The use of a power model to assist the design of an energy harvesting data logger

By

Russell James Preece

A thesis submitted to the University of Birmingham for the degree of MASTERS OF SCIENCE

Department of Electronic Electrical and Systems Engineering

College of Engineering and Physical Sciences

University of Birmingham

May 2017

Abstract

This work shows how the development of a power model for an energy harvesting data logger can be used to predict the energy usage of the hardware as well as significantly contributing to the creation of a business case for a system deployment. The model is also used to demonstrate how investing in power modelling of the hardware can drive overall cost reductions in the design cycle, by reducing prototype costs and maximising design efficiency. This work examines an approach to developing a power model for a specific hardware design and develops a method for determining the feasibility of an operational deployment. Additionally, this work uses the power model developed to create a feedback loop from the model into the iterative hardware and firmware development of the data logger.

Acknowledgements

This thesis was started whilst working as a Knowledge Transfer Partnership Research Associate at the University of Birmingham and Arrowvale Electronics, whilst trying to negotiate the fine line between academia and industry many of the staff of both companies assisted in various ways for which I am eternally grateful.

I am particularly grateful to Roy Amos and Tahir Hanif whose support in the commercial and engineering development of the data logger schematics as part of Arrowvale's product offering, to Mark Anderson for writing the initial firmware test code and to Sam Verity for pointing me back in the right direction.

My academic supervisors, Dr Edward Stewart and Professor Clive Roberts, for recommending that I undertake an MSc as part of the KTP and to Knowledge Transfer Partnerships for providing financial support for me to do so.

My Parents, Family and Friends for keeping me on track and persisting with me throughout, and above all to my wife Rebecca without whom I would never have got to the end.

Table of Contents

Table of Figures

Table of Tables

Table of Equations

Table of Abbreviations

Abbreviation	Description
3G	3rd Generation Wireless Mobile Telecommunications Technology
AC	Alternating Current
ADC	Analogue Digital Converter
CF	Compact Flash
CPU	Central Processing Unit
CSM	Common Safety Method
CSV	Comma Separated Value
DAI	Digital and Analogue Input Card
DAQ	Data Acquisition
DC	Direct Current
DCDC	DC/DC Convertor
DMA	Direct Memory Access
EFM	Energy Friendly Microcontroller
EH	Energy Harvesting
EHDL	Energy Harvesting Data Logger
EMA	Exponential Moving Average
EMC	Electromagnetic Compatibility
FMECA	Failure Mode, Effects and Criticality Analysis
FRAM	Ferroelectric RAM
GPIO	General Purpose Input / Output
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HFXO	High Frequency Crystal Oscillator
HSDPA+	Evolved High Speed Downlink Packet Access

Abbreviation	Description
I/O	Input / Output
I^2C	Inter-Integrated Circuit
ITT	Invitation to Tender
LCD	Liquid Crystal Display
LFXO	Low Frequency Crystal Oscillator
MB	Megabyte (1 Million Bytes)
MHz	MegaHertz
MPPT	Maximum Power Point Tracking
PC	Personal Computer
PCA	Printed Circuit Assembly
PCB	Printed Circuit Board
PRS	Peripheral Reflex System
PWM	Pulse Width Modulation
RAM	Random Access Memory
RTC	Real Time Clock
SGA	Strain Gauge Amplifier Input Card
SPDT	Single Pole Double Throw (Switch)
SPI	Serial Peripheral Interface
SPST	Single Pole Single Throw (Switch)
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

1. Introduction

1.1. Background to the research

One of the driving factors of remote condition monitoring is the detection of potentially catastrophic events before they occur. The landslide and derailment of a train at Watford Tunnels in October 2016 is one such example [1], in this instance only four people were injured, but incidents similar to this could easily result in multiple loss of life. The background to this work comes from a brief to develop a data logger which can be deployed in remote locations to monitor railway infrastructure. The logger was to be designed to monitor fixed assets such as bridges, embankments and access points. The specific scenarios for such deployments include the installation of strain gauges on to fixed structures such as bridges and walls near the railway, as well as monitoring the embankment for slippage prior to it being discovered by an oncoming train.

All of the examples above can often occur in remote locations where external power is not immediately accessible or the cost of laying power cables is prohibitive. In such circumstances there are two primary methods of powering a data logger, the first is battery power, which eventually needs to be visited to replace the batteries. The second method of powering a remote data logger is energy harvesting technology which can be employed in an attempt to create a perpetual deployment which requires no track access or line blockages to service and replace batteries.

1.2. Hypothesis

This thesis investigates the use of a power model to assist the design of an energy harvesting data logger. Through analysis of circuit design and power characteristics of components, this thesis aims to determine if it is feasible to generate a generic framework for power modelling which could be used to investigate the practical life of a battery powered data logger. The main themes to be covered are as follows:

1. Is it possible to generate a power model for specific electronic hardware and use the model to determine the operational feasibility of the hardware?
2. Is it possible to use a power model to feedback into the design of the hardware to develop a more efficient design?
3. Can adopting a power model as a part of the design approach reduce the overall system cost of a hardware design?

1.3. Scope

This thesis covers the hardware design strategy of the data logger and the design analysis of the physical hardware. It also provides analysis of each circuit at a component level whilst the device is operating in the various functioning modes to generate data for the model. Additionally, the thesis covers the generation of a power model for the analysed hardware and the subsequent testing and analysis based on model usage.

This thesis does not cover the physical design of the schematics and PCB layout of the data logger or the firmware which operates within the hardware. It also does not cover the functionality of the logger as a data logger: for example, no detail is provided on signal conditioning or recording parameters which do not directly affect the power model.

This thesis does not cover the user interface of the power model, but does make suggestions as to how this should be undertaken.

1.4. Thesis Structure

This thesis is broken down into 9 chapters, excluding references and appendices. A brief summary for each chapter is provided below.

Chapter 2 focusses on the background of the thesis and the reasons why this project has been undertaken, as well as detailing some of the business case for such as system

Chapter 3 contains a review of relevant literature. This ranges from wider reaching research focussing on energy efficiency as well as focussing more closely on specific topics such as railway condition monitoring.

Chapter 4 is a general overview of the generic approach which has been taken: this section focusses on how the work was undertaken, how energy usage was calculated and additionally looks at how units are used within the document

Chapter 5 is an application specific case study of the Energy Harvesting data logger hardware. This section drills into the detail of the hardware, calculating the energy usage in the various operating modes covered by the power model.

Chapter 6 details how the power model was designed, how the information generated in chapter 5 was used to populate the model and some functional testing of the model.

Chapter 7 details the steps required to tie the model generated in chapter 6 to the hardware for which it was designed. This section focusses on the changes required to the logger to enable it to confirm that the values calculated by the model are accurate.

Chapter 8 details further work which has been considered as part of this thesis which has either minor conclusions or could be continued as an extension to this thesis.

Chapter 9 concludes the thesis detailing the findings and recommendations of the work as well as providing an analysis of the approach taken.

2. Background

2.1. Introduction

Monitoring of remote assets in the rail industry is becoming a major priority; in infrastructure, recent landslips have made national headlines. If the asset that requires monitoring is itself a powered electronic device, the required logger can be powered from the same power source. However, if the asset is not an electronic device, for example earthworks, bridge structures or access points, the task of monitoring performance becomes much more difficult. In circumstances where such monitoring is required, an alternative supply of energy is required

Historically, such logging systems would be battery powered and, as a part of regular maintenance, the battery would be replaced to extend the life of the product. Such activity can be difficult and expensive in other industries but is significantly greater within the rail industry due to track access difficulties and the geographically disparate nature of the asset requires costly coordination to maintain.

In recent years worldwide research has been focussed on methods of harvesting energy from the ambient surroundings, and it is predicted that the total market for energy harvesting devices will rise to $2.6 billion by 2024 [2]. Energy harvesting devices have become more and more common, ranging from calculators that are recharged using solar panels to an attempt at circumnavigating the world in a single flight in a plane powered solely by solar power [3].

When considering deployment of data logging equipment, the major priority is the feasibility of the deployment: if the logger is not fit for purpose then the installation is ultimately going to fail, or require additional attention. A separate, but equally important consideration is the exercise of costing the installation, determining if the business case adds up. If the installation can technically be a success but would cost more to implement and maintain than the value that it brings, it would be near impossible to fund such an installation.

With a rolling stock based aspect, the deployment of sensors within a vehicle can also be a difficult issue. Dependent on the specific asset to be monitored, the same issues with power can occur as with infrastructure. Rolling stock also has its own application specific problems, since the retrofitting of condition monitoring equipment can be a costly endeavour. Even if space within the vehicle can be found, there can still be problems. Depending on where the equipment is mounted, the design of the equipment must meet standards for EMC compliance, ingress protection as well as mechanical integrity. Also, with some vehicles currently in service, the original manufacturer of the equipment will often not have made provision for such monitoring or in some cases the company may have ceased to exist since the vehicle was manufactured.

This thesis focusses on the technical implications of an energy harvesting deployment in an infrastructure based application but is easily transferable to rolling stock based applications.

3. Literature Review

In this section a review is undertaken of existing works relevant to the subject matter. There are two major contributing sources to the following section: the first type of sources is published academic works which touch the area covered by the thesis, the second source of data is a review of commercially available products or applications which target similar objectives, which give a view into the industry within which this work is undertaken.

3.1. Power Modelling

The first section of this literature review focusses on existing work in the field of power modelling. Power modelling is used in varying fields from large scale projects such as modelling the effects that lifestyle can have on the energy consumption of households [4] [5] to modelling the energy consumption of small electronic projects such as solar lights which are used as security lights.

On an industrial scale one of the largest single costs which faces a lot of business is the price paid for energy be that as fuel (Petrol, Diesel, Natural Gas) or as Electricity from the mains, often businesses actively engage to reduce these expenses. Energy or power modelling is often used to improve the efficiency of a process, for example in a manufacturing business if the energy usage of a machining process can be understood, its inefficiencies can be identified and used to reduce the consumption of energy [6].

As the automotive industry moves from its roots with internal combustion engines to emerging technologies such as hybrid cars and all electric cars the need to model the energy usage of new technologies has emerged [7]. Electric cars have come with a new suite of problems which have not been an issue for the automotive industry for centuries, since fuel stations have been widespread the likelihood of running out of fuel has not been a pressing issue, but one that is re-emerging in the modern era. Expanding on this, the following paper [8] models the impact that integrating electric cars into the power grid will have and the cost implications. This 'business case' based approach is one which will be investigated as part of this work.

On a much smaller scale than the previous examples, but at a relevant level is a market for power modelling which has expanded rapidly in recent years. With the introduction of the iPhone in June 2007, the popularity of 'Smart phones' increased significantly. With the increase of functionality on smart phone came additional demand on battery technologies. With the development of such technologies not advancing to meet demand other approaches were taken to manage the power usage of smart phones. One of these was to gain a better understanding of the significant power uses of smartphone [9]and extension of these works is looking into the power consumption of other more specific systems [10] such as the modelling and optimisation of power consumption in various wireless networks such as those used in phones either as WiFi or cellular radios

3.2. Data Logging

The first commercial data logger was announced by IBM in December 1963 [11]. The IBM 7700 Data Acquisition System could simultaneously collect data from 32 sources, process them and transmit the results to as many as 16 remote printers, display units or plot boards. It was succeeded by the IBM 1800 DACs [12].

The general process of data logging is not too dissimilar from the original methodology. Modern data logging is often done though equipment such as National Instruments CompactRIO systems [13] . These systems are modular enclosures which allow for the implementation of varied inputs and outputs within a common chassis and software front end. Such systems are regularly used as part of research and development projects, but can often be deployed as part of larger projects [14]. Such systems allow for rapid deployment of data recording hardware as well as enabling easy data readability using integrated tools.

However, these generic systems may not always be the best solution – they are often designed for generic problems, aiming to cover 90% of applications: unfortunately, in this case, hardware not designed to be ultralow energy may not be best suited to the application. For example, a CompactRio cRIO-9063 draws 18 Watts when fitted with four C series modules [15]. This module is the 'value' model, the 4 slot 'performance' model the cRIO-9034 is recommended for usage with a 100W power supply.

3.3. Energy Harvesting

"Fossil fuels are finite and environmentally costly. Sustainable, environmentally benign energy can be derived from nuclear fission or captured from ambient sources. Large-scale ambient energy (e.g. solar, wind and tide), is widely available and large-scale technologies are being developed to efficiently capture it.

Figure 1 - Wind Farm Installation [85]

At the other end of the scale, there are small amounts of 'wasted' energy that could be useful if captured. Recovering even a fraction of this energy would have a significant economic and environmental impact. This is where energy harvesting (EH) comes in." [16]

Research in the field of energy harvesting is focussed in three major areas: the initial collection of energy from ambient surrounding sources, the optimisation of the conversion and storage of harvested energy and the efficient use of the stored energy to drive increased functionality or introduce new technology.

Traditional energy harvesting has focussed on the harvesting of solar, wind (Figure 1) and tidal energy on an industrial scale. Schemes such as Pelamis [17] have captured the general public's attention but energy harvesting installations on this scale are normally controversial, for example, in various guises, proposals for the Severn Barrage have been proposed since the 19^{th} century, but the project has not been undertaken [18].

On a smaller scale, energy harvesting projects have become more common. These types of project generally fit into two categories, firstly projects which use energy harvesting to extend the battery life of a product and, secondly, projects powered entirely from energy harvesting. There is often some crossover between these two categories.

Of the first type, one of the oldest examples is the solar powered calculator [19]. The principal of this method is to reduce the load on the battery by providing some of the power requirement from an energy harvested power supply such as a small solar panel or in some cases a wind turbine. Additionally, such devices can often be retrofitted by adding an energy harvesting source to an existing battery powered system: an example of such is a solar panel for retaining charge in a car battery [20].

The second category is essentially an extension of the first, the difference being that the amount of harvested energy developed is sufficient to power the device without the need for a battery. Such installations are generally larger to provide the power required to maintain operation [21]. These systems are often deployed in non safety critical applications, where a loss of power due to insufficient harvesting is an acceptable condition. To protect against such failure, storage capacitors or batteries are often installed to cover gaps in harvested energy. This is where the lines between the two categories begin to blur.

The ability to accurately gauge the amount of harvested energy required is critical: various amateur [22] and professional [23] tools are available; however, they are often designed for a specific purpose or to promote a specific product. The main issue with such applications is that they are designed to determine the energy usage of specific devices or circuit designs. One such example of a field where this is prolific is the wireless area networks, such as Bluetooth or Sub 1 GHz network devices. These devices are designed to be used in low power applications and therefore lean towards such power calculation tools. The energy calculation tools such as these focus on the product the manufacturer is trying to sell wheras the purpose of the energy model designed within this work is to investigate the power usage on a much larger scale, across multiple discrete devices.

Energy harvesting calculators often do not take into account a realistic representation for the amount of energy which can be harvested. Calculating the amount of energy from a solar panel is a scientific process, based in this case on the photovoltaic effect, i.e. it has an equation with known variables; however, it is dependent on natural inputs. In laboratory environments or in designs integrating a concentrator, energy can be harvested using solar cells with an efficiency of up to 25% [24]. This, however, translates badly into real-world applications. The work of Aloulou et al [25] details a study of improving the circuit model of a photovoltaic cell. There are two reasons for this; firstly, to accurately determine the relationships inherent in the solar cell such as its I/V and P/V characteristics and dependence on temperature. The second reason for the modelling of the solar cell is to show the benefits of a technology known as Maximum Power Point Tracking [26], or MPPT: this technology is embedded into some solar energy harvesters to maximise the amount of harvested energy. This is achieved by varying the load resistance seen by the solar module – this is demonstrated in figure 9 of Aloulou et al [25].

Another approach which needs considering is how the solar module performs in real life conditions. In the previous example [25], all testing was completed using, for example, a 1000W/m light source. This is suitable for 'calibration' type testing but does not accurately reflect the amount of energy which would be available at the point of use in a real world application. The following paper [27] looks into the possibility of using weather forecast data to predict the amount of harvested energy, looking at both wind and solar technologies. Figure 1 of the referenced paper displays the sporadic nature of harvested energy across time; it is worth noting, however unsurprisingly, the relatively short duty cycle for which solar energy can be harvested at a practical level. The main drawback to this work is that it focusses on predicting using the upcoming days'/weeks' weather forecasts, this could be expanded by the addition of a historical amount of data, i.e. if an average could be levied for the previous, 1, 2, or 5 years, an accurate prediction could be made [28]. By leveraging historical data and some modelling, it is possible to create a profile for the amount of harvested energy, thus allowing for a low error estimate of the amount of energy which can be harvested.

The proposed application for the model developed in this project is a microprocessor based system, controlling various peripherals. By making the processor ‘aware’ of its energy harvesting nature, it might be possible to optimise its operation for the power supply. Such projects have been undertaken previously [29]: such studies look at weighing the balance between saving energy for continued device operation and the requirement to complete the necessary software task in the required time, accompanied with other papers focussing on the duty cycling of energy harvesting system [30].

In recent years, the number of applications for the deployment of energy harvesting devices has increased with the acceleration of the ‘Internet of Things’ [31]. Generally, in such applications, an energy harvesting source is used to power a small device which operates a specification function and reports data to other nearby devices through a wireless link: one example of such connectivity is removing the need for wiring within the home using wireless light switches [32]. In the example shown, the action of pressing the button on the light switch generates enough power to send a signal to the controller to turn on the light. Other more complex systems can operate through meshed networks, which remove a rigid network structure and fixed communication paths an instead dynamically route traffic through whichever nodes are available based on signal strength or power state. Meshed networks increase functionality, reliability and versatility, but at the cost of a higher energy requirements at certain parts of the network. The nature of such products [33] opens up the concept of time-slotted operation: if persistent monitoring is not required, the system could operate a sleep mode; when additional functionality is required the system would wake. Such a method of operation allows the system to preserve or build up stored energy when usage is not required, enabling a more ‘energy intensive’ window when required.

3.4. *Wider Condition Monitoring*

In the bigger picture, the railway industry is just a small part of the heavy industrual sector, and condition monitoring is becoming more and more common for similar reasons. Such deployments can work both ways – for example Perpertuum [34], a company best known for their rail application products, have also deployed their energy harvesting technology across various sectors [35] such as oil and gas, process manufacturing and power generation.

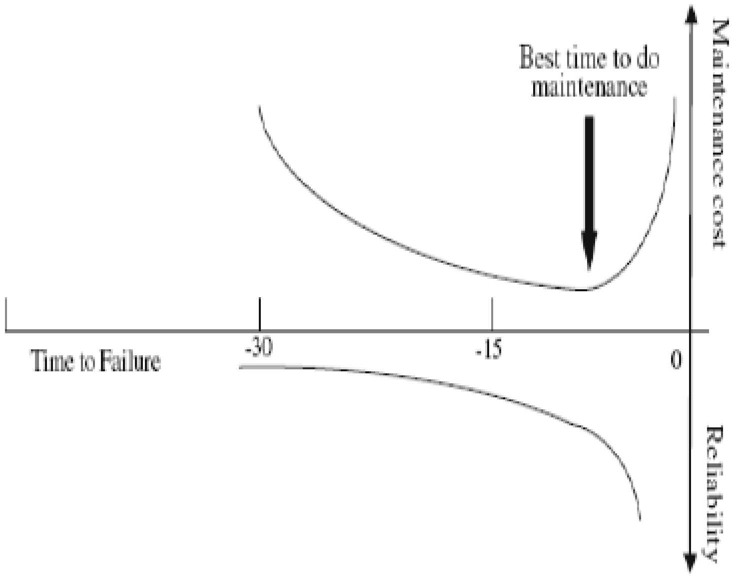


Figure 2 - Correlation among Remaining Useful Life, maintenance cost, and reliability [37]

The wide reaching appeal of condition monitoring is shown by its appearance in journals and conference such as the 2013 IEEE International Conference on Computational Intelligence and Computing Research [36]. This overview details many of the reasons for condition monitoring as well as some of the cross-overs between detection techniques and applications. It also brings to the table the concept of Residual Useful Life (Figure 2); this is common with the concept of Fault Detection and Diagnosis [37]. These concepts focus on the detection and tracking of the deterioration of a mechanical part. For example, if a reciprocating part is beginning to ‘stick’, this can now be detected by condition monitoring, but just because the equipment is only beginning to fail it does not mean it requires immediate maintenance. By analysing a series of measurements it is possible to determine the optimum time for maintenance. If maintenance is carried out too soon, potential operational life is lost from the original part, resulting in increased maintenance activity and increased lifecycle cost. However, if maintenance is carried out too late the part could fail in service, resulting in unexpected down time and loss of production, as shown in Figure 2.

Remote Condition Monitoring of High Voltage Insulators Employing GSM [38] is a novel approach to using GSM networks to relay data – by using the voice transmission capability of the network to transmit a signal rather than its data carrying capacity. Whilst this technique requires a level of post-processing, it significantly reduces the amount of local processing required to almost nothing – simply to open a voice call and record the sound. Also, by using the voice channel rather than data channel, the cost of transmission is greatly reduced, at present – the cost of mobile (GSM) data is 'roughly' £2 per 50MB (on a 'pay as you go' contract). By shunning the use of mobile data (i.e. 3G or HSDPA+) the power requirement of any cellular modem is also reduced, due to the less intensive method of connection [39].

Coming full circle, condition monitoring is now being applied to energy harvesting equipment [40], looking to both the mechanical and electrical signatures of the equipment to determine faults – the paper focusses on studying the difference between the perfect running conditions and induced failure conditions – thus allowing for the building of a fault 'library' where in operation faults can be detected and diagnosed by matching the signatures with those detected during laboratory systems. The only drawback of such as system is that, if a fault condition develops which has not been characterised, it will not be detected by the automated system.

