *Enphase Envoy-S.* Retrieved from Enphase: https://enphase.com/sites/default/files/downloads/support/EnvoyS-DS-EN-US.pdf

Enphase. (n.d.). *An Introduction to Storage*. Retrieved from Enphase: https://enphase.com/en-us/products-and-services/storage/introduction

Enphase. (n.d.). *Enphase Enlighten and Apps*. Retrieved from Enphase: https://enphase.com/en-us/products-and-services/enlighten-and-apps

Fehrenbacher, K. (2011, June 26). *5 reasons Google PowerMeter didn't take off*. Retrieved from Gigaom: https://gigaom.com/2011/06/26/5-reasons-google-powermeter-didnt-take-off/

Flick Electric. (n.d.). *Choice*. Retrieved from Flick Electric: https://www.flickelectric.co.nz/choice/

Google. (n.d.). *AngularJS*. Retrieved from AngularJS: https://angularjs.org/

Kopytoff, V. (2009, February 11). *Google plans meter to detail home energy use*. Retrieved from SFGate: http://www.sfgate.com/business/article/Google-plans-meter-to-detail-home-energy-use-3172167.php

Kuenemann, M. (2014, August 8). *My Off-grid Solar System Monitoring*. Retrieved from Hackaday.io: https://hackaday.io/project/2333-my-off-grid-solar-system-monitoring

Lance, J. (2009). *Living Off-Grid: Our Micro Hydro Alternative Energy System*. Retrieved from Blue Living Ideas: http://bluelivingideas.com/2009/03/24/living-off-grid-micro-hydro-alternative-energy-system/

Maehlum, M. A. (2013, August 14). *Grid-Tied, Off-Grid and Hybrid Solar Systems*. Retrieved from Energy Informative: http://energyinformative.org/grid-tied-off-grid-and-hybrid-solar-systems/

McClure, T. (2015, April 9). *Free panels place solar power within reach for kiwis*. Retrieved from Stuff: http://www.stuff.co.nz/business/67650722/free-panels-place-solar-power-within-reach-for-kiwis

Mosaic. (n.d.). *The cost of solar panels are falling*. Retrieved from Mosaic: https://joinmosaic.com/cost-of-solar-panels-are-falling

Murphy, T. (2012, September 18). *Blow-by-Blow PV System Efficiency: A Case Study for Storage*. Retrieved from Do the Math: http://physics.ucsd.edu/do-the-math/2012/09/blow-by-blow-pv-system-efficiency/

Node.js. (n.d.). *Node.js*. Retrieved from Node.js: https://nodejs.org/en/

Northen Arizona Wind & Sun. (n.d.). *All About Maximum Power Point Tracking (MPTTP) Solar Charge Controllers*. Retrieved from Northen Arizona Wind & Sun: https://www.solar-electric.com/mppt-solar-charge-controllers.html

Peled, E., Yamin, H., Reshef, I., Kelrich, D., & Rozen, S. (1984, September 10). *United States of America Patent No. 4725784 A.*

Pop, V., Bergveld, H. J., Notten, P. H., & Regtien, P. P. (2005, October 31). State-of-the-art of battery state-of-charge determination.

Powershop. (n.d.). *Toolkit*. Retrieved from Powershop: http://www.powershop.co.nz/toolkit/

Radio New Zealand. (2014, November 7). *Radio New Zealand*. Retrieved from 'Outrage' at solar power buyback cuts: http://www.radionz.co.nz/news/national/258787/'outrage'-at-solar-power-buyback-cuts

REUK.co.uk. (n.d.). *Low Voltage Battery Disconnect Circuits*. Retrieved from REUK.co.uk: http://www.reuk.co.uk/wordpress/electric-circuit/low-voltage-battery-disconnect-circuits/

Rouse, M. (2005, September). *watt-hour (Wh)* . Retrieved from WhatIs: http://whatis.techtarget.com/definition/watt-hour-Wh

Shahan, Z. (2016, May 15). *EV Battery Prices: Looking Back A Few Years, & Forward Yet Again*. Retrieved from Clean Technica: https://cleantechnica.com/2016/05/15/ev-battery-prices-looking-back-years-forward-yet/

Sindelar, A. (2011, January). *Off-Grid Batteries - 30 Years of Lessons Learned*. Retrieved from Home Power: http://www.homepower.com/articles/solar-electricity/equipment-products/grid-batteries

Solar Homestead. (2015, August 14). *What Voltage Should My Battery Bank Be? 12, 24, or 48.* . Retrieved from Solar Homestead: http://solarhomestead.com/best-battery-bank-voltage/