Condition based maintenance is one part of the larger field of maintenance planning [41]. When planning the maintenance of a system one significant impact is the commercial cost of the maintenance plan. If maintenance is being carried out too frequently the overall cost of the maintenance is higher than necessary. One method used to reduce these cost is reliability centred maintenance [42]. Reliability centred maintenance looks at the component parts of a system and reviews the failure modes of each individual part. If a failure mode cannot be detected by a maintenance intervention then checking for the failure is inefficient and does not add value. If a component shows no sign of deterioration before failure it may be more cost effective to replace the component on failure, depending on the criticality of the component. Inversely if a component is regularly failing to make the time between maintenance interventions it may highlight that the intervention is carried out too infrequently.

Beyond specific maintenance methodologies, research has also been carried out into the development of methods for the analysis of differing maintenance strategies [43], these studies aim to give metrics against which differing approaches can be compared.

3.5. Rail Industry

For most of its history, for rolling stock, the UK rail industry has focussed on time or mileage based or periodic maintainance [44] [45] [46]. These methodologies work by carrying out replacement or repair of parts once a certain time-frame has elapsed, or when a certain number of

Figure 3 - 6 February - Track engineers' survey the damage [29]

miles have been covered. These methods can be inefficient, for example, if a component part is replaced after a certain amount of time, but the wear has been less than expected and thus maintainance could have been delayed, thus extending the life of the existing part and saving costs. Condition Based Maintainance is an alternative approach which offers one solution to this problem. Previous studies focussed on the benefits of integrating condition monitoring into existing maintainance procedures [47] to create condition based maintenance. Integrating condition monitoring can create a number of benefits, firstly to extend the life of healthy equipment and secondly to detect defects prior to failure in service [37] as well as reducing the lifecycle cost of the various components and systems.

Additionally, within the UK and wider rail industries, the deployment of the Common Safety Method for risk evaluation and assessment [48] is used as the primary method for approval of engineering change: application of the CSM has been in force since 1st July 2012 [49], and aims to harmonise the risk management processes used across the rail industry [50]. A major component of the system for mitigating risk within the CSM is whether the proposed modification can be monitored either during routine maintenance or in service. Monitoring through routine maintenance relies on easy access to the part in question but does not require additional equipment. Monitoring the equipment whilst in service involves fitting monitoring devices to the train which can collect data on the asset.

The monitoring of asset condition is frequently used to extend the maintenance periodicity of certain maintenance tasks [51], where it is noted that mechanical components are often removed from the vehicle with a minimal amount of wear, well within tolerance. It is therefore possible to extend the periodicity of the maintenance task. The addition of in service monitoring to this methodology can enable the significant cost savings mentioned previously. For example, if the life of a bearing can be extended before it needs replacing, over the lifetime of a vehicle this could in fact save an entire set of bearings. i.e. if the replacement of bearings can be extended to from 1 million miles to 1.2 million miles (a 20% increase) then over the lifetime of the vehicle (approximately 10 million miles), the cost of two complete sets of bearings could be saved. The business case of such monitoring is very simply described above, but it doesn't often deviate too far from the simple maths presented.

For most applications such monitoring can be powered from local power supplies; however, requirements such as EMC compliance and Ingress protection can often lead to large bulky enclosures, and increased footprint. These issues, coupled with the cost of installing additional cabling onto the vehicle and modifying wagons, can often lead to a difficult business case. However, there are other possible solutions, one of which is to use small scale energy harvesting technologies [34], Perpetuum is one example of this practice. The Perpetuum system uses a vibration energy harvester to generate power from the sub 20Hz vibrations generated by rail vehicles. The system described above can be installed in various locations around the vehicle, but is most effective when mounted to the bogie or vehicle underframe due to the filtering effect of the vehicle's primary suspension. To further reduce the amount of cabling required (in particular due to the required breach of the vehicle's fire-wall), the Perpetuum system uses a wireless mesh network to transmit data from the vehicle. Such systems are becoming more common in the rail industry [52], where wireless sensor networks are being deployed to overcome this problem: this is detailed further in section 3.5, where wider development in the field are covered.

On an infrastructure front, the UK rail industry has had a range of high profile infrastructure failures within the previous few years, the most severe of these incidents was the collapse of the sea wall at Dawlish in early 2014 [53]. On Tuesday 4th February a 100m section of the sea wall at Dawlish collapsed during heavy storms – this resulted in the track being suspended across the gap, with trains unable to pass for just under two months. As part of the repair work, Network Rail also had to engineer a controlled landslip to prevent future incidents

Other examples of such incidents include a spoil heap collapse at Hatfield Colliery [54], a landslip at Harbury Tunnel in Warwickshire [55] and a rail void at Colwall in Worcestershire [56] [57]. In such incidences, the problem is often discovered primarily by the first train to encounter the problem: this has become such a problem for Network Rail that Future Railway issued a competition to find a solution [58], inviting bids to show 'How to prevent trains running across track which is rendered unusable by movement of earthworks such as cuttings and embankments'. There are several technical challenges detailed in the ITT [59], the most significant of these is the remote location of monitoring, since in such locations there is often not power available to power monitoring devices, which clearly plays into the field of energy harvesting.

3.6. Wireless Sensors

One major area of development for low power data logging equipment is in the field of wireless sensors, this area has pushed the boundaries of what can be achieved with small battery powered or energy harvesting systems.

As mentioned previously in section 3.3, this is the sector where most existing energy modelling tools have been developed. The primary purpose for these tools is to enable a designer to predict the power usage of individual devices in various operating modes. The idea behind these models is to convince a customer that a device is suitable for a project prior to the cost heavy investment of prototyping and testing. On a fundamental level the same could be said for this work, but on a larger scale, looking more at the feasibility of a whole product deployment rather than individual devices.

A lot of academic research has been focussed on the energy usage of wireless networks. Within a mobile phone, the largest usage of energy comes from the GSM module when in sleep mode [9]: when the phone is active, other functions overtake the GSM but the average power consumption of the GSM module is still significantly higher than any other single source. [60]

Additionally large amounts of academic work have been undertaken in the field of local wireless networks: such networks can be mainstream networks such as WiFi or Bluetooth which are relatively well known, as well as less mainstream networks such as Zigbee, or other Sub 1 GHz networks such as 433 MHz or 868 MHz networks [61]. These studies are also advantageous to this work as they discuss how best to reduce the power consumption of wireless networks. The main differences between the various wireless networks is the software protocols operating or the physical layer of the antenna and frequencies.

3.7. Energy Efficiency

Capacitors are commonplace within modern electronics but pose a unique problem within the field of energy harvesting. To fully charge a capacitor, additional energy is required beyond that stored within the capacitor. i.e charging a capacitor is a very inefficient process since, when fully charging a capacitor, only half of the energy drawn from the power supply is stored by the capacitor.

The amount of work required to charge a capacitor can be calculated (in joules) as

Equation 1 - Energy Stored in a capacitor

$$W = \frac{1}{2}CV^2$$

Where C is Capacitance (in Farads) and V is Voltage (in Volts)

The energy drawn from the power supply to charge the capacitor is QV. This power is, however, not applied instantaneously to the capacitor. The Capacitor charge curve is generally recognisable: the voltage on the capacitor at any single moment is shown in Equation 2. During this charging phase, the current flowing from the power supply to the capactor is passing through a resistive element (even if the resistance is minimal) .

Equation 2 - Voltage on Capacitor

$$V_{capacitor} = V_b\left[1 - e^{-t/RC}\right]$$

In Equation 3, $V_{capacitor}$ can be calculated using Equation 2. Assuming the voltage from the power supply is constant, which can be deemed a safe assumption as long as the current drawn by the capacitor is below any physical limiatation. Whilst the capacitor is charging, the remainder of the voltage drawn from the power supply must be dispersed within the intermediary resistance.

Equation 3 - Voltage dissipation during capacitor charging

$$V_{Total} = V_{Resitor} + V_{Capacitor}$$

If no series resistor is provided the system will draw a maximum current through the PCB tracking on which the component is mounted. The maximum current drawn by the capacitor can be calculated using Ohm's law within the series resistor – at the moment the system is powered on, the voltage present in the capacitor is zero. Therefore, all of the voltage is dissipated in the series resistor. This maximum current must be within the tolerance of the power supply otherwise the voltage rail will be pulled down or the power supply could be damaged.

The resistance of a 0.01 inch wide, 1 oz/ft^2 PCB trace is approximately 0.05 Ω per inch (note: imperial units are used as these are still commonplace units for PCB design). In a 3.8 V system this gives a maximum current draw of 76 A. The initial charging current on the capacitor must be controlled by introducing a series resistance. If a 100R resistor is placed in series, the maximum current draw is reduced to 0.038 A. A value between these extremes is preferable as the increase in resistance increases the time taken to charge the capacitor ($Q = IxT$)

The value of the series resistor affects the total amount of power drawn from the power supply: the larger the value, the larger the total power required to charge the capacitor. Regardless of the charge current, $P_{Resistor} = P_{Capacitor}$. Therefore, a larger charge current and larger charge time has double the effect on the power consumption.

To calculate the power used in charging a capacitor the following process must be followed:

Equation 4 - Derivation of total Power to charge capacitor equation

$$P = I^2 R$$

$$I = \frac{V_B}{R} e^{\frac{-t}{RC}}$$

$$P = \left(\frac{V}{R} e^{\frac{-t}{RC}}\right)^2 R$$

$$P_{Total} = \int_0^t \left(\frac{V e^{-\frac{t}{RC}}}{R}\right)^2 R dt = \left[-\frac{1}{2} C V^2 e^{-\frac{2t}{RC}}\right]_0^t$$

$$P_{Total} = \left[-\frac{1}{2} \times C \times V^2 \times e^{-\frac{2t}{RC}}\right] - \left[-\frac{1}{2} \times C \times V^2 \times e^{-\frac{0}{RC}}\right]$$

By calculating the instantaneous current and defining a point at which the capacitor is 'charged' it is possible to calculate t.

Where V = 3.8 V, C = 4700 µF, R = 10 Ω.

Table 1 – Instantaneous charging current against time

t	I
0	0.38
0.1	0.045263985
0.2	0.005391653
0.3	0.000642231
0.4	7.64998E-05
0.5	9.11233E-06
0.6	1.08542E-06
0.7	1.29291E-07
0.8	1.54006E-08
0.9	1.83445E-09
1	2.18512E-10

The threshold for when the capacitor is classed as charged will be defined as the time at which the instantaneous charging current drops below 10^{-9} A. (1nA), at this point the rate of charge is minimal and the stored energy is near maximum, this threshold is highlighted in orange in Table 1.

At 1 Second, the total power drawn to charge the capacitor is 0.0339 W. It must however be remembered that only half of the power drawn by the capacitor is used to directly charge the capacitor, so this resulting number must be doubled to account for the power lost in the series resistor.

When designing a circuit with large capacitance requirements the capacitors cannot be overlooked. When charging a single 1 µF capacitor to 5 V, the power requirement on the power supply is 25 µW. When charging a 4700 µF capacitor to 5V the power requirement is 0.12 W.

4. Principals of Energy Harvesting Models

4.1. Energy Harvesting Power Architecture

The first step of developing a data logger which has a very low power usage is to look at the phases of an energy harvesting system. There are two sections to this, one is taking an overall power view of the system to determine the harvesting methodology and the second being an energy conscious design approach.

This section gives a generic high level overview of how low power data loggers are designed and function. This breaks the system down into subsystems and details how the interrelation between these subsystems can be used to design a low power data logger.

4.1.1. Low Power Design Approach

The low power design approach looks at the system purely from an energy based standpoint; this view regards the functionality of the hardware as largely irrelevant and focusses on where the power will come from and how it will be provided to the load.

At its most basic level a low power data logger has the following power block diagram, as follows: Harvested Energy, Power Conditioning, Power Storage and Load. Any energy harvesting system can be broken down into these specific components

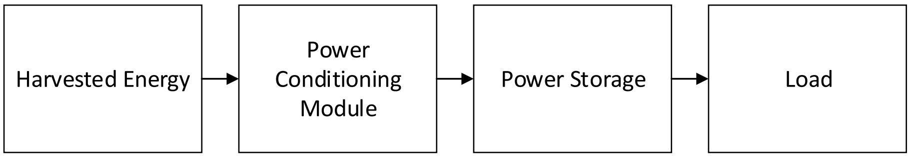


Figure 4 - Energy harvesting power block diagram

The first sub-system of an energy harvesting data logger is the harvested energy itself, whether this comes from wind, solar, Peltier or RF harvesting. The major drawback of energy supplied by any harvested source is that the power supplied is irregular, for example solar panels harvest little to no energy during the night, and wind turbines do not harvest when there is no wind.

The solution to this problem takes place in the next two sub-systems shown in Figure 4 (above), Firstly the hardware used to interface to the energy harvesting source must make the most of the power available to it. The 'power input' of the system must be highly efficient and have as little parasitic loss as is reasonably possible. Electronically, this means reducing the number of inline components such as resistors, inductors and diodes and carefully balancing any parallel components such as capacitors, tranzorb diodes or varistors, so to not waste any of the energy being harvested. The purpose of the power conditioning module is to convert the incoming power supply into an output supply which is suitable for the operation of the load. This will often either be directly powering the load, or by charging a battery which is then used to power the load.

By integrating power storage capability, the logger is able to continue to function without harvested energy, or through time windows with low levels of harvested energy. This storage capability is normally provided by conversion to a different type of energy or electronic storage. Energy conversion is less common in electronic applications, but in large scale electrical projects this can be kinetic energy (such as flywheels [62]), gravitational potential (such as holding water in a dam [63]) or chemically (such as electrolysis and hydrogen storage [64]). In smaller applications, the inefficiencies of re-claiming this energy from such storage methods are impractical. Electronic storage normally takes place using either batteries or Supercapacitors: batteries are better suited for long term compact energy storage whereas Supercapacitors are better suited for applications requiring many rapid charge and/or discharge cycles.

The final sub-system of the low power design approach is in essence what drives all of the previous steps; the load is the power requirement of the remainder of the hardware, the reason why energy is being harvested in the first place. Without knowing how much energy is required or how often the logger is required to operate, it is impossible to determine the required energy source.

4.1.2. Circuit Design

Electronic design companies often have existing designs for circuit applications which are proven and can be re-used when required, using the idiom that 'if it ain't broke don't fix it.'

This approach doesn't work well for energy harvesting applications. For example, a digital input channel in a conventional data logger would be designed to interface with the outside world, with little regards to the power consumption of the channel. Standard channels often utilise devices such as resistors and zener diodes: the resistors are used to drop the incoming current, and the zener diodes are used to drop the incoming voltage, both wasting power through the component. In an energy harvesting device, the input is often powered by the device itself, so any wasted energy is drained from the energy harvesting device's own battery.

However, there is one thing which cannot be avoided. Regardless of whether the system is mains powered, battery powered or powered by energy harvesting, the device will be subject to EMC (Electromagnetic compatibility) compliance. EMC is designed to ensure that devices within an operating environment do not interfere with one another when operational. The simplest example of EMC in practice is when a mobile phone is placed beneath an FM radio: the interference of the phone can be heard as static or pulsed clicking within the audio output of the radio.

Devices such as varistors, tranzorb diodes and gas discharge tubes are used to protect devices from transient phenomena and surges. If the correct devices are chosen the protection can be tuned to only act when such phenomena are present, and subsequently draw no power from the device power supply itself when in normal operation.

By providing a very high impedance input at a low voltage, the amount of power drawn from the supply can be minimised.

Equation 5 - Joule's Law for direct current

$$P = IV$$

Equation 6 - Joule's Law substituted for Voltage using Ohm's Law

$$P = I^2R$$

This can be seen in Equation 6 where the major component of the power drawn is the current rather than the resistance. Applying Ohms law in Equation 7 shows that an increase in resistance reduces the current flowing in a system, assuming voltage is a constant.

Equation 7 - Relationship between current and resistance in Ohm's Law

$$I \propto \frac{1}{R}$$

4.2. Definition of Units

Prior to any technical work, it is worth defining the terms which will subsequently be used frequently within this work.

Energy (Electrical Energy) – in this context means energy that has been converted from electrical potential energy. Energy is supplied by the combination of electric current and electrical potential. The SI unit of energy is Joules (J), An explanation of units of measurement for energy and power from David MacKay [65] follows:

Here's the connection to energy and power. Energy is like water volume: power is like water flow. For example, whenever a toaster is switched on, it starts to consume power at a rate of one kilowatt. It continues to consume one kilowatt until it is switched off. To put it another way, the toaster (if it's left on permanently) consumes one kilowatt-hour (kWh) of energy per hour; it also consumes 24 kilowatt-hours per day.

The joule is the standard international unit of energy, but sadly it's far too small to work with. The kilowatt-hour is equal to 3.6 million joules (3.6 megajoules).

A power of one joule per second is called one watt. 1000 joules per second is called one kilowatt. Let's get the terminology straight: the toaster uses one kilowatt. It doesn't use "one *kilowatt per second." The "per second" is already built in to the definition of the kilowatt: one kilowatt means "one kilojoule per second."*

In the context of this work the unit of measurement for energy will most likely be Watt-hours (Wh) since, whilst Joules are too small a unit of measurement, kilowatt-hours are too large to be meaningful.

Power (Electrical Power) – is the rate at which something uses energy [65]. The SI unit of power is the Watt (W)

Voltage (Electrical Potential) – Technically, one Volt is the energy of 1 joule that is consumed when electrical charge of 1 coulomb flows in the circuit. Practically voltage is the difference in electrical potential between two points, Ohm's law (Equation 8) could also be written as $I = \frac{\Delta V}{R}$. Put simply, the direct current flowing through a resistor is defined as the change in voltage across the resistance. The SI unit of voltage is the Volt (V)

Current – is the flow rate of electrical charge, or the change in electrical charge per unit time. The SI unit of current is Amperes, or Amps (A)

4.2.1. Commonly Used Equations

Equation 8 - Ohm's Law

$V = IxR$ Voltage (V) = Current (A) x Resistance (Ω) - Ohms Law

Equation 9 - Joule's Law

$P = IxV$ Power (W) = Voltage (V) x Current (A) (Joule's Law)

Equation 10 - Electrical Energy

$E = Pxt$ Electrical Energy (J) = Power (W) x Time (s)

4.3. Calculating Predicted Energy Usage Methodology

The process of calculating the predicted energy usage of the system requires an in depth look at each component within the system.

To calculate the energy usage of the system, a stepped approach is taken. The logger is designed as a modular system, each module being an assembly of multiple printed circuit assemblies, or PCAs: the calculation of energy usage will be carried out on the individual PCAs. By using the information provided from the manufacturer's datasheet, each major component will be analysed and a calculation of its predicted energy usage made in various modes.

For many components within the system, the energy calculations will be relatively simple as they will either be functional at all times or they will be electrically isolated when not in use. The more complex calculations are required when calculating the energy usage of microprocessors and other 'active' components.

Below are two demonstrations of these energy calculations.

4.3.1. MIC5225YM5 Low Dropout Regulator

The MIC5225 is a 150mA highly accurate, low dropout regulator with high input voltage and ultra-low ground current. This combination of high voltage and low ground current makes the MIC5225 ideal for a wide variety of applications including USB and portable electronics applications, using 1-cell, 2-cell or 3-cell Li-Ion battery inputs [66]

The MIC5225YM5 is used within the design to produce a local 3V3 for interface to the on board 3G Modem. This component is powered from the 3G card power supply so is only powered on during the time the 3G modem is in use.

Table 2 - MIC5225YM5 Power Usage Table

	Active Mode			Sleep Mode		
Component	Supply Voltage (V)	Current Usage (mA)	Power Usage (mW)	Supply Voltage (V)	Current Usage (mA)	Power Usage (mW)
MIC5225YM5	3V8	5	19	0	0	0

The data shown above is extracted from the schematic design and the component datasheet [66], the supply voltage is taken from the application schematic, the ground current is taken from the Maximum value for Ground Current with a 150mA load, Page 3.

The power consumption when operating is 19mW, the power consumption when the 3G device is in sleep mode is 0mW.

For single use or basic components this process is simple: however, for larger more complex components, a more detailed approach is required.

4.3.2. EFM32GG380F1024

The design is focussed around an EFM32GG380F1024 Microprocessor [67]: The Energy Friendly Micro or EFM, is a very low power processor sold by Silicon Labs based around an ARM M3 Core.

“The EFM is built on top of a low-power platform that included innovative low energy techniques, fast wake-up times and energy saving modes” [67].

To calculate the power usage of the component requires a detailed study of the component datasheet – the majority of the information is taken from Table 3.4.

Note: Where a frequency is required the 32MHz System Clock is used, or stated in the comments

Table 3 - EFM32GG Power Usage Table

	Active Mode			Sleep Mode		
Component / Feature	Supply Voltage (V)	Current Usage (mA)	Power Usage (mW)	Supply Voltage (V)	Current Usage (mA)	Power Usage (mW)
EFM32(EM0)[1]	3.3	3.8	12.54	N/A	N/A	N/A
EFM32(EM1)[2]	3.3	2	4.29	N/A	N/A	N/A
EFM32(EM2)[3]	3.3	0.002	0.007	N/A	N/A	N/A
EFM32(EM3)	3.3	0.0013	0.0043	N/A	N/A	N/A
EFM32(EM4)	3.3	0.0007	0.0023	N/A	N/A	N/A
Flash Write	3.3	14	46.2	0	0	0
GPIO (Out)	3.3	20	66	0	0	0
HFXO	3.3	0.165	0.5445	0	0	0
LFXO	3.3	0.19	0.6270	0	0	0
ADC	3.3	0.067	0.2211	0	0	0
USART	3.3	0.24	0.7920	0	0	0
UART	3.3	0.18	0.5940	0	0	0
I^2C	3.3	0.2	0.6600	0	0	0
Timer	3.3	0.28	0.9240	0	0	0
LETimer	3.3	0.00015	0.0005	0	0	0
RTC	3.3	0.0001	0.0003	0	0	0
GPIO (Idle)	3.3	0.169	0.5577	0	0	0

[1]EM0 is calculated running from an internal HFRCO (High Frequency Resistor/Capacitor Oscillator) at 14 MHz

[2]EM1 is calculated running from an internal HFRCO (High Frequency Resistor/Capacitor Oscillator) at 14 MHz

[3]EM2, EM3 and EM4 are calculated running from an external HFXO (High Frequency Crystal Oscillator) at 32 MHz.