SolarCity. (n.d.). Retrieved from SolarCity: http://www.solarcity.com/

Solarcity. (n.d.). *Pricing & Savings*. Retrieved from Solarcity: http://www.solarcity.co.nz/pricing-savings/

SparkFun. (n.d.). *Alternating Current (AC) vs. Direct Current (DC)*. Retrieved from SparkFun: https://learn.sparkfun.com/tutorials/alternating-current-ac-vs-direct-current-dc

SparkFun. (n.d.). *Electric Power*. Retrieved from SparkFun: https://learn.sparkfun.com/tutorials/electric-power

SparkFun. (n.d.). *Voltage, Current, Resistance, and Ohm's Law*. Retrieved from SparkFun: https://learn.sparkfun.com/tutorials/voltage-current-resistance-and-ohms-law

SparkFun. (n.d.). *What is Electricity?* Retrieved from SparkFun: https://learn.sparkfun.com/tutorials/what-is-electricity

SunFarmer. (2015, January 8). *Introducing Our New Monitoring Technology, Energy X*. Retrieved from SunFarmer: http://www.sunfarmer.org/energy-x/

Tesla. (n.d.). *Powerwall*. Retrieved from Tesla: https://www.tesla.com/powerwall

Transpower. (2016). Transmission Tomorrow 2016.

Vector Limited. (n.d.). *Tesla Energy*. Retrieved from Vector Limited: https://vector.co.nz/tesla-energy

WattsUp Meters. (n.d.). *Smart Circuit Controllers*. Retrieved from WattsUp Meters: https://www.wattsupmeters.com/secure/products.php?pn=20&wai=0&more=2

Wesoff, E. (2015, July 30). *SolarCity Has More Than 250,000 Customers, Reports Record Bookings, Installs*. Retrieved from GreenTechMedia: http://www.greentechmedia.com/articles/read/SolarCity-Reports-Record-PV-Bookings-and-Installs-in-Q2

What Power Crisis. (n.d.). *Micro-Inverters*. Retrieved from WhatPowerCrisis.com: http://www.whatpowercrisis.co.nz/Residential+Packages+with+Micro-Inverters.html

Woodbank Communications Ltd. (n.d.). *Battery Life (and Death)*. Retrieved from Electropaedia: http://www.mpoweruk.com/life.htm

Woodbank Communications Ltd. (n.d.). *State of Charge (SOC) Determination* . Retrieved from Electropaedia: http://www.mpoweruk.com/soc.htm

Xantrex. (n.d.). *Inverters/Chargers*. Retrieved from Xantrex: http://www.xantrex.com/power-products/inverter-chargers/overview.aspx

Zipp, K. (2015, August 21). *What is the best type of battery for solar storage?* Retrieved from Solar Power World: http://www.solarpowerworldonline.com/2015/08/what-is-the-best-type-of-battery-for-solar-storage/

Appendix A – Backend Model Schema

Below is a simplified version of the schema used by the backend. Bold represents relationships.

People:

- Email address
- Name
- **Buildings** <Building[]>

Building:

- Name

User defined building properties (see
- Appendix B - Configurable Parameters)
- **Bridges** <Bridge[]>
- **People** <People[]>
- **Exports** <Export[]>
- **States** <State[]>
- **Recalibrations** <Recalibration[]>
- **EnergySources** <EnergySource[]>

User defined sensor relationships (see
- Appendix B - Configurable Parameters)
- CurrentState <State>
- FutureStates <FutureState>

Bridge:

- Name
- Secret
- **Sensors** <Sensor>
- **Readings** <Reading>
- **Building** <Building>

Sensor:

- Name
- Type
- **Bridge** <Bridge>

EnergySource:

- Name
- ChartColour
- SortIndex
- PredictionPatternType
- IsRenewable
- PredictionPattern
- CurrentSensor <Sensor>
- **Building** <Building>

Reading:

- Timestamp
- Values <Sensor ID, Value>
- **Bridge** <Bridge>

Recalibration:

- Reason
- Timestamp
- **Building** <Building>

State:

- Timestamp
- PreviousEnergyInSinceLastC0
- PreviousEnergyOutSinceLastC0
- CurrentEnergyInSinceLastC0
- CurrentEnergyOutSinceLastC0
- BatteryCapacity
- BatteryLevelLowSince
- EmptyLevelLowSince
- EmpyLevelEstablished
- CurrentChargeLevel
- MaximumPrelimPhaseChargeLevel
- ChargeEfficiency
- IsBatteryCharging
- Sources
- Consumption
- **Building** <Building>
- **Reading** <Reading>