The power modes of the processor are best described in Figure 5 below.

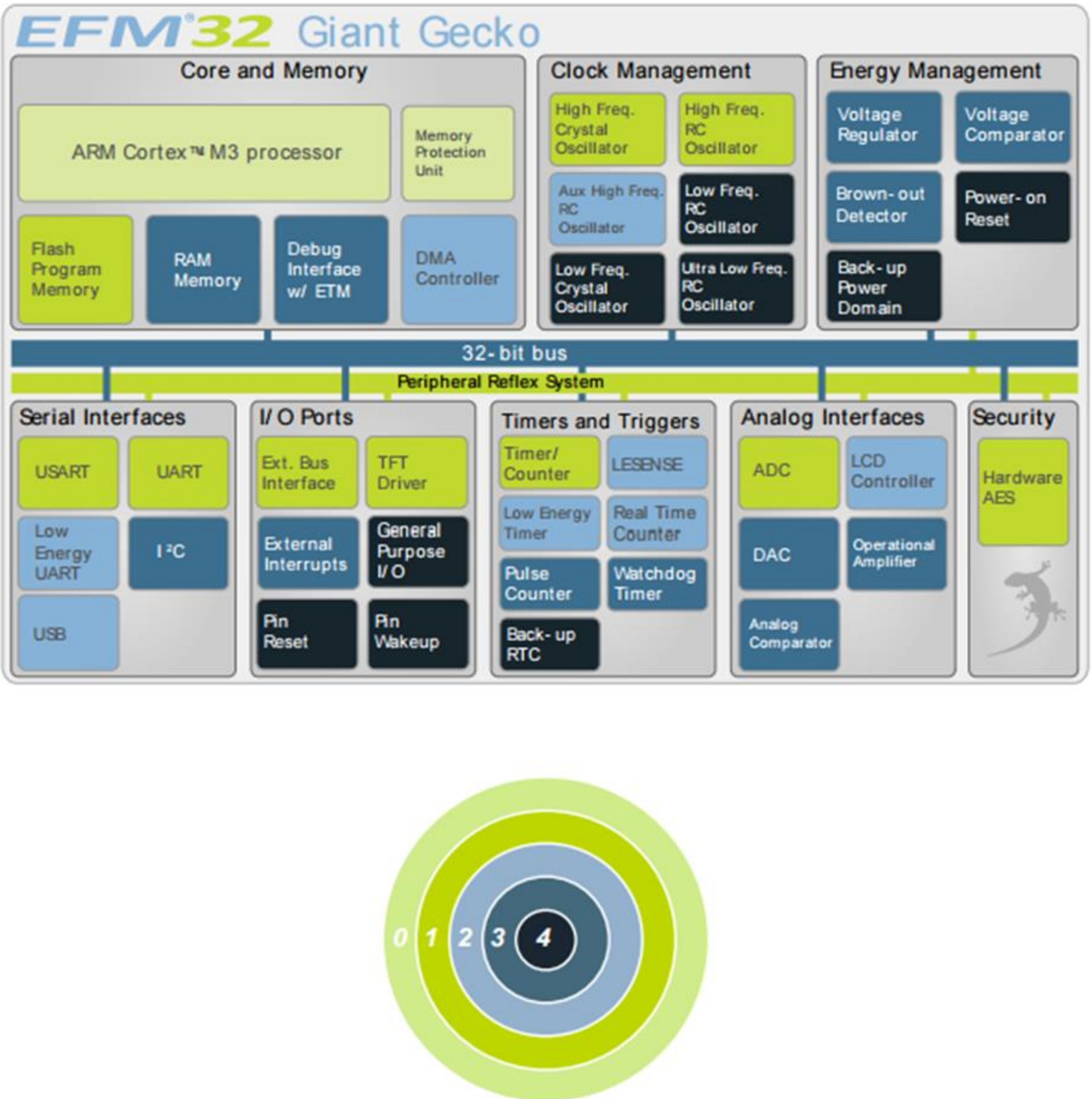


Figure 5 - Block Diagram of EFM32 and Energy Mode Indicator [67]

The diagram in Figure 5 breaks down which functions of the processor can be implemented in which energy mode; the most key parts of the processor are as follows:

In Energy Mode 4, very little of the processor's functionality is active: only power and timing related functions are possible, such as low speed oscillators, resets and GPIO triggers as either wake up or reset.

In Energy Mode 3, some peripherals are added to the device: the key functionalities in this application are the addition of RAM, I^2C and the internal voltage regulator. EM3 also introduces voltage management and a debug interface for software development.

Moving up to Energy Mode 2, further peripherals are available such as the internal Real Time Counter (RTC), USB and the DMA controller. This DMA controller allows peripherals to interface in specific ways without using the processor directly, this is often used to store data from serial links into memory. In introduction of the DMA controller enables the use of the low energy UART, USB, Timers, LCD controller and auxiliary clocks.

At Energy Mode 1 the remainder of the peripherals are available, such as external high frequency oscillators, serial interfaces and analogue inputs (ADCs). At this level the processor's proprietary Peripheral Reflex System is available: this allows a range of peripherals to reflexively cause an action on a different peripheral, again this does not require the processor, but the routes are set. The best example of how this system could be used in other applications (but not this work) is that the PRS can the receive data from a serial port, USB or other external source and then directly display that data on an LCD Display without using the processor

The highest level of functionality enables all previous levels of functionality, but also enables the Cortex M3 core which is contained within this processor: at this power level, the EFM32 can be used as a generic Cortex M3.

By using the applicable power levels within the EFM32, the power consumption of the overall device can be significantly reduced. The method used to determine the required energy mode is to determine the required peripherals for a specific task, and only using the lowest energy mode applicable to those needs.

Outside of the main processor there will be differing levels of functionality for each component within the logger. In addition, for certain components, there will be multiples of the device running at certain times and, although this will not be exactly possible to determine until software has been completely written, it is possible to make estimations during certain conditions: in all cases a realistic or worst case condition will be taken.

5. Data Extraction

The previous section of this thesis looked at power modelling of a logger in a generic sense: this next section focusses on a case study of the Energy Harvesting Data Logger itself. Detailed below is the hardware makeup of the logger as well as the power consumption of the various parts of the logger. This section also details the energy modes in which the logger will function.

The energy harvesting data logger is designed to meet the requirements detailed previously in section 2.1. The logger is designed to be a battery powered device which is supplemented by energy harvested from external sources, in this case, solar power or a wind turbine. The logger has been designed to be low powered when operational and draw close to no power when in a sleep state.

By both reducing the sleep current requirements of the hardware and using harvested energy to re-charge to battery, it is intended that it will be possible to operate the Energy Harvesting Data Logger persistently without a requirement to change or charge batteries: the net result of this would be to reduce the maintenance required on the logger and subsequently reduce track access time and also expanding the locations in which monitoring can occur.

Further to this, the Energy Harvesting Data Logger incorporates a 3G modem alongside its local data storage to allow the offload of data to a remote server. This further removes the need to attend the site of the logger to retrieve the data and opens the door for real time data transfer where necessary. From preliminary investigation it is thought that the 3G modem would be the component with the highest power requirement of the Energy

Harvesting Data Logger; this thesis aims to investigate if this preliminary theory is accurate and what impact this has on the frequency of use, of the 3G modem, if the unit is to remain a feasible deployment when considering power usage.

Accompanying the Energy Harvesting Data Logger hardware is a combination of embedded software and a PC based configuration utility: these two applications work together as part of an Energy Management Utility to assist with the configuration and practicality of deployment of the Energy Harvesting Data Logger. The Energy Management Utility has three separate sections: a functional configuration utility, a power configuration utility and an iterative improvement utility.

The configuration utility is similar to those used by conventional data loggers; it allows the user to configure the various input channels, setup simple filtering or thresholds and implement a very simple rules engine. For the Energy Harvesting Data Logger, this section of the utility also allows the user to set up the 3G connectivity options for data offload.

The power configuration utility will work alongside the configuration utility to assist the user in an initial estimate for deployment feasibility: the intention is to analyse the configuration chosen in the configuration utility and combine it with further information such as battery configuration, energy harvesting source and geolocation to give a yes or no answer as to the feasibility of the deployment. The most obvious causes for a no answer would commonly be due to scenarios such as ‘too high frequency of server update’ or ‘too low level of harvested energy’: in such situations, the utility should suggest alternative configurations to assist the user in achieving a workable solution.

The final section of the energy management utility should work as an iterative improvement utility. This utility should work in tandem with the onboard firmware to develop a 'history' of the systems power level as well as monitoring the power requirement of the various operating functions within the unit. These readings aim to give an accurate picture of the logger in operation enabling the utility to iteratively improve the deployment. Coupled with this, each set of data received in this way allows the utility to feed back to improve the accuracy of future developments. At a simple level, for each reading a delta is made against the predicted reading: this delta is then added to a moving average weighted towards the existing value to remove any anomalous results. If the predicted value is significantly and persistently less or more than the recorded value, future predicted values will be amended accordingly. This is detailed further in section 7.2.1.

This section is broken down into 5 sections. The first section details the proposed hardware design, showing the proposed layout and block diagrams of the constituent parts of the logger.

The second section studies the components in the design one at a time and determines their power consumption whilst in various operations, primarily Active and Sleep if the devices have such modes.

The third section details the operating modes of the logger, looking at the modes of the logger as a whole rather than at component level: this section results in a table of the power usage by each 'board' in the design for each operating mode.

The fourth section details the time parameters of the various operating modes. Each of the operating modes in section 3, section 4 contains an estimate for the 'on-time' of that mode.

The fifth section details the parameters of the logger which are required for the energy model to be configured around, split into three parts. These parameters detail the Hardware Configuration, i.e. which boards are fitted to the logger being modelled, and the Recording Parameters, i.e. how often the logger performs certain actions. These parameters are key to the accuracy of the energy model. The final part of this section details the power storage parameters of the logger, i.e. the size of the battery and whether energy harvesting is available.

5.1. *EHDL Hardware Description*

The makeup of the EHDL is modular in design, the architecture is based around a central processing board with optional expansion boards which allow the logger to monitor different channels

5.1.1. EHDL CPU Board

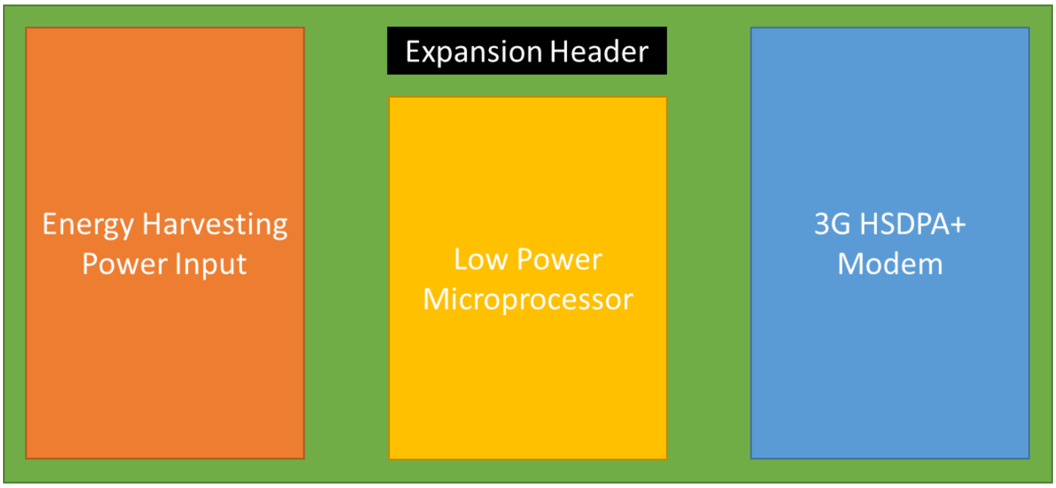


Figure 6 - EHDL CPU Block Diagram

The EHDL CPU, Figure 6, is divided into three main sections, the energy harvesting power input, A low power microprocessor and 3G HSDPA+ modem. The CPU board also has an expansion header which is used to interface with additional boards for the capture of data.

The additional cards can be stacked six boards high with addressing and power control connections build into the header.

On the board are various components which support the processor and 3G modem. Where possible, power switching is used to control the power to various devices: this is either through the use of high side switches or through the use of the processor's own output pins being used to power the devices – the only flaw with this second method is that if this method is used too frequently then too much power could be drawn through the processor, leading to the processor overheating.

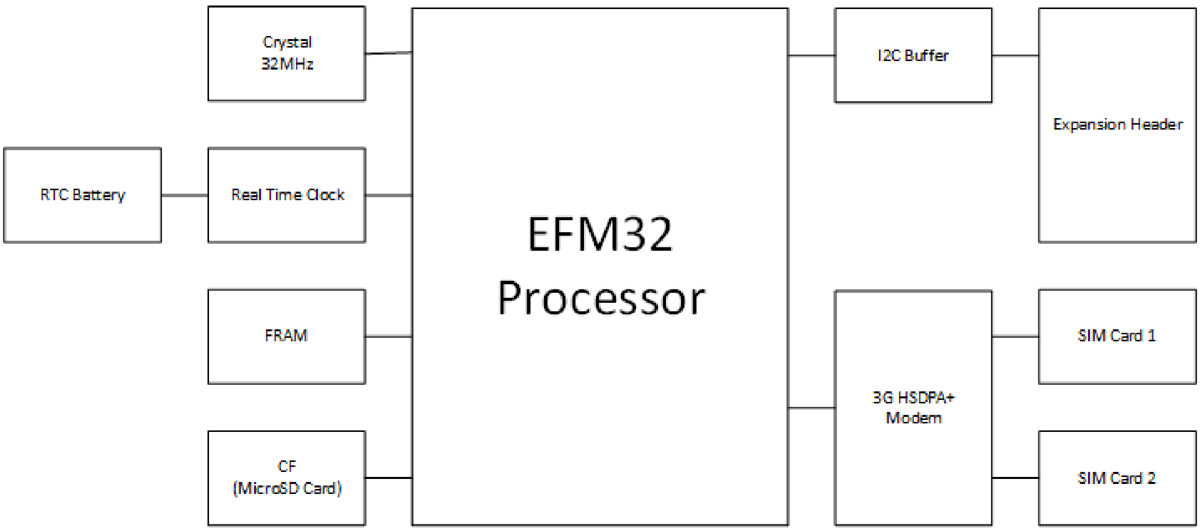


Figure 7 - EHDL CPU Detailed Block Diagram

The Energy harvesting power input circuitry is a selection of IC's which collect power from three major sources. The selection of power sources is designed to typify a standard power input from certain types of Energy Harvesting technology. Details on the types of energy harvesting and why such options have been taken are contained in section 3.3.

The first power source, a 5V DC input can be used: this power source is designed to harvest power from a USB connection, the major purpose of this input is to enable a service engineer to charge the device from a laptop as well as enabling the unit to be powered whilst any configuration takes place.

The second power source is a 12V DC input, this input is designed for interfacing to a wind turbine. The power supply output from wind turbines currently available on the market is roughly equally split between DC and AC, however if the circuit were designed for AC, it would not be possible to harvest from a DC source whereas, although inefficient, an AC supply can be simply converted to DC through a bridge rectifier.

The final power supply is a 17.5V DC input: this input is designed for interfacing to a solar panel. The harvesting device can operate in a MPPT mode (see section 3.3) allowing for the maximisation of the power harvested from the solar cell.

Built into the power input stage is a prioritised PowerPath controller: this device is designed to select which power input to use (depending on a configured priority) and prevents reverse and cross conduction currents. The configured priority is as follows: USB, then Wind, then Solar.

This configuration is designed with the following assumptions. If a USB power supply is present it is assumed that the supply capacity is 'unlimited' i.e. if a USB supply is present it is being powered from a PC or other bulk power supply. Therefore, the USB is prioritized over all supplies. The decision between solar and wind power was based on the general condition of the different supplies. If wind is present, the amount of power produced will be greater than that provided from the solar panel, whereas at all times a certain level of solar power will be collected (it is acknowledged that this is a wide statement, which doesn't include periods of darkness – in this configuration the USB and wind power during darkness will be prioritised over the solar anyway).

The next stage of the power input stage is the battery charger. This component takes the output from the harvesting devices and conditions it for charging the Li-Ion batteries within the system. Finally, the battery voltage is converted from a storage level of 7.2 V to a 5 V working voltage for the remainder of the system.

The processor section is based around a low power microprocessor and its supporting peripherals: in brief, the peripherals are a 32MHz crystal, local FRAM, a compact flash (MicroSD) slot, a battery backed Real Time Clock (RTC) and power switches allowing the processor to supply or remove power from various parts of the circuit.

The 3G HSDPA+ modem is a self-contained device but does have a few supporting peripherals – these are required to enable the device to fully function. The supporting peripherals are a voltage level shifter for the comms interface between the modem and processor (1.8 V for the modem interfacing to the 3.3V processor). The modem also has two external SIM card slots so that the network can be switched to give some versatility. Finally, the modem has a supercapacitor to provide power for the large current loads drawn when the radio is transmitting.

5.1.2. EHDL Digital and Analogue Input Board

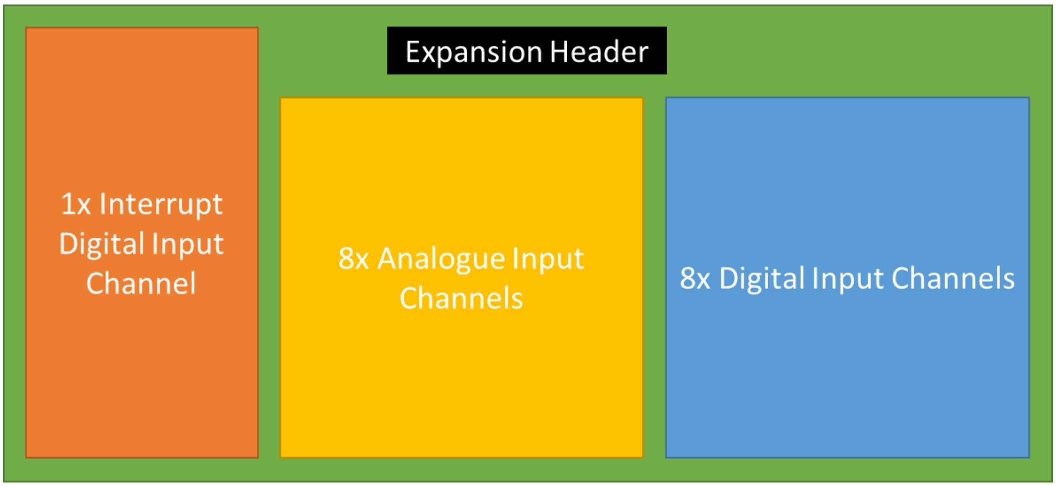


Figure 8 - Digital & Analogue Input Card Block Diagram

The digital and analogue input card, Figure 8, is a multiple input card formed of 8 Digital Input Channels, 8 Analogue Input Channels and a single Digital Interrupt input channel. Power to the DAI card is controlled via a high side switch which is driven from the CPU card.

Data from the digital input channels is collected by a I/O expander which is then read back to the CPU over a serial link which exists in the expansion header. For each channel a 5V power supply output is provided to power a sensor if required.

Data from the analogue input channels is collected by a 12 Bit ADC which is then read back to the CPU over a serial link. For each channel a 5V Power Supply output is again provided to power a sensor if required.

Finally, the DAI card has a Digital Interrupt input channel. This channel can be independently switched off and on by the CPU card as well as being directly read back to the Processor on the main CPU board.

The expansion header on the DAI card is the same as the header on the CPU board but the header exists on both sides of the board, allowing the DAI card to be stacked with other I/O cards: the majority of pins are 'straight through' but the addressing and power enable pins are 'incremented' in that the boards pass and increment the addressing up through the boards.

5.1.3. EHDL Strain Gauge Amplifier Board

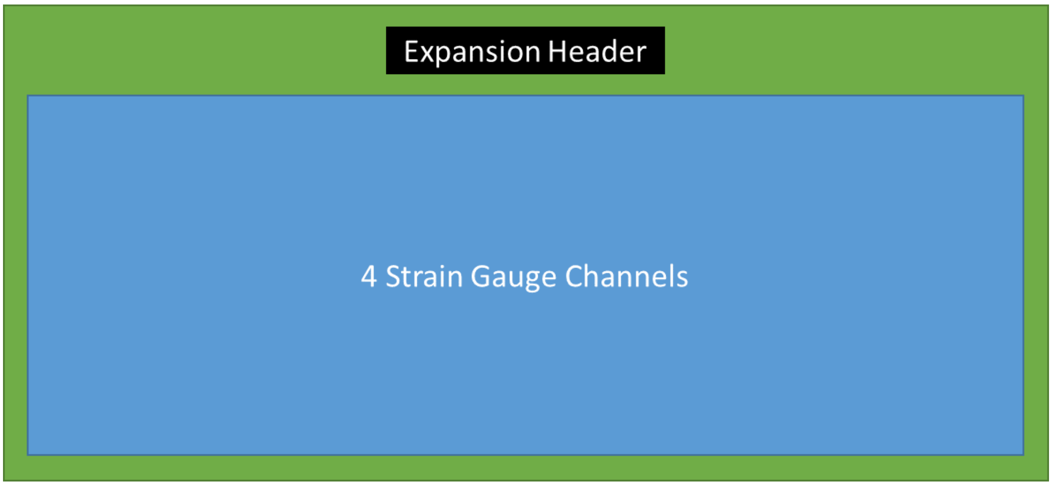


Figure 9 - Strain Gauge Card Block Diagram

The second type of expansion card is the Strain Gauge Amplifier Card, Figure 9; this card is made up of 4 independent Strain Gauge Amplifier drivers. Power to the SGA card is controlled via a high side switch which is driven from the CPU card.

Each amplifier can be configured as either a full bridge or half bridge configuration. The data is read back from the strain gauge amplifiers over a serial link back to the processor via the expansion header

The expansion header on the DAI card is the same as the header on the CPU board but the header exists on both sides of the board, allowing the DAI card to be stacked with other I/O card: the majority of pins are ‘straight through’ but the addressing and power enable pins are ‘incremented’ in that the boards cascade the addressing up through the boards.

5.1.4. Existing Firmware Description

This section focusses on the existing firmware of the system, how it is designed to function and how this affects its energy usage. The system exists in a sleep state unless performing other functions.

Figure 10 shows the software flow diagram of the existing data logger firmware. For each stage in the flow diagram a description is provided.

Wake from Sleep Mode

The Data Logger will wake from sleep when the timer overflows, the timer is set as the last action of the previous loop. Upon waking, the logger will call the relevant functions required by this loop. The following options can be set

- Scan I/O Cards (1-6) – these are 6 separate options (any combination is an acceptable option)
- Data Transmission – less frequent

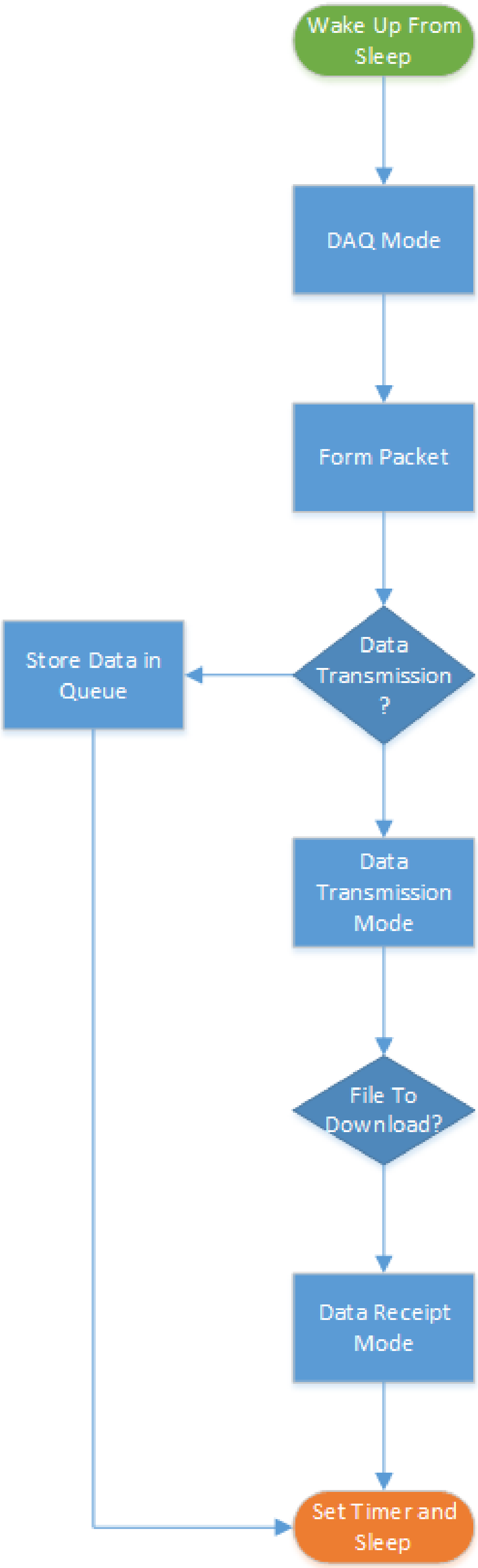


Figure 10 - Software Flow Block Diagram

DAQ Mode

DAQ mode is a sweep of all of the required I/O cards. This breaks down into the following steps (repeated for each card)

- Power Up I/O Card
- Read/Confirm Hardware Type
- Configure Additional I/O for Card Type
- Read Data from Card
- Power Down Card
- Process Data for Event triggers
- Store Data to external memory/SD card
- Reset GPIO for I/O cards

Data Transmission

If a data transmission is required, the logger must perform the following functions

- Confirm sufficient power is available in the battery – This is necessary to prevent the remaining power stored in the battery being drawn into the storage capacitor and powering the system down)
- Check SIM card is installed – The purpose of this is to prevent an ‘expensive’ (in terms of power) charge of the 3G Module to later discover no link is possible.)
- Power up the 3G module – This includes managing the system power whilst the capacitors are charging
- Initialise Modem communication to the 3G module – a multiplexed serial link is required to interface with the 3G Module, this requires a setup phase.

- Once a connection to the server is achieved, transmit data stored in queue to the remote server – the 3G Module is used to manage the web connection, data is passed in a raw format to the 3G module before being packaged as TCPIP or UDP packets (as appropriate).
- Check returning messages for updates or file downloads (and download if appropriate)
- Hang up data call
- Power down 3G module
- Clear Local Queue

File Download

This has been covered within the functionality of the Data Transmission, but technically the check for file download and subsequent download of the files are within this section (the energy usage of the system is increased if a file is downloaded).

Set Timer and Sleep

The final phase is to set the sleep timer and set the unit to sleep mode. Within the sleep phase the logger uses the configuration parameters to determine what functions the next cycle of the logger should perform.

If the scan frequency for the cards is the same and more frequent that the data transmission frequency the following cycle will occur

- …
- Scan I/O Cards
- Scan I/O Cards and Upload Data

- Scan I/O Cards
- …

At the end of each cycle, the logger will set a flag for each required function in the next cycle.

5.2. Power Data Tables

5.2.1. EHDL CPU Board

Below is the complete listing of all components analysed for the energy consumption. This is an extended version of the data tables detailed in in section 4.3.

Table 4 - EHDL CPU Board Power Usage Table

		Running			Sleep		
Device	Quantity	Voltage (V)	Current (mA)	Power (mW)	Voltage (V)	Current (mA)	Power (mW)
EFM32GG							
EFM32 (EM0)	1	3.3	3.8	12.5400	0	0	0
EFM32 (EM1)	1	3.3	1.3	4.2900	0	0	0
EFM32 (EM2)	1	3.3	0.002	0.0066	0	0	0
EFM32 (EM3)	1	3.3	0.0013	0.0043	0	0	0
EFM32 (EM4)	1	3.3	0.0007	0.0023	0	0	0
Flash Write	1	3.3	14	46.2000	0	0	0
GPIO (per)	25	3.3	20	66.0000	0	0	0
HXFO	1	3.3	0.165	0.5445	0	0	0
LFXO		3.3	0.19	0.6270	0	0	0
ADC	1	3.3	0.067	0.2211	0	0	0
USART	1	3.3	0.24	0.7920	0	0	0
UART	1	3.3	0.18	0.5940	0	0	0
I2C	1	3.3	0.2	0.6600	0	0	0
TIMER	1	3.3	0.28	0.9240	0	0	0
LETIMER		3.3	0.00015	0.0005	0	0	0
RTC	1	3.3	0.0001	0.0003	0	0	0
GPIO (Idle)	1	3.3	0.169	0.5577	0	0	0
PRS	1	3.3	0.089	0.2937	0	0	0
3G Modem[1]							
PHS8-E	1	3.8	635	2413.0000			
LD39300	1	5	1.2	6.0000	5	0.001	0.005

MIC5225YM5	1	3.8	5	19.0000	5	0.1	0.5
FSA2567MPX	2	3.8	150	570.0000	0	0	0
MIC2091	1	3.3	0.00011	0.0004	0	0	0
SN74AVC4T774	2	3.8	8	30.4000	0	0	0
PSU Components							
LTC3112	1	5	0.000075	0.0004	0	0	0
MCP1703	1	5	0.005	0.0250	0	0	0
Peripheral Components							
FM25V20	1	3.3	3	9.9000	0	0	0
SD Card	1	3.3	45	148.5000	0	0	0
PCA9517D	2	3.3	1	3.3000	0	0	0
M41T93	1	N/A	N/A	N/A	N/A	N/A	N/A

[1] It should be noted that no calculation is made at this stage for the energy lost in charging and discharging the capacitors for peak load of the 3G Modem.

5.2.2. EHDL Digital and Analogue Input Board

Below is the complete listing of all components analysed for the energy consumption

Table 5 - EHDL DAI Power Usage Table

		Running			Sleep		
Device	Quantity	Voltage (V)	Current (mA)	Power (mW)	Voltage (V)	Current (mA)	Power (mW)
ISF0505DEC	2	5	25	125.0000	5	0	0
PCA9555D	1	5	160	800.0000	5	0	0
MAX11615	1	5	0.33	1.6500	5	0	0
HCPL-4701	5	5	0.5	2.5000	5	0	0
ACPL-C87x	8	5	27	135.0000	5	0	0
AD820	8	5	15	75.0000	5	0	0
PCA9517D	1	5	1	5.0000	0	0	0
DI Connected (On)	8	5	0.005	0.0250	0	0	0
AI Connected (On)	8	5	0.5	2.5000	0	0	0

5.2.3. EHDL Strain Gauge Amplifier Board

Below is the complete listing of all components analysed for the energy consumption

Table 6 - EHDL SGA Power Usage Table

		Running			Sleep		
Device	Quantity	Voltage (V)	Current (mA)	Power (mW)	Voltage (V)	Current (mA)	Power (mW)
AD7195	4	5	6.9	34.5000	0	0	0
4x SG Connected	4	5	7.142857143	35.7143	5	0	0
PCA9517D	1	5	1	5.0000	0	0	0
PCA9555D	1	5	160	800.0000	5	0	0
HEF4011BT	1	5	0.0000075	0.0000	5	0	0

From the information collected within the previous three sections it is now possible to generate power usage values for each device on each board depending on what functionality is required in any given operating mode.

5.3. Operating Modes

The next section focusses on converting the individual power usage values for each individual component as generated in Sections 4.3 and 5.2 to a system wide power usage. This cannot be carried out as a single calculation since the logger functions in various modes depending on the current state. In practice there are three main power modes and a handful of more minor modes which are covered below: a synopsis of each mode is contained here to explain how they relate and additional information is provided in the structure of the following section.

The three major power modes are Sleep (section 5.3.1), Data Acquisition or DAQ (section 5.3.2) and Data Transmission (section 5.3.3). Information is also provided for Data Receipt (section 5.3.4).

Sleep mode is when the logger is performing no other function, and acts as the rest state of the logger between functions. Sleep mode is the lowest power mode of the logger, but there is no functionality beyond incrementing the real time clock and monitoring for the next function to be required.

Data Acquisition or DAQ is where the logger collects information, this is the mode where the logger performs its primary functionality of monitoring the required channels. This phase covers the collection and storage of data to the local storage device.

Data transmission is the function which takes the data stored in the local memory and transmits it to a remote web server – this data transfer will be minimal, but the initialisation phase of the module will be significant

The Data Receipt mode is the inverse of the data transmission mode, where data such as configuration or firmware updates is passed back to the logger over the 3G connection.

The following sections expand the operating modes detailed above. The method taken here is to approach the system board by board. For each function, the energy requirement from the CPU, DAI, SGA Boards are taken into account after looking at the state of each component in each mode. During the data transmission section there is an additional heading which covers the power usage when the 3G module is transmitting data. This burst of power must be averaged across the periodicity of the data burst; this is covered in the appropriate section below.

5.3.1. Sleep

The first Operating mode of the logger is Sleep mode. Sleep Mode is the most commonly used mode of the EHDL. In Sleep mode the functionality and subsequently the power usage is reduced to a minimum. In sleep mode all devices external to the main processor are turned off or electrically isolated. The processor itself runs in a reduced energy mode, maintaining the minimum functionality to wake on interrupt or run a timer to trigger the next data acquisition.

The Sleep power of the system is calculated simply by totalling the sleep current of all of the components of the board.

The total power estimation for this is as follows:

CPU Board

The Sleep power for the CPU card is a sum of the required energy modes of the processor, coupled with the remaining components of the board.

For Sleep mode the processor is required to run in energy mode EM4, the lowest energy mode. The reason the processor can be operated in such a way is due to the external Real Time Clock. By using the external RTC, the processor does not need to power its own internal RTC. By implementing functions within the external RTC, timers can be used to wake the processor when the next function is due. The processor can also be woken from EM4 mode when an interrupt is toggled. All other processor peripherals will be powered off during sleep mode.

The power devices which directly power the processor will be operating in an active state but all other devices will be in sleep mode. This means that the DCDC between the battery and the processor and the linear regulator providing 3.3 V to the processor are both ‘Active’.

The other external supporting devices will be powered down during sleep mode. The Real Time Clock device will be powered by its own supporting battery.

The 3G Module will be powered off during sleep mode, all components are un-powered with the exception of the local power supply regulator for the 3G module, which is in ‘Sleep’ mode.

Power Usage Table

Device / Mode	Voltage	Current (mA)	Power (mW)
FM32(EM4)	3.3	0.0007	0.0023
LTC3112	5	0.000075	0.0004
MCP1703	5	0.005	0.0250
LD39300	5	0.001	0.005

The total estimated power usage of the CPU in full sleep mode is 0.032685 mW or 32 µW.

DAI Board (Interrupt Disabled)

In sleep mode the main power to the DAI board is completely isolated but the board does contain a single interrupt input (see below)

DAI Board (Interrupt Enabled)

if the interrupt channel is enabled the power consumption is as follows. In sleep, the Interrupt channel requires the following devices to be active.

- ISF0505A (at minimum load 20 mA – 25 mA current draw)

- HCPL-4701 (To perform the signal isolation)
- DI Connected (this is the power consumption drawn when the logger provides power for the sensor to use as the digital input back to the logger)

Power Usage Table

Device / Mode	Voltage	Current (mA)	Power (mW)
ISF0505DEC	5	125	125
HCPL-4701	5	0.5	2.5
DI Connected (On)	5	0.005	0.025

The total calculated power usage of an interrupt channel on the EHDL is 127 mW. The total sum of the power usage of the DAI card in this setup comes from the interrupt channel: this is covered in detail in section 8.1.2.

SGA Board

The main power to the SGA board is completely isolated. Therefore, a negligible amount of power is drawn.

Summary

To summarise, the power usage of the device in Sleep mode is

Table 7 - Power Usage in Sleep Mode

Circuit Section	Power Usage (mW)
CPU	0.033
DAI	0.00
Interrupt Channel	127.00
SGA	0.00

5.3.2. DAQ

The second Operating Mode of the logger is Data Acquisition or DAQ which is when the EHDL acquires data from its sensors. The software control from this is that the logger reads data sequentially from each board in turn. Therefore, the energy for each board within a configuration can be summed.

CPU Board

The DAQ power for the CPU will depend on the card being read. There will be a minimum required for all cards; this common power requirement will be detailed here: where specific channels are required for a specific card they will be detailed in the appropriate section.

To operate in DAQ mode of any card, EM0 – the highest energy mode is required, this is due in part to the required peripherals, but also that the cortex M3 processor is required. Detailed information on energy modes and their meanings are shown in section 4.3.2.

The required processor peripherals for DAQ mode are:

- HFXO – The External Bus devices on USART, UART and I^2C require the use of a higher speed/accuracy external clock.
- ADC – The on board ADC reads the Battery Voltage, DCDC Loads and charging states
- USART – The USART is used to interface with the RTC, FRAM and SD Card.
- GPIO – The GPIO are used to control various devices on the board, most commonly to power the devices on, directly or through power switches.
- RTC – the on board RTC is updated from the RTC chip on the board- this reduces the requirement to run an accurate on board clock, and improves the overall accuracy.

Additionally, the power devices which directly power the Processor will be operating in an active state, all other devices will be in sleep mode. This means that the DCDC between the battery and the processor and the linear regulator providing 3.3 V to the processor are both 'Active'.

External to the main processor, the following peripherals will be powered up during a DAQ:

- FM25V20 – The local FRAM chip is required to retrieve any configuration information as well as store any additional information.
- SD Card – the SD card is required to store the acquired data. The data is stored ready for transmission.
- The 3G Module will be powered off during DAQ mode, all components are un-powered with the exception of the local power supply regulator for the 3G module, which is in 'Sleep' mode.

Power Usage Table

Device / Mode	Voltage	Current (mA)	Power (mW)
EFM32 (EM0)	3.3	3.8	12.5400
HXFO	3.3	0.165	0.5445
ADC	3.3	0.067	0.2211
USART	3.3	0.24	0.7920
RTC	3.3	0.0001	0.0003
GPIO	3.3	0.169	0.5577
LTC3112	5	0.000075	0.0004
MCP1703	5	0.005	0.0250
LD39300	5	0.001	0.005
FM25V20	3.3	3	9.9000
SD Card	3.3	45	148.5000

The total estimated power usage of the CPU in DAQ mode is 174.759 mW.

DAI Board

When scanning the DAI boards, the additional processor peripheral which is required is an I^2C Bus. Due to the processor already operating in EM0, no additional power is drawn from the energy mode of the processor. The additional power required is:

Device / Mode	Voltage	Current (mA)	Power (mW)
U1 / I^2C	3.3	0.2	0.66

When the board is powered on, the following devices are powered on:

Device	Component Ident	Quantity	Voltage (V)	Current (mA)	Power (mW)
ISF0505DEC	DC101, DC102	2	5	25	125.0000
PCA9555D	IC131	1	5	160	800.0000
MAX11615	IC7	1	5	0.33	1.6500
HCPL-4701	ISO101-105	5	5	0.5	2.5000
ACPL-C87x	ISO201, 211, 221, 231, 241, 251, 261, 271	8	5	27	135.0000
AD820	U201, 211, 221, 231, 241, 251, 261, 271	8	5	15	75.0000
PCA9517D	IC132	1	5	1	5.0000
DI Connected (On)	N/A	8	5	0.005	0.0250
AI Connected (On)	N/A	8	5	0.5	2.5000

The total estimated power usage of the DAI boards in DAQ mode is 2770.0 mW. This is a significant power requirement and greater than initially expected. The large power requirement of the DAI card will impact on the maximum scan rate of the system.

SGA Board

When scanning the SGA boards, the additional processor peripherals which are required are an I^2C Bus and a USART channel. Due to the processor already operating in EM0, no additional power is drawn from the energy mode of the processor. The additional power required is:

Device / Mode	Voltage	Current (mA)	Power (mW)
U1 / I^2C	3.3	0.2	0.66
U1 / USART	3.3	0.24	0.7920

When the board is powered on, the following devices are powered on

Device	Component Ident	Quantity	Voltage (V)	Current (mA)	Power (mW)
AD7195	U101, U111, U121, U131	4	5	6.9	34.5000
4x SG Connected	N/A	4	5	7.142857143	35.7143
PCA9517D	IC212	1	5	1	5.0000
PCA9555D	IC211	1	5	160	800.0000
HEF4011BT	U213, U214	1	5	0.0000075	0.0000

The total estimated power usage of the SGA boards in DAQ mode is 1087.3092 mW. This is less than the DAI Card, but still significant.

Summary

To summarise, the power usage of the device in DAQ mode is:

Table 8 - Power Usage in DAQ mode

Circuit Section	Power Usage (mW)
CPU	174.76
DAI	2770.00
Interrupt Channel	127.00
SGA	1087.31

5.3.3. Data Transmission

The next operating mode for the CPU is the mode used when the EHDL is transmitting data over the air to a remote server. This is predicted to be the highest single use of energy in the system. It is assumed that for transmission of data the energy used by the rest of the system is the same as drawn when the system is in sleep mode. The processor will be powered on in a suitable energy mode to allow for the interface to the 3G module. This will require EM0, a UART interface and access to the SPI FRAM and SD Card

The remainder of the power requirement will be drawn by the 3G module itself and its supporting components. The most difficult part of the system to model is the power storage capacitor required to enable 3G communications. The peak current draw for the 3G module is 2 A, this is provided from the 3.8 V capacitor (C203). As a general rule, capacitors have been 'ignored' as part of this power model. The capacitors used at this stage are too large to be ignored – see Section 3.7.

The total power calculation for this mode is as follows:

CPU Board

To perform a data transfer, the processor will require use of the Cortex M3 processor, to enable this the processor must be in EM0 this is the highest energy mode, detailed information on energy modes and their meanings are contained previously in section 4.3.2.

The required processor peripherals are

- HFXO – The External Bus devices on USART, UART and I^2C require the use of a higher speed/accuracy external clock

- USART – The USART is used to interface with the RTC, FRAM and SD Card.
- UART – The UART is used to interface with the 3G Module.
- GPIO – The GPIO are used to control various devices on the board, most commonly to power the devices on, directly or through power switches.
- RTC – the on board RTC is updated from the RTC chip on the board- this reduces the requirement to run an accurate on-board clock and improves the overall accuracy.

The power devices which directly power the Processor will be operating in an active state: all other devices will be in sleep mode. This means that the DCDC between the battery and the processor and the linear regulator providing 3.3 V to the processor are both 'Active'.

The following external peripheral devices will be powered up during a data transmission:

- FM25V20 – The local FRAM chip is required to retrieve any configuration information as well as store any additional information.
- SD Card – the SD card is required to read the acquired data. The data is stored ready for transmission.