FutureState:

- Timestamp
- BatteryCapacity
- CurrentChargeLevel
- Sources
- Consumption
- **Building** <Building>

Export:

- Time started
- Time finished
- Status
- Export data until
- Export data after
- **Building** <Building>

Appendix B - Configurable Parameters

Building Configuration:

- **Critical low voltage level:**
  This is the voltage considered low. This is used to adjust the battery state of charge and automatically activate the generator. This is typically the LVD trigger voltage.
- **Low voltage trigger time:**
  The battery is considered empty once its voltage is at or below the critical low voltage level for this amount of time. Used to adjust the battery state of charge. Typically, this is around half the time until LVD.
- **Daily Aging Percentage:**
  The amount the battery level will decrease by each day. This is used to ensure the battery capacity is correct (the battery capacity will be re-calibrated if set too low). The recommended amount is 2% per day.
- **Tolerance Percentage:**
  The tolerance percentage defines when re-calibration events will occur. The recommended amount is ±5%. This means the battery capacity will be adjusted upwards once it reaches 105%, and state of charge won't be re-calculated until it reaches -5%.
- **High power threshold:**
  This is the power in watts that is considered high and out of the ordinary. LVD events caused by high power loads don’t represent low battery levels so do not contribute to the state of the system. Can be used for generator automation.
- **Process data after *(optional)*:**
  Processes data starting at the defined date and time. If not defined, all data up to "Process data until" is processed.
- **Process data until *(optional)*:**
  Stops processing data after the defined date and time. If not defined, all data up to the latest reading is processed.
- **Re-calculate charge efficiency multiplier:**
  For the charge efficiency to be calculated, there must be sufficient energy in/out since the last time the battery was empty. The level required for this is a multiple of the current battery capacity. The recommended value is 2 x the current battery capacity.
- **Size of reading gaps to ignore *(optional)*:**
  When there are large gaps between readings, this can cause incorrect data while processing. This allows the threshold for a reading to be ignored to be defined. This is defined in

minutes. If not defined, there is no limit for the size of gaps. The recommended value is 5 minutes.

- **Battery current sensor:**
  This is the sensor that is used to measure the battery current used in state of charge calculations. Assumes positive means charging.
- **Battery voltage sensor:**
  This is the sensor that is used to measure the battery voltage used in state of charge calculations. This is also used to detect a likely LVD state with the LVD voltage.
- **Building power sensor:**
  This is the sensor for the building's total power usage. This is used with the high power threshold to detect situations where LVD may occur but not due to low battery level.
- **Load current sensor:**
  This is the sensor that records the current through the inverter/charger. If this is above zero, then the system is being charged by an external source. This allows the incoming charge to be measured.
- **House consumption colour:**
  This is the colour to display house consumption as in graphs.
- **Standard axis for power graphs:**
  This is the power that is used as the Y axis for power graphs, unless a higher value is required.
- **Standard axis for daily energy graphs:**
  This is the daily energy in Wh that is used as the Y axis for daily energy graphs, unless a higher value is required.

Energy Source Configuration:

- **Name:**
  The name given to the energy source. Used to identify the source in charts or tables.
- **Source Current Sensor:**
  This is the current sensor that provides the current for this source. Expects values above zero mean charging, and any value below zero is ignored.
- **Graph sort index:**
  A number that defines the position in the sort order of energy sources. A low number means at the front of the sensor order, a high number means at the button of the sensor order. This is used to define the position of energy sources within graphs.
- **Chart colour:**

This is the colour to display the source with in graphs.

- **Prediction Type:**

  This defines the type of prediction model to use. This is used in future state estimations.

Appendix C - User Authorisation Rules

In additional to token based authentication, a user can only access content as per the following rules:

- **People:** Can access the basic information about all people so users can invite each other to a building.
- **Building:** Can only access buildings you own.
- **Export:** Can access exports for buildings you own.
- **Recalibration:** Can only access recalibrations for the buildings you own.
- **State:** Can only access states for the buildings you own.
- **FutureState:** Can only access future states for the buildings you own.
- **Bridge:** Can only access bridges for buildings you own.
- **EnergySource:** Can only access energy sources for buildings you own.
- **Sensor:** Can only access sensors belonging to bridges of buildings you own.
- **Reading:** Can only access readings recorded against a bridge of a building you own.