The 3G Module will be powered on during DAQ mode – all peripherals are fully active. Also, when the 3G Module is powered on the Capacitor C203 will be fully charged by the power supply.

Power Usage Table

Device / Mode	Voltage	Current (mA)	Power (mW)
EFM32 (EM0)	3.3	3.8	12.5400
HXFO	3.3	0.165	0.5445
USART	3.3	0.24	0.7920
UART	3.3	0.18	0.5840
RTC	3.3	0.0001	0.0003
LTC3112	5	0.000075	0.0004
MCP1703	5	0.005	0.0250
PHS8-E	3.8	250	950
LD39300	5	1.2	6.0000
MIC5225YM5	3.8	5	19.0000
FSA2567MPX	3.8	150	570.0000
MIC2091	3.3	0.00011	0.0004
SN74AVC4T774	3.8	8	30.4000
FM25V20	3.3	3	9.9000
SD Card	3.3	45	148.5000
Storage Capacitor (C203)	3.8	Calculated Externally	0.067

The calculation for the power consumption to fully charge the capacitor is contained within section 3.7, once the capacitor is initially charged all power drawn is that of the individual functionality (i.e. power will be drawn from the capacitors then recharged from the battery).

The total estimated power usage of the CPU board in Data Transmission mode is 1748.364 mW (taken from the sum of the Power column above).

DAI Board

The main power to the DAI board is completely isolated, the board does however contain a single Interrupt input, if the interrupt channel is used the same calculation is used as in section 5.3.1

SGA Board

The main power to the SGA board is completely isolated. Therefore, a negligible amount of power is drawn.

Data Transmission

When the 3G module transmits data the current consumption of the PHS8-E increases to a maximum value of approximately 2A [68]. The management strategies for such a large power burst, as shown in Figure 11, are contained explained in the Application Note for the Wireless Module [69].

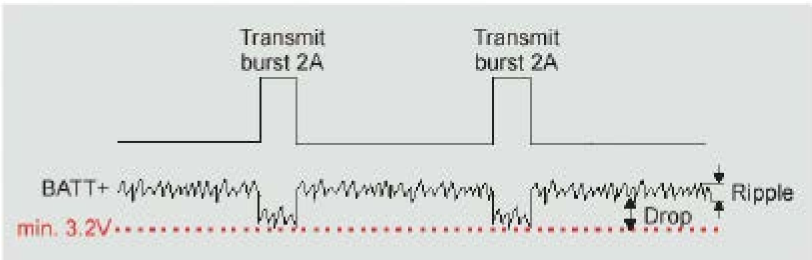


Figure 11 - 3G module Transmit Current and Battery Voltage

The application note details the difference between a low current regulator IC and a high current regulator and their effect on the size of the capacitor.

The low current regulator requires a larger capacitor on the inside of the regulator: the capacitor is used to provide the peak current pulses to the 3G module and is then re-charged through the linear regulator.

The high current regulator requires a smaller capacitor on the outside of the regulator, the capacitor is used to hold up the primary power rail against the surges, and the current pulses are pulled through the regulator, thus the requirement for a higher current capability. Either way, the capacitor is still a high value at 2200 µF.

Equation 11 - Charge Comparison

$$\text{Smaller Capacitor/Higher Voltage} – 0.0022\ F \times 5\ V = 0.011\ C$$

$$\text{Larger Capacitor / Lower Voltage} – 0.0047 \times 3.8\ V = 0.01786\ C$$

Equation 12 - Energy Storage Comparison

$$\text{Smaller Capacitor/Higher Voltage} – \frac{1}{2} \times 0.0022 \times 5^2 = 0.0275$$

$$\text{Larger Capacitor / Lower Voltage} – \frac{1}{2} \times 0.0047 \times 3.8^2 = 0.0339$$

The charge requirement and Energy Storage requirement of the inside capacitor are larger. This however isn't the only consideration – if the smaller external capacitor were used it would be directly connected to the 5 V power rail. The major side affect of such a setup would be that if the capacitor was fully discharged, then the battery would be exposed to the full charge current of the capacitor. If the internal capacitor were used, although there is a larger energy requirement to charge the capacitor, it can be controlled by the processor. This enables the larger capacitor to only be charged when sufficient power is available.

The actual transmission pulse is 2 A during a transmit burst. Between pulses the current drops down to the standard current consumption of the 3G module. The Pulse pattern is 577 µs bursts every 4.615 ms (see Figure 12). To calculate the draw on the overall system battery an average will be taken as an $\frac{1}{8}^{th}$ of the maximum power draw, in this case, 0.25 A on average. At the operating voltage of 3.8 V the average power draw of the actual transmission is 0.95 Watts.

Function	Signal name	IO	Signal form and level	Comment
Power supply	BATT+_GSM	I	$V_I$max = 4.2V $V_I$norm = 3.8V $V_I$min = 3.3V during Tx burst on board Imax ≈ 2A, during Tx burst (GSM)  n Tx = n x 577µs peak current every 4.615ms	Lines of BATT+ and GND must be connected in parallel for supply purposes because higher peak currents may occur. Minimum voltage must not fall below 3.3V including drop, ripple, spikes.

Figure 12 - Transmit Current Pattern

The total sum of the power used during transmission is the sum of the operating power and the transmission power requirement.

$$1.748364\ \text{W} + 0.95\ \text{W} = 2.698\ \text{Watts}$$

Summary

To summarise, the power usage of the device in Data Transmission mode is

Table 9 - Power Usage in Data Transmission mode

Circuit Section	Power Usage (mW)
CPU	2698.00
DAI	0.00
Interrupt Channel	127.00
SGA	0.00

5.3.4. Data Receipt

The final operating mode for the CPU is the mode used when the EHDL is receiving data over the air from a remote server. This is likely to have less impact than the transmission mode, but will be active for a more sustained time so could have a larger impact overall. Certain customer desirable functions are dependent on this power mode, particularly the device's ability to receive files from remote servers for firmware updates.

The power requirement for the data logger in receive mode is the same as that calculated in the previous section for the data transmission mode, but excluding the transmission power. Therefore, the calculated power is 1.748364 W.

Summary

To summarise, the power usage of the device in Data Receipt mode is

Table 10 - Power Usage in Data Receipt mode

Circuit Section	Power Usage (mW)
CPU	1748.00
DAI	0.00
Interrupt Channel	127.00
SGA	0.00

5.4. Timing Calculations

The power usage for each operating mode is only half of the story since for each of the operating modes a timing calculation is also required. At its most basic level this is the amount of 'On' time required per operation. For each of the energy modes detailed in section 0, a section below will detail the estimated time for each operation.

5.4.1. Sleep

Timing Calculation for sleep mode is relatively simple, the data logger is in sleep mode for all time that it is not performing another function. The calculated values for power usage in sleep mode were as follows (as calculated in section 5.3.1)

Table 11 - Power Usage in Sleep Mode

Circuit Section	Power Usage (mW)
CPU	0.033
DAI	0.00
Interrupt Channel	127.00
SGA	0.00

5.4.2. DAQ

The timing calculations for DAQ energy mode will make the following assumptions:

- That when a DAQ operation is made all channels on the logger are scanned

- That the interrupt channel is a separate entity not scanned during a 'scan'

With that in mind, the calculated values for power usage for DAQ mode were as follows (as calculated in section 5.3.2)

Table 12 - Power Usage in DAQ mode

Circuit Section	Power Usage (mW)
CPU	174.76
DAI	2770.00
Interrupt Channel	127.00
SGA	1087.31

The current status' of the DAI channels are taken as two separate readings, one from the I/O expander and one from the ADC. The estimated time for the devices to power on, stabilise and provide a reading is 100 ms.

The status' of the SGA channels are taken as four single readings of the separate devices on the board. The estimated time for the devices to power on, stabilise and provide a reading is 50 ms.

The best method for viewing power usage against time is on a graph. Technically the displayed information is a histogram for Energy usage.

For example, Figure 13 is a comparison of the Energy usage of a DAI card (left) compared with an SGA Card (Right)

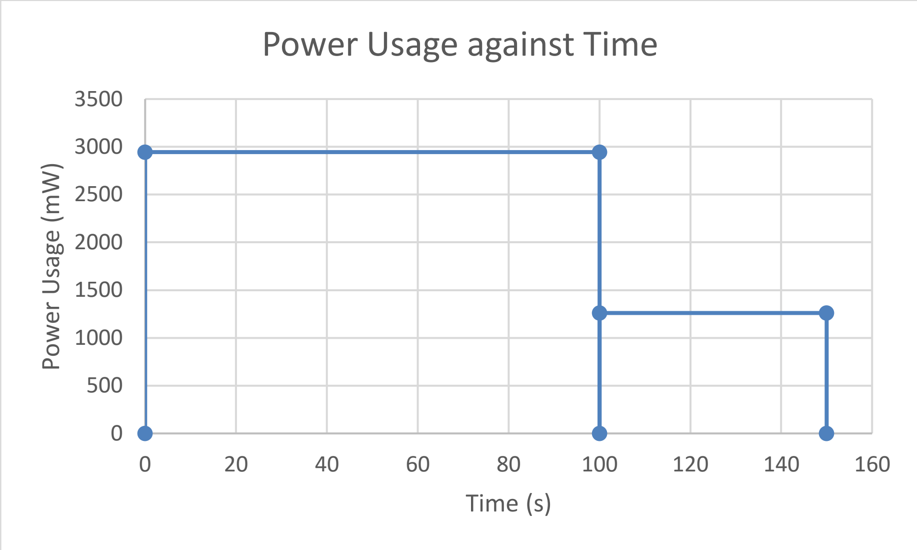


Figure 13 - Power Usage for DAI and SGA cards against Time

This is a useful method to view energy usage as the larger the area, the greater the amount of energy is consumed. The overall energy usage of the system will be increased by either drawing more power instantaneously or by leaving the component drawing the power on for longer.

The drawback of this method for presenting the data however is that, once the time-base begins to expand, the detail can often be lost. i.e. if this method was used to present a logger which was sleeping for 90% of its functional time, the detail of the remaining 10% would be eclipsed by the relatively large time base.

5.4.3. Data Transmission

The two operating modes making use of the 3G module are the most complex, and have the highest risk of inaccuracies. The following assumptions are made about the power usage when in data transmission mode:

- The average power usage of the 3G module is constant across a data transmission window. i.e. that the same amount of power is consumed during initialisation of the 3G module, during transmission and during shutdown. This should be caveated as inaccurate; however, this method is used to account for the peaks detailed in Figure 11 and Figure 12.
- That whenever the 3G Module is used a power up and initialisation time is required – this will be accounted for in the standard 'On' time.

The calculated values for power usage for data transmission mode were as follows (as calculated in section 5.3.3)

Table 13 - Power Usage in Data Transmission mode

Circuit Section	Power Usage (mW)
CPU	2698.00
DAI	0.00
Interrupt Channel	127.00
SGA	0.00

Taking into account the assumptions made above, the following must be accounted for. The time taken for the 3G module to power on, initialise and be ready to transmit is approximately 60 seconds across this time period: the logger will be assumed to be drawing the value calculated in section 5.3.3, additionally 15 seconds will be factored in for the transfer and receipt of acknowledge packet – Note: this is deliberately generous amount of time – this could be reduced once the transmission time has been determined.

The currently proposed time for data connection and transmission is 75 seconds. The impact of this decision is detailed in section 6.1.2.

5.4.4. Data Receipt

As with the data transmission mode, the data receipt mode has a high risk of inaccuracies. The following assumptions are made about the power usage when in data transmission mode.

- That all traffic during a data receipt is 'down' i.e. there is no transmission of acknowledge packets or any other 'upwards' traffic.
- That any data receipt will follow a data transmission and subsequently the power up and initialisation time will be contained within the associated data transmission. This also stands for the shutdown phase of the 3G module.

The calculated values for power usage for data receipt mode were as follows (as calculated in section 5.3.4)

Table 14 - Power Usage in Data Receipt mode

Circuit Section	Power Usage (mW)
CPU	1748.00
DAI	0.00
Interrupt Channel	127.00
SGA	0.00

The amount of time taken for a file download again is dependent on two major factors, the size of the file and the speed of the download. If the file is a firmware or configuration update to the data logger, common uses for file downloads in remote devices, the file sizes will be approximately 150 KB (1.2 Mbits) or 3KB (24 Kbits) respectively and the serial link between the processor and the 3G Module is running at 115.2 Kb/s then the download time for the files would be approximately 10.5 seconds or 0.2 seconds per file. The longer time of

10.5 seconds will be used as a preliminary value for the data receipt mode, this is due to it being the worst case and will give a better view of a realistic battery life

5.5. Configuration Parameters

This section covers the required configuration of the energy harvesting data logger as described in the introduction to section 5. To be able to calculate meaningful values for various hardware and configuration options a skeleton configuration file is required. The configuration file contains all of the information required for the hardware to determine the physical makeup of the device as well as the desired recording parameters.

5.5.1. Hardware Configuration

For each hardware type, the following options are available

- CPU Card (6210)
- DAI Card (6211)
- SGA Card (6212)

The physical makeup of the device is formed of a single CPU card which incorporates the 3G module as well as any combination of 6 I/O cards (DAI and SGA in any combination). This number of expansion cards is limited by the hardware design on the expansion connector, a compromise between the physical size of the header and the limit to functionality. Also, if many more cards were added the system might be unable to power itself in a sustainable way.

5.5.2. Recording Parameters

For each card there is set of parameters which are required for the power model to work

- Card Scan Frequency (1 per card)
- 3G Module Upload Frequency

Other parameters are required for the functioning of the device as a data logger i.e. functions to allow for de-bounce on digital inputs or averaging on analogue channels, for example. These settings are not relevant to the power model and have been omitted from this thesis as part of the scope in section 1.3.

5.5.3. Battery / Power Storage Parameters

One topic which has not been covered until now is the actual parameters of the battery. For power calculations to work correctly, the power model must know how large the battery of the system is.

Current industry data loggers are often powered from single use batteries such as standard C or D cells which are commonly available at short notice. The thought behind this approach is that if the batteries are dead on site then replacements can be obtained without much difficulty. The drawback of such generic batteries is that they are non-rechargeable. This approach works for short term deployments but if the installation exceeds the life of the battery, the logger could stop functioning, resulting in a technician having to attend the site to replace the battery. The basic functioning of the power model will cover the use of non-rechargeable batteries as a comparison.

A brief comparison on some common batteries is contained here to give some background to the EHDL battery section of the model.

The standard EHDL battery is a GP Batteries NTA2712 Rechargeable Lithium Ion battery [70] which operates at a voltage of 7.4 V with a capacity of 4400 mAh. The EHDL is designed to take two of these batteries in parallel giving a total capacity of 8800 mAh or 65.12 Wh.

A standard C Cell is a 1.5 V battery and, to give a usable working voltage, multiple batteries must be put in series. If 6 cells are combined, it gives a voltage of 9 V. Each battery has a capacity of 7750 mAh [71], so the total capacity of 69.75 Wh. However, the physical size of the batteries is significantly larger.

An alternative battery is the Tadiran Batteries SL-2770 Size C Lithium Thionyl Chloride. These batteries are regularly used for long term high performance applications: the SL-2770 [72] has a nominal voltage of 3.6 V and a Nominal capacity of 8.5 Ah. To get to the required voltage for the EHDL, this would need to be formed of 2 C size batteries in series. The total capacity for the SL-2770 is 61.2 Wh. These batteries are rechargeable.

To add a comparison to other commonly available batteries, the information follows for a standard PP3 Battery (9V Cell) [73]. The 9V Alkaline battery has a capacity of 625 mAh, giving an energy storage capacity of 5.625 Wh, a tenth of the batteries mentioned previously.

5.6. Summary

From the previous sections the following data has been extracted

Table 15 - Summary of Operating Mode Power and Time Data

Operating Mode	Power Usage (mW)	On-Time (ms)
Sleep	0.032685	N/A
DAQ (DAI)	2944.76	100
DAQ (SGA)	1261.76	50
DAQ (Interrupt)	127	All / 0
Data Transmission	2698.00	75000
Data Receipt	1748.00	10500

Additionally, it has been detailed which variables are required to generate a power model from the collected data.

6. Energy Harvesting Data Logger Power Model

Using the information collated within sections 4 and 5 it is possible to create a power model. This section focusses on the model itself exploring the methodology of the model, details of the code behind the model and details of the preliminary testing to confirm that the model functions as expected.

6.1. Model Methodology

The model is designed in Microsoft Excel, using integrated Visual Basic Macros. The reasoning for doing this was primarily to reduce the need for a Graphical User Interface, or GUI, in the initial development. Visual basic is a simple programming language with specific integrated functions for interfacing directly with Excel worksheets. For each part of the model, pseudo code will be provided with a commentary and the complete code will be provided in an appendix.

At this stage in the work, the power model will focus on the calculation of the power usage of the data logger in the first three energy modes – sleep, DAQ and data transmission. At this stage in the work the functions of Interrupts and Data Receipt have been omitted in this section. Later in this thesis, in sections 8.1 and 8.2 these two topics are covered in detail to explain how they affect the power model in differing ways.

The following section breaks the power model down into three main sections, section 6.1.1 looks at the hardware configuration of the logger. This covers the types of card fitted to the logger as well as looking at the scan frequency. Future expansions (outside of the scope of this thesis) would include the recording parameters of the logger such as filtering and the averaging of data.

The second section, 6.1.2, is the core of the power model, this section focusses on the power usage and timings of the logger, bringing together the data generated in 5.3 and 5.4 to calculate the energy use of the logger given the parameters provided.

The third section, 6.1.3, looks at the effect the values calculated in section 6.1.2 have on the expected life of the battery whilst performing this application.

The final, minor section, 6.1.4, displays how the power model looks in development, explaining the function of the various parts

6.1.1. Hardware Type

The first module of the code deals with the parameters provided in section 5.5.1. This is done by using a tab within the Power Model which details the current hardware assembly types. For each hardware assembly type the following information is required.

Table 16 - Hardware Type template

Assembly Part Number	CPU Card	I/O Slot 1	I/O Slot 2	I/O Slot 3	I/O Slot 4	I/O Slot 5	I/O Slot 6

For each existing card type there will be a reserved string defining the card type i.e. 6210.01 is the part number for the current CPU Card.

At present there is only one top level hardware configuration, the details of which are detailed in the format specified above in Table 17. The default Logger configuration is a CPU card with 2 DAI cards and 2 SGA cards. This is designed to give a wide range of channels whilst reserving capacity for further expansion. Realistically if a card is completely unused it should be removed so that the unused card draws no power from the system

Table 17 - 8006200.01 Hardware Configuration

8006200.01	6210.01	6211.01	6211.01	6212.01	6212.01	None	None

If no card is installed in a slot the string ‘None’ is used to inform the program.

As shown in Figure 14, the integration of the Hardware type sheet is done through a ‘Dropdown List’ and a ‘Button’. The dropdown list is used for the user to select which hardware type from the list of options available in the ‘Hardware Type’ tab (Figure 15). When the user clicks the ‘GO’ button on the main screen a macro is run which performs the following functions

- Reads the users selection from the dropdown

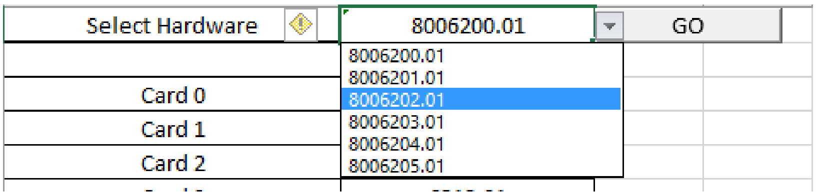


Figure 14 - User Selection of Hardware Configuration

- Compares the user’s selection with those available in the Hardware Type Tab

	A	B	C	D	E	F	G	H
1	8006200.01	6210.01	6211.01	6211.01	6212.01	6212.01	None	None
2	8006201.01	6210.01	6211.01	6211.01	None	None	None	None
3	8006202.01	6210.01	6212.01	6212.01	None	None	None	None
4	8006203.01	6210.01	6211.01	6211.01	6211.01	6211.01	6211.01	6211.01
5	8006204.01	6210.01	6212.01	6212.01	6212.01	6212.01	6212.01	6212.01
6	8006205.01	6210.01	6211.01	6211.01	6211.01	6212.01	6212.01	6212.01
7								

Figure 15 - Hardware Type Tab

- If a match is found, the main page is populated with the hardware configuration selected.

Select Hardware	8006200.01	GO
Card 0	6210.01	
Card 1	6211.01	
Card 2	6211.01	
Card 3	6212.01	
Card 4	6212.01	
Card 5	None	
Card 6	None	

Figure 16 - User Selection Populated in Energy Model

- `If no match is found, the program returns an error.`

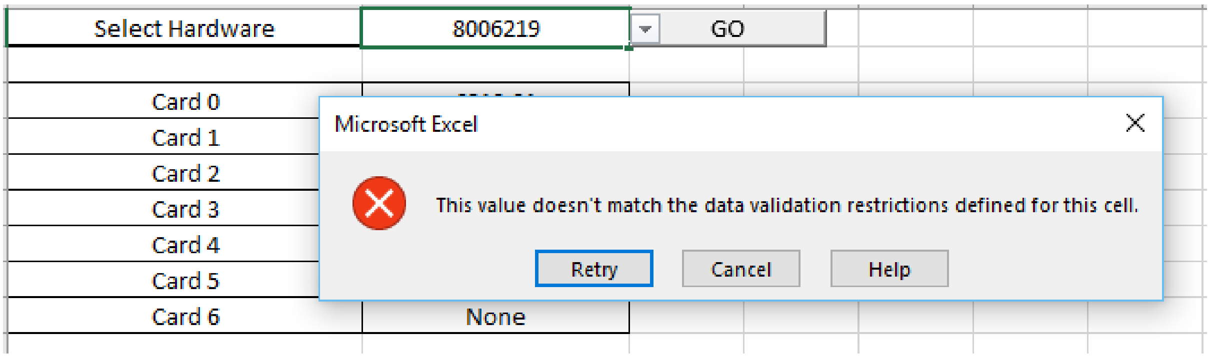


Figure 17 - Error Message

This function then displays the hardware configuration on the main page of the power model. It should assist a new user of the power model with only knowledge of the hardware to get started quickly. This section provides the basis for the power model by structuring the information required by the later parts of the model.

6.1.2. Time and Power Calculations

The second module of the code is the most significant part of the model. The entire model is based around a single tab per type of hardware. i.e. a tab for the CPU card, a tab for the DAI card and a tab for the SGA card. Within each tab is the power usage in each energy mode. So, for the CPU Card the following information is available

Table 18 - CPU Power Usage

Sleep Power	0.033
DAQ Power	174.76
Data T Power	2698
Data R Power	1748

This information is taken from Table 11, Table 12, Table 13 and Table 14 as defined in section 5. The same information is collated for both the DAI and SGA cards. From these data tables, the power model is capable of determining power usage for each card in each different mode of operation.

DAQ Mode On-Time

The power model uses the information detailed in section 5.5.2 to determine how many times each card is read per hour. This is done by dividing the number of seconds in an hour by the scan frequency.

$$3600/Card\ 1\ Scan\ Frequency = number\ of\ scans\ per\ hour$$

The variables for each card are stored in an array for ease of data handling. If the scan frequency is 0, the program returns an error (which is handled allowing the user to continue but alerting them to the fact that there has been a '0' entered)

To calculate the On Time of the system a function is used. The following process is followed

- `For Each I/O Card:`
- `Read Hardware Configuration and Store in Array`
- `For the current I/O Card:`
    - o `Case DAI`

        - Single On Time x Card Scan Frequency
    - Case SGA
        - Single On Time x Card Scan Frequency
    - Case "None"
        - Return 0
    - Case N/A
        - Hardware Selection Invalid Error
- Return the On Time for current I/O Card

This process is repeated for each I/O card location with the return value for each card added to existing value. This gives a sum for the total on-time of I/O cards.

3G Module On-Time (Data Transmission)

The data transmission frequency is also calculated in a similar way to the I/O. However the value for data transmission frequency is permitted to be either a fraction or a whole number.

The equation for data transmission is slightly different because the variable is requested in hours not seconds. Therefore, the equation is

$$\frac{1}{Data\ Transmission\ Frequency} = number\ of\ data\ transmissions\ per\ hour$$

The result is stored. If the data transmission frequency is 0, then it is assumed that there is no requirement for data offload – the on-time is stored as 0.

To calculate the 3G on-time of the system caused by the 3G Module the following equation is used

- `Multiply number of transmissions per hour x on-time per transmission`

The obvious flaw in this calculation method is that the energy usage by the 3G module is averaged across the total 3G transmission cycle window – this will cause discrepancies in the exact time within a cycle that the power will 'run-out'. The system battery life will only be as accurate as the number of complete 3G module cycles.

System On-Time

The total On-time of the system is a simple addition

$$Total\ OnTime\ (per\ hour) = DAQ\ OnTime\ (per\ hour) + 3G\ Module\ OnTime (per\ hour)$$

And inversely, the total time asleep per hour is calculated as

$$System\ OffTime = 1\ Hour - Total\ OnTime\ (per\ hour)$$

Sleep Energy Calculations

The equation for sleep energy is Equation 10 (below)

$E = Pxt$ Electrical Energy (J) = Power (W) x Time (s)

The total energy used is the power usage in sleep mode multiplied by the sleep time. It is worth noting that Joules is not a suitable unit of measurement for this work, see section 4.2, in this case, Amp hours or Ah, will be used.

DAQ Energy Calculations

To calculate the energy usage (per hour) a function is used.

- `For Each I/O Card:`
- `Read Hardware Configuration and Store in Array`

- For the current I/O Card:
  - Case DAI
    - Calculate Electrical Energy consumed in one Hour (Wh)
    - $Energy = \frac{Power\ Usage\ (mW)}{1000} \times \frac{On\ Time\ (ms)}{3.6 \times 10^6}$
  - Case SGA
    - Calculate Electrical Energy consumed in one Hour (Wh)
    - $Energy = \frac{Power\ Usage\ (mW)}{1000} \times \frac{On\ Time\ (ms)}{3.6 \times 10^6}$
  - Case "None"
    - Return 0
  - Case N/A
    - Hardware Selection Invalid Error
- Return the Energy Usage for current I/O Card

This process is repeated for each I/O card location with the return value for each card added to existing value.

Additional to the energy used by the cards themselves, the Energy used by the CPU must be taken into account. This is done with the following equation

$$Energy\ Usage\ (CPU) = \frac{Power\ Usage\ (mW)}{1000} \times \frac{Total\ DAQ\ Time\ (s)}{3600}$$

Adding the CPU energy usage to the card energy usage gives a sum for the total energy consumption of I/O cards within one hour.

3G Module Energy Calculations

To calculate the energy usage of the 3G module the same method is used, the On-Time per hour of the 3G Module has been previously calculated, therefore the energy consumption of the 3G module is:

$$Energy\ Usage\ (Wh) = \frac{Power\ Usage\ (mW)}{1000} \times \frac{OnTime\ (s)}{3600}$$

Total Energy Calculations

To calculate the total energy used by the module in one hour the energy calculated for the various modes is summed.

$$\begin{aligned} &Total\ Energy\ Use\ (1\ hour) \\ &\quad = DAQ\ Energy\ Usage\ (1\ hour) + 3G\ Module\ Energy\ Usage (1\ hour) \\ &\quad + Sleep\ Energy\ Usage\ (1\ hour) \end{aligned}$$

This final value is then fed back into the Excel spreadsheet.

6.1.3. Battery Life Calculations

To Predict the battery life of the system, standard Excel formulas are used. The user is presented with a list of popular battery types/sizes: these are not necessarily suitable to the data logger, but are given to provide additional information, section 5.5.3 provides greater detail on batteries.

To compare the battery life using the power model, the voltage (V) and capacity (mAh) are required. This information, coupled with the Estimated Energy Usage (Wh), can be used to determine the battery voltage. Equation 13 and Equation 14 are calculated automatically and the results presented to the user.

Equation 13 - Estimated Current

$$Estimated\ Current\ Draw\ (Ah) = \frac{Estimated\ Energy\ Usage\ (Wh)}{Voltage\ (V)}$$

The calculated energy usage is calculated in Watt-hours (a meaningful unit of kWh which would be the standard) however batteries are often quoted in ampere-hours or milli-ampere-hours. To compare the two units, the voltage of the battery is required – using Joule's Law (Equation 9) it is possible to convert between Power and Current.

Equation 14 - Estimated Battery Life

$$Estimated\ Battery\ Life\ (Hours) = \frac{Battery\ Capacity\ (mAh)}{1000 \times Estimated\ Current\ Draw\ (Ah)}$$

$$Estimated\ Battery\ Life\ (Days) = \frac{Estimated\ Battery\ Life\ (Hours)}{24}$$

$$Estimated\ Battery\ Life\ (Years) = \frac{Estimated\ Battery\ Life\ (Days)}{365.25}$$

To make the estimated energy usage meaningful it is necessary to convert the energy usage into a predicted lifetime. Once the Energy requirement has been converted into Ampere-hours, the capacity of the battery can be divided by the hourly usage to give a time (in hours). Subsequently this time in hours can be transformed into various other time periods: the two chosen formats for display are days and years.

6.1.4. Graphical User Interface

This final section displays the visual aspects of the power model which although basic, are functional to perform the requirements. The power model shown in Figure 18 only displays the parts of the configuration file relevant to the power model: the rest is outside of the scope of this thesis.

The option to select the hardware is followed by the construction of the current hardware. As shown in Figure 16. Below the hardware configuration are 7 cells for the user to enter the card scan frequencies and data transmission frequency as shown in Figure 18.

Select Hardware	8006200.01	GO
Card 0	6210.01	
Card 1	6211.01	
Card 2	6211.01	
Card 3	6212.01	
Card 4	6212.01	
Card 5	None	
Card 6	None	
Card 1 Scan Frequency	60	s
Card 2 Scan Frequency	60	s
Card 3 Scan Frequency	60	s
Card 4 Scan Frequency	60	s
Card 5 Scan Frequency	0	s
Card 6 Scan Frequency	0	s
3G Module Update Frequency	2	h
Estimated Energy Usage (Wh)	0.029021503	Calculate Power Usage
Battery Voltage	8.4	
Battery Capcity (mAh)	4400	
Estimated Current Draw (Ah)	0.003454941	
Lifetime (Hours)	1273.538455	
Lifetime (Days)	53.06410231	
Lifetime (Years)	0.145281594	

Figure 18 - Data Logger Power Model GUI

Finally, at the bottom of the GUI is the output or results, displaying the estimated energy usage, battery setup and projected battery lifetime.

This section has a 'Calculate Power Usage' button, which runs the calculations behind the scenes and populates the result into the 'Estimated Energy Usage cell. This value is presented in Watt Hours for ease of calculation.

The remaining cells are populated as per Equation 14, this gives the predicted battery life in Hours, Days and Years.

These four sections summarise the logger, detailing its design and the methodology taken to implement it. The source code for the varying sections is contained within Appendix B - Power Model Source Code

6.2. Preliminary Testing

This section focusses on the testing of the power model. The aim is to determine if the logger is functioning as specified or as it should be expected to. By isolating parts of the model (using the in-built functionality to enable or disable cards) it is possible to confirm through manual calculation what the estimated energy usage should be.

To test the power model a series of tests have been carried out. To enable the testing, various message boxes can be enabled or disabled in the code. These message boxes are a simple visible method of displaying the variables contained within the model without having to 'break into' the code with debugging software.

The tests are broken into three types of testing, firstly timing tests confirming that the model is calculating the on and off times correctly. The second type of tests are energy tests; these confirm that the data used to drive the model is being called correctly and manipulated to give accurate readings. The final type of tests are overall system tests where multiple functions are combined together to confirm the implementation has been carried out successfully, this validated by manually calculating the power usage and comparing the energy usage as calculated by the model

6.2.1. Timing Tests

The first set of testing is to confirm that, at its simplest, the model reports the correct values for pre-determined setups.

Test 1.1

If all values for scan frequencies are set to zero, the power model should default to sleep mode permanently. The power model should estimate an energy usage of the CPU in sleep mode (0.033 mWh).

Result: Pass – After 7 Warnings about dividing by zero the output from the logger appeared as follows

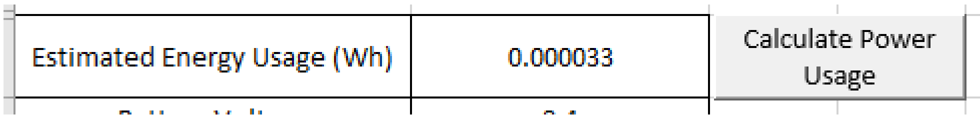
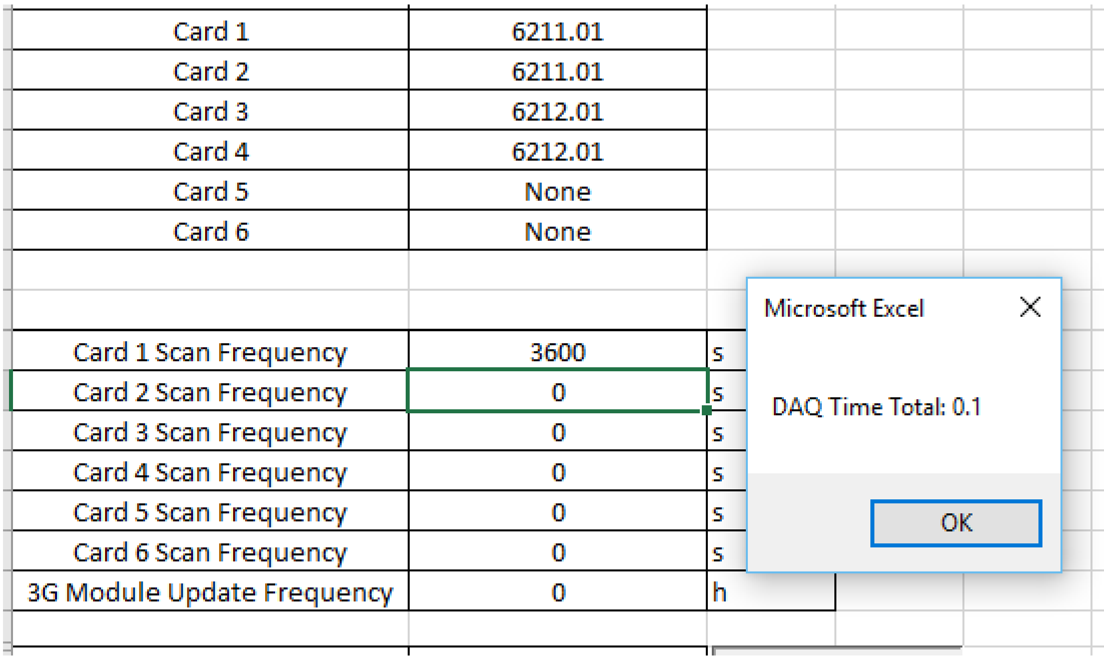


Figure 19 - Results - Test 1.1

Test 1.2

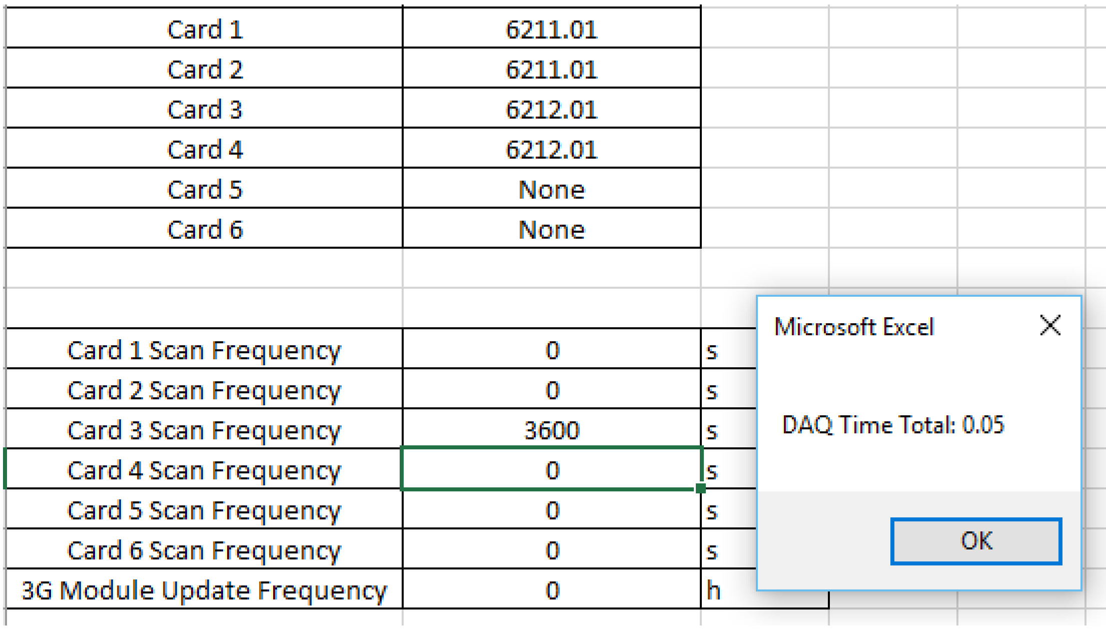
Using the same hardware setup if a single card is set to a scan frequency of 3600 a single sweep of the channels on that card will take place every hour. For a DAI Card the time should be 100 ms and for a SGA card the time should be 50 ms.

Result: Pass – Both cards displayed the correct time for the DAQ phase.



Card 1	6211.01	
Card 2	6211.01	
Card 3	6212.01	
Card 4	6212.01	
Card 5	None	
Card 6	None	
Card 1 Scan Frequency	3600	s
Card 2 Scan Frequency	0	s
Card 3 Scan Frequency	0	s
Card 4 Scan Frequency	0	s
Card 5 Scan Frequency	0	s
Card 6 Scan Frequency	0	s
3G Module Update Frequency	0	h

Figure 20 – Results – Test 1.2a



Card 1	6211.01	
Card 2	6211.01	
Card 3	6212.01	
Card 4	6212.01	
Card 5	None	
Card 6	None	
Card 1 Scan Frequency	0	s
Card 2 Scan Frequency	0	s
Card 3 Scan Frequency	3600	s
Card 4 Scan Frequency	0	s
Card 5 Scan Frequency	0	s
Card 6 Scan Frequency	0	s
3G Module Update Frequency	0	h

Figure 21 - Results - Test 1.2b

Test 1.3

Using the same hardware setup, if only the 3G Module Update frequency has a value of 1 (i.e. 1 update per hour) it should show the on time for a single update (75 Seconds)

Result: Pass – see Figure 22.

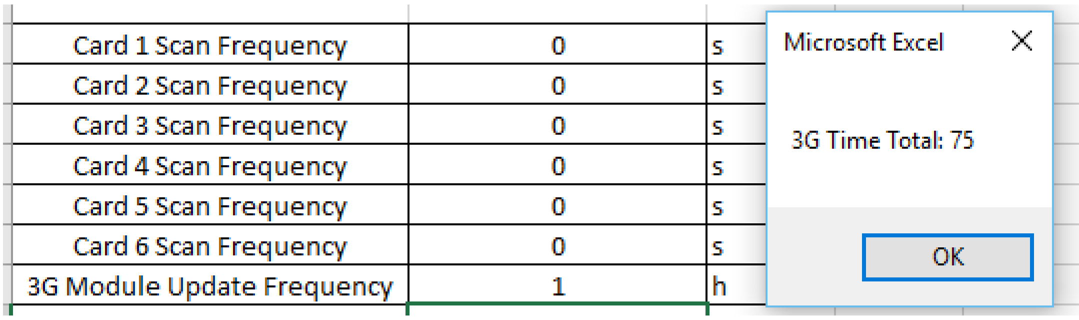


Figure 22 - Results - Test 1.3

Tests 1.4 – 1.9

Beyond these simple tests, by varying the parameters it is possible to prove the mathematics within the timing section of the model is correct. The following tests were performed with debug messages for the various timing stages turned on to enable the timings to be extracted.

Test	Description	Result
1.4	Repeat of Test 1.2 with scan frequency of 1800s	Pass
1.5	Repeat of Test 1.3 with frequency of 0.5 hours	Pass
1.6	Repeat of Test 1.2 with scan frequency of 900s	Pass
1.7	Repeat of Test 1.3 with frequency of 2 hours	Pass
1.8	Set Card 1 and Card 3 scan frequencies to 3600. Time for DAQ should be 0.15s	Pass
1.9	Set Card 1, 2, 3 & 4 scan frequencies to 3600. Time for DAQ should be 0.3s	Pass

6.2.2. Energy Tests

Testing the energy calculations is slightly more complex, but can still be worked out manually to compare with the answer.

Test 2.1

If all values for scan frequencies are set to zero, as proved by test 1.1, the estimated energy usage is given by the sleep power usage of the CPU card. By turning on the message boxes in the code it should be possible to confirm that the logger is permanently in sleep mode and the source of the energy is the CPU in sleep.

Result: Pass – the DAQ and GSM Energy Are Zero.

Test 2.2

As per test 2.1, it is possible to determine the DAQ energy when the card scan frequency is set to maximum i.e. if the sleep time is set to zero.

To achieve this the scan frequency of a DAI card is set to 0.1 seconds. This means that the 'On Time' of the system is 3600s. With no other parameters set, the logger is not performing any other function.

The estimated power draw for a DAI card should be 2944.76 mWh or 2.94476 Wh.

Result: Pass – The estimated energy usage is 2.94476 Wh. – see comment below

Card 1 Scan Frequency	0.1	s
Card 2 Scan Frequency	0	s
Card 3 Scan Frequency	0	s
Card 4 Scan Frequency	0	s
Card 5 Scan Frequency	0	s
Card 6 Scan Frequency	0	s
3G Module Update Frequency	0	h
Estimated Energy Usage (Wh)	2.94476	Calculate Power Usage

Figure 23 - Results - Test 2.2

What these tests are directly looking for in particular are scaling errors. In this instance, the formula for the card on time was incorrect – i.e. rather than working in seconds, the energy usage of the individual cards was being calculated on milliseconds – this was reducing the energy usage of the card within the model by a factor of 1000. With this correction made, the test result was a pass. Although some of the above testing may seem superfluous, results like the initial results to 2.2 can save a lot of time in the long run.

Test 2.3

This is a repeat of test 2.2 but focussing on the SGA card. The calculated output should be 1261.76 mWh or 1.262 Wh.

Result – Pass

Card 1 Scan Frequency	0	s
Card 2 Scan Frequency	0	s
Card 3 Scan Frequency	0.05	s
Card 4 Scan Frequency	0	s
Card 5 Scan Frequency	0	s
Card 6 Scan Frequency	0	s
3G Module Update Frequency	0	h
Estimated Energy Usage (Wh)	1.26176	Calculate Power Usage

Figure 24 - Results - Test 2.3

Note: The same fix was required as in Test 2.2.

Test 2.4

This test focusses on the 3G Module. The aim is to perform the same effect as that which was used in tests 2.2 and 2.3. To achieve this with the 3G module the 3G card must be run 48 times each hour (so that the total on time of the 3G card is 3600 Seconds (75 x 48)).

To do this the cell containing the 3G module transmission frequency must read 1/48 or $0.0208\dot{3}$. To accurately achieve this the formula of 1/48 was inserted into the cell.

With the 3G module on permanently the energy usage should be 2698 mWh (2.698 Wh)

Result - Pass

Card 1 Scan Frequency	0	s	
Card 2 Scan Frequency	0	s	
Card 3 Scan Frequency	0	s	
Card 4 Scan Frequency	0	s	
Card 5 Scan Frequency	0	s	
Card 6 Scan Frequency	0	s	
3G Module Update Frequency	0.020833333	h	
Estimated Energy Usage (Wh)	2.698	Calculate Power Usage	

Figure 25 - Results - Test 2.4

Test 2.5

The final test is to confirm that the three separate parts of the model are summed correctly. To model this the scan frequencies are configured to give a third of the operating time to the following operating modes – Sleep, DAQ and Data Transmission.

This will give 1200 seconds (20 minutes per hour) of operation for each mode, the energy use per operation should be one third of the total energy when the mode is run 100% of the time.

$$Total\ Calculated\ Energy = \frac{2.94476\ \text{Wh}}{3} + \frac{2.698\ \text{Wh}}{3} + \frac{0.000033\ \text{Wh}}{3}$$

This gives 1.880931 Wh.

Card 1 Scan Frequency	0.3	s	
Card 2 Scan Frequency	0	s	
Card 3 Scan Frequency	0	s	
Card 4 Scan Frequency	0	s	
Card 5 Scan Frequency	0	s	
Card 6 Scan Frequency	0	s	
3G Module Update Frequency	0.0625	h	
Estimated Energy Usage (Wh)	1.880931	Calculate Power Usage	

Figure 26 - Results - Test 2.5a

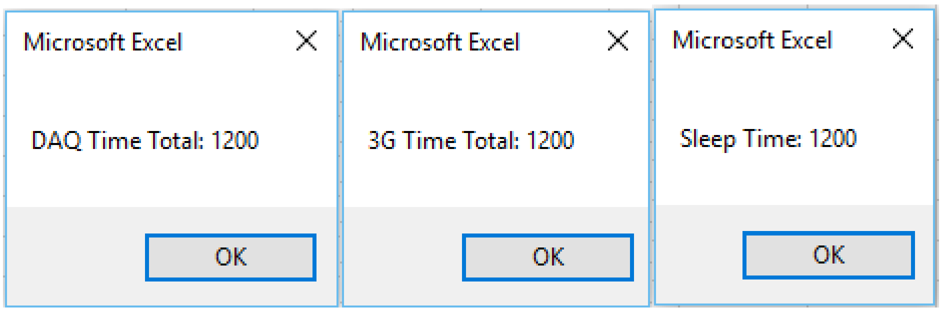


Figure 27 - Results - Test 2.5b

6.2.3. Overall System Tests

To confirm the various sections of the model are working in harmony, it is necessary to perform a more complex test utilising each of the specific sections of the model.

To perform this test, the standard hardware configuration will be used again. For each of the configurable fields the variables have been given arbitrary values which may or may not be practical in a deployed application, but allow for the testing of the code.

Card 1 Scan Frequency	10	s
Card 2 Scan Frequency	20	s
Card 3 Scan Frequency	30	s
Card 4 Scan Frequency	40	s
Card 5 Scan Frequency	0	s
Card 6 Scan Frequency	0	s
3G Module Update Frequency	2	h

Figure 28 - Test Variables for Test 3.1

Firstly, the total on-time for the DAQ must be calculated, From the variables it is possible to calculate the following

$$Total\ DAQ\ time = \left(\frac{3600}{10}\times 0.1s\right) + \left(\frac{3600}{20}\times 0.1s\right) + \left(\frac{3600}{30}\times 0.05s\right) + \left(\frac{3600}{40}\times 0.05s\right)$$
$$= 64.5\ seconds$$

Secondly the 3G on-time must be calculated:

$$3G\ On\ Time = \frac{1}{2}\times 75\ seconds = 37.5\ seconds$$

Therefore, the total on-time is 102 seconds.

The Sleep time is subsequently:

$$3600 - 102 = 3498\ seconds$$

The Energy consumed by the logger in sleep mode is:

$$\frac{3498}{3600}\times 0.000033 = 0.000032\ Wh$$

Energy Usage by Card 1:

$$2.770\times\frac{\frac{3600}{10}\times 0.1s}{3600} = 0.0277\ Wh$$

Energy Usage by Card 2:

$$2.770\times\frac{\frac{3600}{20}\times 0.1s}{3600} = 0.01385\ Wh$$

Energy Usage by Card 3:

$$1.087 \times \frac{\frac{3600}{30} \times 0.05s}{3600} = 0.001812\ Wh$$

Energy Usage by Card 4:

$$1.087 \times \frac{\frac{3600}{40} \times 0.05s}{3600} = 0.00135875\ Wh$$

Energy Usage by CPU during DAQ:

$$0.17476 \times \frac{64.5}{3600} = 0.00313\ Wh$$

Total Energy Usage in DAQ mode:

$$0.0227 + 0.01385 + 0.001812 + 0.00136 + 0.00313 = 0.04785\ Wh$$

Energy Usage by 3G Module:

$$2.698 \times \frac{37.5}{3600} = 0.0281\ Wh$$

Total Energy Usage:

$$0.000032 + 0.04785 + 0.0281 = 0.07598\ Wh$$

Rounding errors are expected because Excel holds values to a higher resolution.

Select Hardware	8006200.01	GO
Card 0	6210.01	
Card 1	6211.01	
Card 2	6211.01	
Card 3	6212.01	
Card 4	6212.01	
Card 5	None	
Card 6	None	
Card 1 Scan Frequency	10	s
Card 2 Scan Frequency	20	s
Card 3 Scan Frequency	30	s
Card 4 Scan Frequency	40	s
Card 5 Scan Frequency	0	s
Card 6 Scan Frequency	0	s
3G Module Update Frequency	2	h
Estimated Energy Usage (Wh)	0.075987765	Calculate Power Usage

Figure 29 - Results - Test 3.1

Result – Pass – the calculated answer is accurate to 5 decimal places, beyond which rounding accuracies take hold.

6.2.4. Test Summary

The testing plan has proved its worth by discovering the error in calculation in Test 2.2. By validating the functionality of the logger in a controlled environment it allows a level of confidence when applying the power model into practical applications. From the results of these tests it is acceptable to release the power model for investigative work.

7. Model Utilisation and Accuracy Validation

This next section of the thesis focusses on the implementation testing of the Logger, this includes a range of sections which are all related to the implementation or work required to implement a feedback loop from the power usage data which can be collected from the loggers and use the data to improve the accuracy of future predictions.

The first section, 7.1, focusses on using the model to learn more about the application of the hardware in a working application. The second section, 7.2, details how the firmware within the logger would need to be modified to collect data on the energy usage of the logger. The third section, 7.3, details how the data recorded in section 7.2 could be used to calculate the energy usage of the logger and be fed back into the logger to give more accurate predictions going forwards. The final section, 7.4, investigates the implementation of the power model into a business environment looking at where cost savings and other benefits can be drawn from the usage of the power model

7.1. Hardware Investigation

The power model allows for an investigation of the design of the hardware, in section 5.1 it was predicted that the most significant power usage would be the 3G module. On the face of it this prediction is correct as the calculations for the various energy modes show that the largest instantaneous power draw is the data transmission, however this may not be the complete picture. This section of the thesis focusses on the balance between the functions of the logger.

7.1.1. DAQ vs 3G Transmission

This section looks at the balance between the energy usage of the logger whilst in data transmission mode with the energy usage whilst in DAQ mode. The distinguishing factors here are the frequencies of operation: to best compare the energy usage, a series of tests will be carried out with various scan and update frequencies to determine the effect of the various modes.

The basic premise for this is to investigate the energy usage calculated by the model for the two modes with the other mode disabled as well as investigating the crossover between the two.

Tests 1 to 25 detail the energy usage of the logger when only performing a data acquisition, with the data transmission mode disabled. Tests 26 to 33 detail the energy usage of the logger when only performing a data transmission, with data acquisition mode disabled – it is noted that this mode performs no useful function because with no data acquisition there would be no data to send. The final set of tests, 34 to 49 detail a hybrid of the two modes showing the crossover between the modes.

Table 19 - Test Cases for DAQ and Data Transmission comparison

Case #	Card Scan Frequency (seconds)	3G Card Scan Frequency (hours)	Est Energy Usage (mAh)	On Time (Per Hour)	Comments
DAQ Only					
1	3600	0	0.0002316	0.3	1 Hour Card Scan – no Data upload
2	1800	0	0.0004303	0.6	Card Scan Only – 30 Minutes
3	1200	0	0.0006289	0.9	Card Scan Only – 20 Minutes
4	600	0	0.0012249	1.8	Card Scan Only - 10 Minutes
5	540	0	0.0013573	2	Card Scan Only - 9 Minutes

Case #	Card Scan Frequency (seconds)	3G Card Scan Frequency (hours)	Est Energy Usage (mAh)	On Time (Per Hour)	Comments
6	480	0	0.0015228	2.25	Card Scan Only - 8 Minutes
7	420	0	0.0017357	2.57	Card Scan Only - 7 Minutes
8	360	0	0.0020194	3	Card Scan Only - 6 Minutes
9	300	0	0.0024167	3.6	Card Scan Only - 5 Minutes
10	240	0	0.0030127	4.5	Card Scan Only - 4 Minutes
11	180	0	0.0040059	6	Card Scan Only - 3 Minutes
12	120	0	0.0059923	9	Card Scan Only - 2 Minutes
13	60	0	0.0119516	18	Card Scan Only - 1 Minutes
14	30	0	0.02387	36	Card Scan Only - 30 Seconds
15	20	0	0.0357889	54	Card Scan Only - 20 Seconds
16	10	0	0.071545	108	Card Scan Only - 10 Seconds
17	5	0	0.143057	216	Card Scan Only - 5 Seconds
18	2	0	0.357592	540	Card Scan Only - 2 Seconds
19	1	0	0.715151	1080	Card Scan Only - 1 Second
20	0.5	0	1.430269	2160	Card Scan Only - 0.5 Seconds
21	0.2	0	3.575623	5400	Card Scan Only - 0.2 Seconds
22	0.1	0	7.151214	10800	Card Scan Only - 0.1 Seconds
23	0.05	0	14.302395	21600	Card Scan Only - 0.05 Seconds
24	0.02	0	35.755938	54000	Card Scan Only - 0.02 Seconds
25	0.01	0	71.511843	108000	Card Scan Only - 0.01 Seconds
3G Only					
26	0	0.1	0.5621095	750	3G Only - 6 Minutes
27	0	0.2	0.2810712	375	3G Only - 12 Minutes
28	0	0.3	0.1873918	250	3G Only - 18 Minutes
29	0	0.5	0.1124483	150	3G Only - 30 Minutes
30	0	1	0.0562406	75	3G Only - 1 Hour
31	0	2	0.0281368	37.5	3G Only - 2 Hours
32	0	3	0.0187689	25	3G Only - 3 Hours
33	0	6	0.009401	12.5	3G Only - 6 Hours
Hybrid Operation					
34	60	6	0.02132	30.5	1 Minute Scan Rate, 6 Hour Update
35	120	6	0.01536	21.5	2 Minutes
36	180	6	0.01337	18.5	3 Minutes

Case #	Card Scan Frequency (seconds)	3G Card Scan Frequency (hours)	Est Energy Usage (mAh)	On Time (Per Hour)	Comments
37	240	6	0.01238	17	4 Minutes
38	300	6	0.01179	16.1	5 Minutes
39	360	6	0.01139	15.5	6 Minutes
40	420	6	0.0111	15.07	7 Minutes
41	480	6	0.01089	14.75	8 Minutes
42	540	6	0.01073	14.5	9 Minutes
43	600	6	0.01059	14.3	10 Minutes
44	1200	6	0.00999687	13.4	20 Minutes
45	1800	6	0.009798	13.1	30 Minutes
46	3600	6	0.009599585	12.8	1 Hour Scan Rate, 6 Hour Update
47	3600	3	0.0189675	25.3	1 Hour Scan Rate, 3 Hour Update
48	3600	2	0.0283355	37.8	1 Hour Scan Rate, 2 Hour Update
49	3600	1	0.0564393	75.3	1 Hour Scan Rate, 1 Hour Update

This table demonstrates the difference between what have been identified to be the most significant power use cases. The three stages of the table depict the different power uses firstly in isolation followed by a comparison of the effect of mixing them.

The most interesting learning point from the values from the modes in isolation is to discover the point at which the two modes become equal in their power usage, the graph below is generated from the top two thirds of Table 19

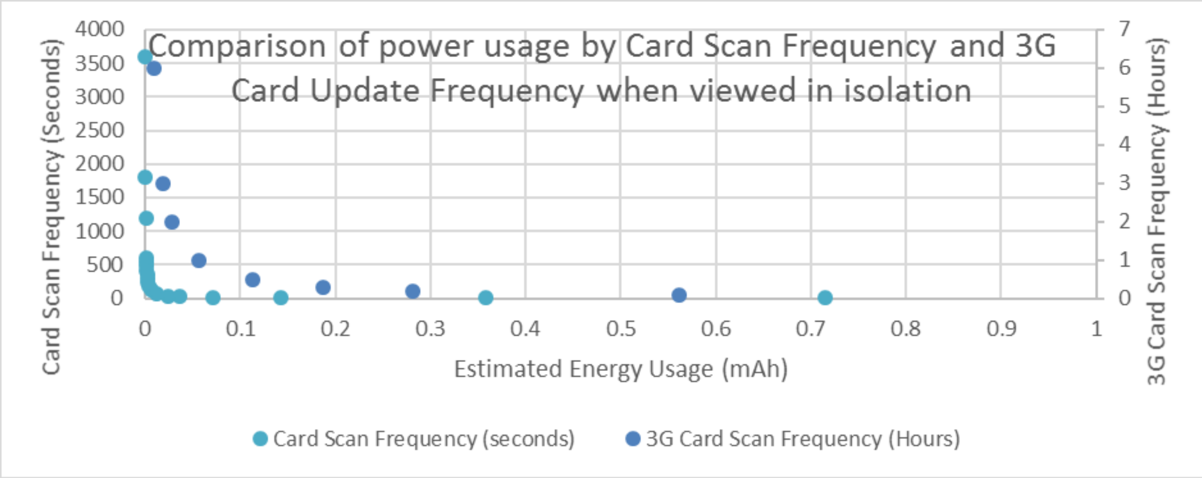


Figure 30- Comparison of Power Usage by Card Scan Frequency against Power Usage by 3G Card

The graph in Figure 30 shows is that with the values entered into the table that the 'cross-over' point is still not reached – i.e. there is still not a value for which the update frequency in hours draws as little power as the lowest scan rate – to go beyond the value of 6 hours moves away from the purpose of the logger to provide 'real-time' data as some of the data will be over 6 hours old even at this threshold

When used in the hybrid mode Table 19 shows the impact which the 3G update frequency can have on the overall power usage, but shows the overwhelming priority which should be given to the scan rate of the logger. The following section looks into applying real world values into the model to get a realistic view of the loggers performance.

7.1.2. Real World Values

A sensible setup for the logger is likely to be a scan rate of 1 second for the DAQ and a Data Transmission every 2 hours. With these values entered into the model the following results are given:

System On Time = 1117.5 Seconds per hour

Energy Usage by DAQ mode = 0.715128 Wh

Energy Usage by Data Transmission Mode = 0.0281 Wh

Total Energy Usage = 0.7433 Wh

This contradicts the predictions made previously that the 3G module would draw the most energy. The prediction was correct in that the 3G module draws the largest amount of power, but when the data acquisition mode is running every second, the amount of energy used dwarfs the 3G module. In this configuration the 3G module is only responsible for 3.8 % of the total energy usage.

Another product on the market [74] performs a data scan every two minutes, with the data reported back each hour, for this configuration the model gives the following results:

System On Time = 84 Seconds per hour

Energy Usage by DAQ mode = 0.00596 Wh

Energy Usage by Data Transmission Mode = 0.0562 Wh

Total Energy Usage = 0.0621 Wh

In this configuration (Scan 120s, Transmission 1hr) the 3G module (data transmission mode) is responsible for 90 % of the energy usage.

The two examples above can be used to prove one significant point. By contrasting them, it can be seen that the critical part of the configuration is not the frequency at which data is uploaded to the server, but in fact the frequency at which the data acquisition occurs, although the DAQ mode uses less power for less time (per occurrence) the fact that it can occur hundreds of times between data uploads can result in it having a vastly increased 'on-time' by comparison.

Delving more deeply, Table 20 gives a direct comparison between the two previous examples measured through various indicators. This summary makes viewing and comparing the data significantly more simple.

Table 20 - Comparison of Real World Examples

Example	Example 1	Example 2
DAQ Frequency	1 Second	120 Seconds
Data Transmission Frequency	2 Hours	1 Hour
System 'On-Time' (Per Hour)	1117.5 seconds	84 Seconds
Energy Usage (DAQ mode)	0.715128 Wh	0.00596 Wh
Energy Usage (Data Transmission)	0.0281 Wh	0.0562 Wh
Total Energy Usage	0.7433 Wh	0.0621 Wh

Using the data comparisons drawn from Table 19, the impact of the scan frequency is starkly displayed in this example, at a scan rate of 1 second, the DAQ function runs 3600 times per hour, even with its on-time of 0.3s this equates to an 'On-time' per hour of 1080 seconds or 18 minutes, over a quarter of an hour. To achieve this level of 'on-time' (power usage excluded) the data transmission would need to operate 14 times, or every 4 minutes, the increased power usage of the logger reduces this value down, but this serves to demonstrate the how the frequency of Data Acquisition could be misconceived as negligible, but have a significant impact.

As shown in Figure 13, the energy usage is defined not only by the peak power usage but also the amount of time that power is drawn. The power requirement of the data logger when in DAQ mode is significantly higher than desired: to achieve a higher scan rate a more power efficient design would be required and the implications of this are discussed in section 8.1.

7.2. Measuring Energy Usage

The next section focusses on how to modify the firmware of the EHDL to enable the validation of the power model by directly recording the energy usage of the logger. The reasoning behind this is to enable to logger to self-monitor whilst deployed in a remote location. A primary purpose for this is to monitor the amount of energy harvested. For bench testing during development and prior to deployment the energy usage of the logger can be externally monitored using either self-powered or mains powered test equipment, this however is not an option for remote or outside environments. This section looks at how to amend the existing firmware as detailed in section 5.1.4 to record this additional data.

7.2.1. Modifying firmware to measure energy usage

To turn the data logger into a 'smart logger' it is necessary to make changes to the existing firmware. These changes require the monitoring of additional channels which are fitted to the data logger and additional mathematics integrated to the logger and the GUI. The purpose of such changes allows the logger to monitor its own energy usage over a period of operation and when next connected to the PC software feedback energy usage data. This data can be assimilated into future predictions of deployment feasibility.

The Data logger is capable of monitoring the following information regarding its power supply and current usage as detailed in Table 21, the table details the type of channel and gives a brief description of the channel.

Table 21 - Additional Energy Usage Monitoring Channels

Signal	Direction	Active	Description
DC/DC Mode	Output	Both	This signal controls the mode in which the main DC/DC convertor between the battery and the logger is operating
I_Load	Input	ADC	This signal is an analogue input to the data logger showing the current load on the main DC/DC
EH_Fault	Input	H	This signal indicates a fault on the Solar and wind harvesting device
EH_Charge	Input	H	This signal indicates that the battery is being charged by either solar or wind power
Battery Level Detection	Output	H	This signal enables the Batt_LVL channel (see below)
Batt_Lvl	Input	ADC	This signal is an analogue input to the data logger giving a reading of the current battery level, when not being directly read, this channel is disabled to prevent un-necessary power loss. Battery Level Detect is used to enable or disable this channel

From these channels it is possible to draw a picture of the energy usage of the logger. It is not without irony that the monitoring of such channels by the data logger increases the current drawn by the system, however this increase can be accounted for by subtracting the power directly attributed to the monitoring or by considering it negligible.

To measure out the energy usage of the system the following variables are required:

- Current Battery Voltage
- Instantaneous current draw from main DC/DC

Single readings are insufficient to give enough information because energy usage is time based, a historic log of both variables is required to perform the desired calculations.

7.2.2. Recording Meaningful Data

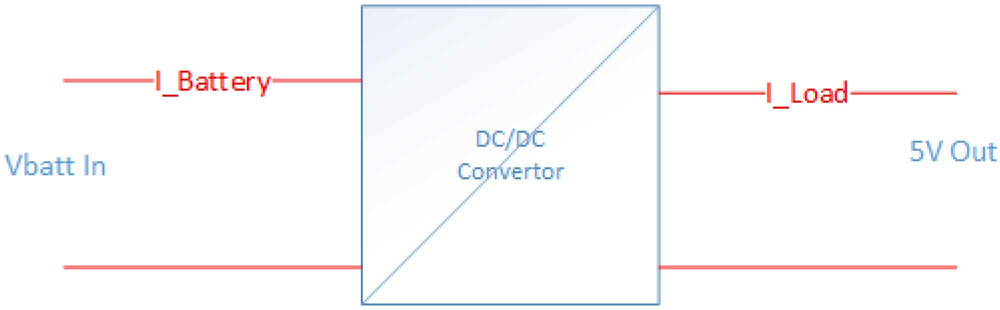


Figure 31 - Battery and Current about DC/DC Convertor

The first problem caused is that technically the voltage and current values available are not directly related. The main power supply to the logger is 'tranformed' through a DC/DC convertor. In Figure 31 the isolation is depicted by the diagonal line through the convertor. Due to this isolation barrier and the fact that the measured variables are from opposite sides of the main DC/DC convertor, this means that the two variables cannot be multiplied to give a power usage. The current flow from the battery is not monitored and neither is it fixed. The output voltage from the DC/DC however is 'constant'. The device is an LTC3112 [75]. The output voltage of the device is defined by the equation:

Equation 15 - Output Voltage calculation for LTC3112

$$V_{OUT} = 0.8V \times (1 + \frac{R1}{R2})$$

Where R1 and R2 are 845 KΩ and 158 KΩ respectively

$$0.8 \times \left(1 + \frac{845K}{158K}\right) = 5.078V$$

This voltage – dependant on resistor accuracy, can be used for power calculations. If the value of 5.078 V is used, then the power drawn by the system (from the DC/DC convertor can be calculated.

Equation 16 – Conversion between Power and Current

$$P_{OUT} = 5.078V \times I\_Load$$

To calculate the current draw on the power supply, the efficiency of the DC/DC convertor must be considered. The datasheet information is shown in Figure 32. This shows that with the device run correctly by the processor the efficiency of the device depends on the current drawn. It is a safe assumption that when the unit is functioning normally it is not drawing the higher currents shown in PWM operation, therefore the efficiency is likely to be 80% (or less). Therefore, to calculate the values for I_battery the efficiency must be built into the equations.

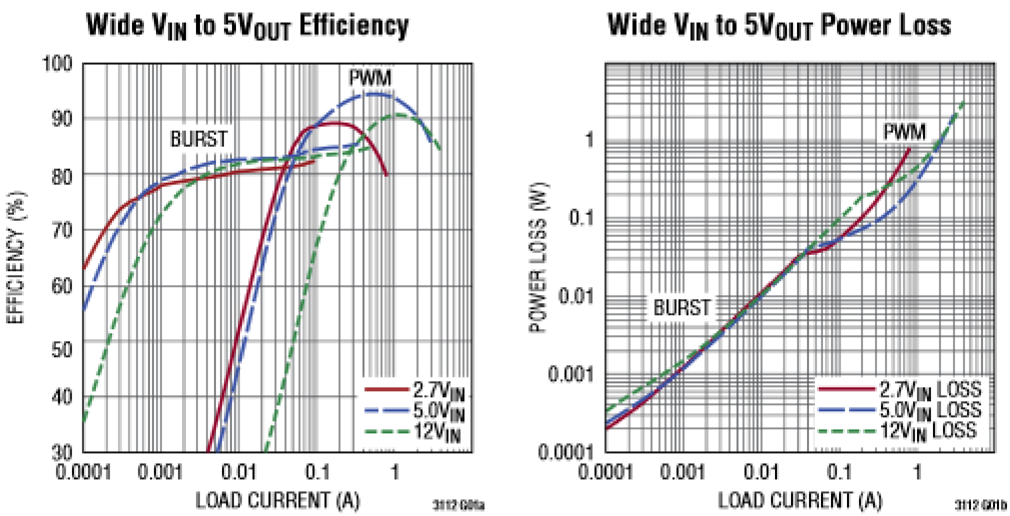


Figure 32 - LTC3112 Efficiency & Power Loss Graphs [75]

For an ideal DC/DC convertor the power entering the device is the same as the power existing the device i.e.

Equation 17 - Ideal DC/DC Convertor Equation

$$V_{IN} \times I_{IN} = V_{OUT} \times I_{OUT}$$

To integrate the efficiency of the DC/DC convertor the following must be added

Equation 18 - Non Ideal DC/DC Convertor Equation

$$V_{IN} \times I_{IN} = \left(\frac{1}{Efficiency}\right) \times (V_{OUT} \times I_{OUT})$$

So with an Efficiency of 80%. The equation reads

$$P_{IN} = 1.25 \times P_{OUT}$$

Substituting Equation 16 into Equation 18 gives

$$P_{IN} = 1.25 \times 5.078V \times I\_Load$$

Furthermore, knowing the voltage of the battery (recorded separately as V_batt), the current drawn from the battery is

$$I_{BATT} = \frac{6.3475 \times I\_Load}{V\_batt}$$

By inputting the variables recorded by the logger, this equation can be solved.

7.3. Calculating Energy Usage

The final section of chapter 7 looks at how the energy usage monitored by the EHDL can be integrated into the PC application to give better future predictions of energy usage.

7.3.1. Extrapolating Recorded Data

Until now, the values recorded by the logger have all been values for voltage, power and current (and subsequently directly transferrable). As discussed in previous sections, power is an instantaneous value – the rate at which energy is being consumed. Calculating the energy that is being consumed by the logger requires a time element. For Example, the following data in Table 22 represents the data logger waking from sleep, reading 2 DAI cards and 2 SGA cards in succession. This is an example of the information recorded by the logger as described above. The problem which requires a solution is how to extrapolate the data between the recorded points.

Table 22 contains an example range of data which would be captured by the logger. The recorded values have been converted into power usage. Displayed below in Figure 33 is the same information displayed as a bar graph.

Table 22 – Sample logger power usage data output

Time (s)	Power Usage (mW)
0.0	0.033
0.1	0.033
0.2	2944.76
0.3	2944.76
0.4	1261.76
0.5	0.033
0.6	0.033
0.7	0.033
0.8	0.033
0.9	0.033
1.0	0.033

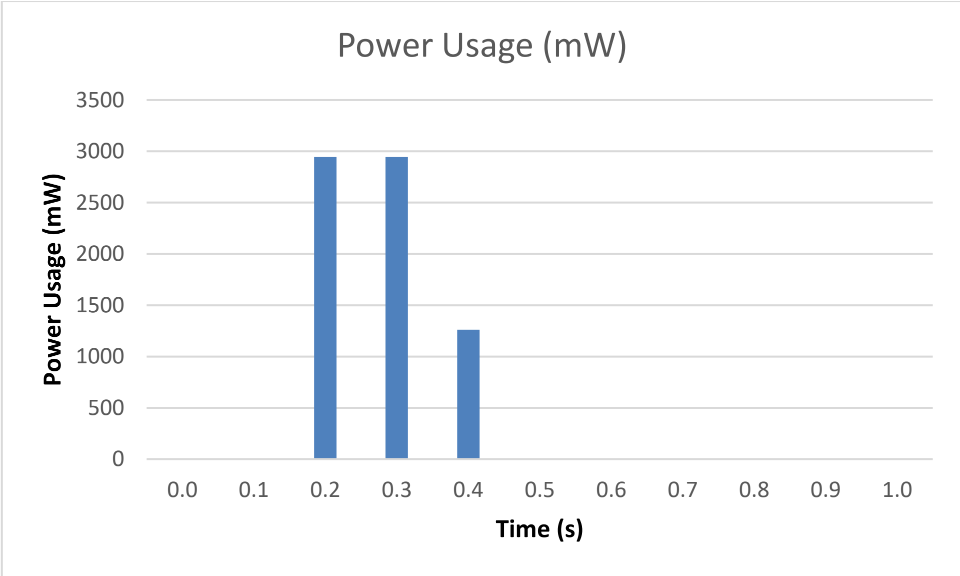


Figure 33- Power Usage against Time (Bar Graph)

At its most simple level, the Energy used by the data logger is the area under the curve or the definite integral of the function $f(x)$. Figure 33 shows the power usage of the logger across the following processes, for the first 0.2 seconds the logger is in sleep mode, due to the scale of Figure 33 these values cannot feasibly be displayed, i.e. 0.03 mW does not show on a linear scale of 3.5 W, however if a logarithmic scale is used, as in Figure 34, the energy usage in sleep mode can be displayed, showing how much more 'significant' the power usage of the three columns is. It could be determined from this that the energy usage in sleep mode is negligible, but this is inaccurate. What such a graph fails to show is the extended period for which the logger is in sleep mode. The second two columns of the graph are the energy usage from the sweep of two DAI cards, and the third column as both SGA card scans. From the graph in Figure 33 it is possible to reverse engineer the functionality which has taken place, and for this reason linear graph will be used going forwards. The reason this reverse engineering can take place is simply that the time resolution of the graph is detailed enough to break down into meaningful chunks.

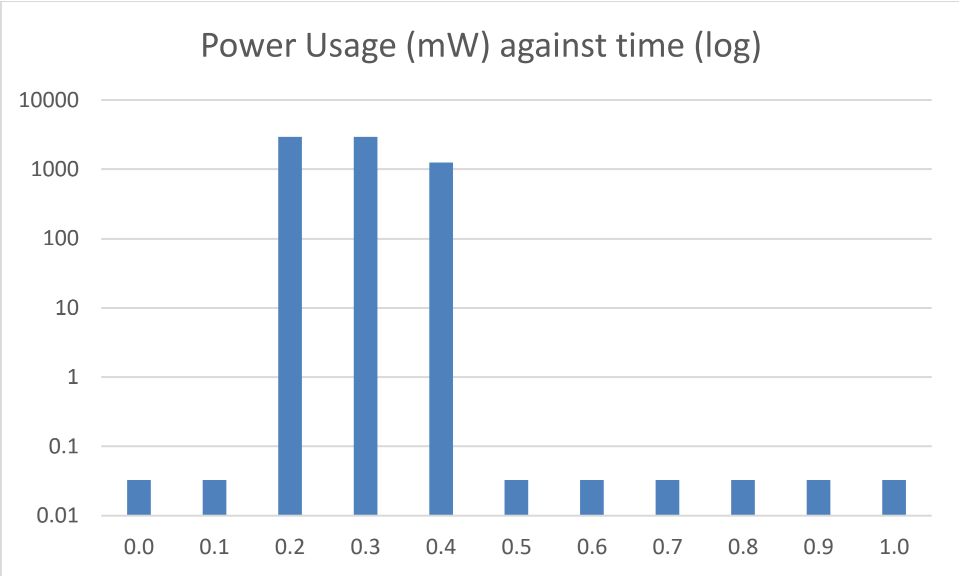


Figure 34 - Power Usage against time (Log scale)

For comparison of the power usage of different operation data modes, a bar graph is the correct way to display the data because the values are of the same scale factor.

In reality, the scatter graph shown in Figure 35 is the likely to be the most useful method for displaying the data as well as being the most suitable way of post processing the data.

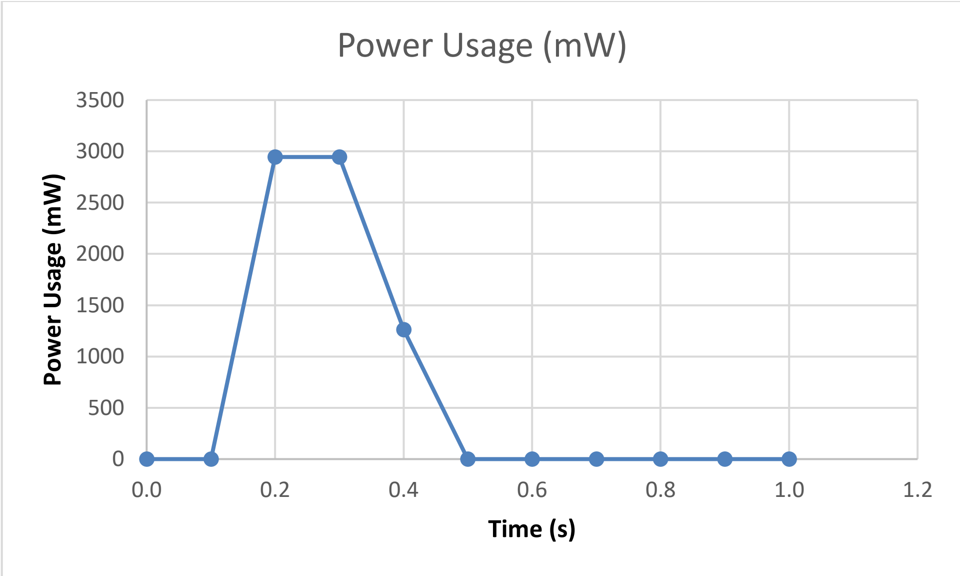


Figure 35 - Power Usage against Time (Scatter)

The proposed method for extrapolation and approximation of the data is the trapezium rule. To achieve this, the area beneath the function is divided into several vertical strips, and the area of each strip is estimated by assuming that it has the shape of a trapezium [76]. The individual areas are then summed together to get a value for the total area.

In this case, assuming that the strips are trapeziums is unnecessary, the function displayed in Figure 33 is not a continuous function (and therefore cannot necessarily be integrated mathematically). To test the accuracy of this method, the data given above will be tested by calculating the 'actual' estimated energy usage and comparing this to reverse engineered estimation from the graphs. Whilst not the most complex data set it should give a good impression of the accuracy.

Method 1 – Actual Energy Usage

The actual energy consumed can be calculated simply enough:

2 DAI card scans and 2 SGA card scans with the remainder of the single second in sleep mode gives the following

$$Total\ Energy\ Usage = (2944.76 \times 0.2) + (1261.76 \times 0.1) + (0.0033 \times 0.7)$$

This gives a total energy usage of 715.13 mWs

Method 2 – Trapezium Rule Estimation of Energy Usage

The dataset contained within Table 22 must be split down into sections, with a strip width of 0.1 seconds. The equation for area of a trapezium is

Equation 19 - Area of a Trapezium

$$\frac{a+b}{2} \times h$$

Where a and b are the two side lengths and h is the height. Table 23 is a replication of the data in Table 22, the additional column calculates the area of the trapezium to the left of the time given in the first column, this results in no value in the first row.

Table 23 - Area of Trapeziums (0.1s Timebase)

Time (s)	Power Usage (mW)	Area of Trapezium (Left)
0.0	0.033	
0.1	0.033	0.0033
0.2	2944.76	147.2397
0.3	2944.76	294.476
0.4	1261.76	210.326
0.5	0.033	63.08965
0.6	0.033	0.0033
0.7	0.033	0.0033
0.8	0.033	0.0033
0.9	0.033	0.0033
1.0	0.033	0.0033

The sum of all of the areas is 715.1511 mWs. The result accurate to 0.003%.

The shorter the time base the higher the accuracy will be; this is shown below by examining the same data but using a larger time base. Table 24 calculates the area of the trapeziums using a time base of 0.2 seconds (in place of 0.1 seconds from Table 23)

Table 24 - Area of Trapeziums (0.2s Timebase)

Time (s)	Power Usage (mW)	Area of Trapezium (Left)
0.0	0.033	
0.2	2944.76	294.4793
0.4	1261.76	420.652
0.6	0.033	126.1793
0.8	0.033	0.0066
1.0	0.033	0.0066

The sum of all of the areas is 841.3238 mWs. The result is an overestimation of 17.646% The main reason for this is shown in Figure 36, with the trapeziums re-drawn, the area beneath the graph is clearly larger than the previous graph.

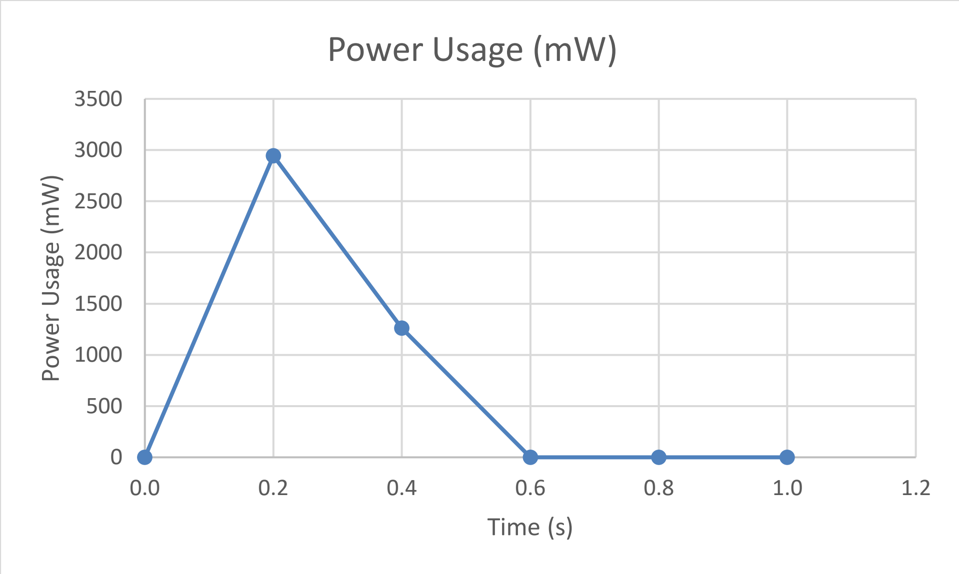


Figure 36 - Power Usage against Time (0.2s Time Base)

An extreme example of this is to change the time base to 0.5s, the peak captured in the previous readings is completely missed, this results in a reading of 0.0033 mWs. An underestimation of 99.99954%. This highlights the importance of selecting the correct time base. If the energy usage is not being recorded at a high enough resolution it will not produce meaningful data.

The ideal resolution for the logger is to record as frequently as the fastest operation, as in this setup, no data is lost, however this may not always be practicable.

7.3.2. Modifying firmware to actively manage energy usage

The findings from the previous section show the inaccuracies with the proposed method, however there needs to be a trade-off between monitoring the power consumption persistently (which will in itself use power) to get an accurate reading, and periodically monitoring the energy usage of the system to get an estimate of the power usage.

As part of the development of the software the extremes of the time base will be calculated. Ideally the extremes should be 'Always on' and 'off' however this may not be entirely possible. As part of a wider deployment, once installed in a stable capacity this functionality could be removed to improve the battery life if necessary.

The code for the processor should be as detailed in the pseudo code below with the following caveats:

- A function is defined as any mode detailed in section 0 (with the exception of sleep mode)
- When a function is required the firmware should endeavour to capture a minimum of 3 readings for energy usage. This should be as the unit powers up, during the function and as close as possible to the end of the function.
- A measured value for sleep current will be built into the model – this will prevent the logger from extrapolating the final reading across the entire sleep section

```
When the EHDL begins performing a function

Take a power usage reading

Record which Function is taking place (a recorded 'packet'
should be formed of the following information:

    • A Timestamp
    • The Function
    • The Instantaneous power usage

This will make the data usable)
```

Using the average time estimated in section 0 divide by 3.

Determine the next scan time (this will be a decision between the minimum scan time (in the user configured variable) or the value calculated in the previous step)

Set the timer

When the timer rolls over, make a power usage reading

Repeat until the function ends.

When the function end, insert a sleep ‘packet’ into the recorded data (a sleep packet is formed of a timestamp, function ‘sleep’ and the predetermined value for sleep)

During the functions, the variables should be stored in volatile RAM. As the function ends, the values should be transferred into non-volatile RAM.

The information recorded by the logger is of little meaning to the logger itself, it has more value when analysed after a download. The data recorded by the logger should be stored in a file which can be transferred to either the same web server that the data is being transferred to (although this has power implications of its own, alternatively the data could be downloaded locally from the logger over the USB connection – this is not a long term deployment proposition, but for development the function should be built in. If the power model is functioning correctly there should be ample energy available to transfer this small amount of additional data to the server during the standard data messages. This would be a primary deliverable of the software development.

7.3.3. Post Download Analysis of Logged Data

After the data has been downloaded, be that over-the-air or locally downloaded over USB, the data must be re-constructed into meaningful information. For this a PC algorithm is required

The Pseudo code below details the required functionality of the data recovery from the file downloaded from the logger.

```
Open the file downloaded from the data logger

Parse the Power usage records using the following method
```

- `Read each 'packet' as a Row in a CSV file (producing a table of 3 columns and x rows, (where x is the number of recorded packets')).`
- `Export the Raw CSV as a file for future analysis (this is to allow for human analysis of the data using programs such as Excel, also using a CSV file allows for easy transfer of data between applications)`
- `For each delta (gap between readings) calculate the power usage.`
- `For each function, calculate the sum of the energy per event. In an additional database, record each separate`

event. Recording the following fields: Time of event, Function and Power Usage

To whilst processing the data, the following variables should be calculated and displayed

- Average Power usage of each function
- Maximum and Minimum power usage of each function
- Average Duration for each function
- Maximum and Minimum duration for each function

For each of the fields above the results from the new data should be compared with the historic data. The purpose behind this is to validate that the energy usage predicted by the model is representative of the physical hardware. For Example, bench testing can give a certain level of accuracy, but one the equipment is deployed, particularly in large quantities the amount of data will significantly increase. This is part of the 'learning' phase of the model. Should the newly recorded values differ from the predicted values then the data would need to be assimilated into the power model. To assimilate the data into the model, the logger must be capable of providing this data, with information of what functionality is currently occurring. By breaking this data down and developing a method for integrating the data into the existing calculated data, it is possible to iteratively improve the model using real world data.

It has been mentioned previously that the energy monitoring of the hardware is inaccurate, but also, if the calculated data values are incorrect then the power model should have functionality to adapt to the incoming data. To overcome this, long term 'trending' will be used. This will be used to ensure that the original calculated data is given preference over the new data. The importance of this is to provide some stability against extreme results, for example if the logger recorded some anomalous or un-representative data i.e. the logger is defective or performing additional functions alongside the one it is currently monitoring. The simplest example of this is when the logger is still performing data acquisition functions whilst the 3G card is powering up. To protect against this a method is required which gives a weighting to the existing data. To implement this a moving average will be used, this will use one of the four following methods.

The first possible method is to use a simple moving average, the premise behind this method is to smooth out significant peaks or troughs from a set of data. The approach for a simple moving average is to take a mean of the previous x number of data points – i.e. If the period of the moving average is 10, then an average is taken over the previous 10 values. Typical applications for such a method are where the data contains cyclical or periodic fluctuations, in such cases the variation is eliminated.

The second possible implementation of a moving average is a Weighted Moving Average, or WMA. This method applies a weighting to the different data points. This method is often used to give the highest weighting to the latest data points in a series, it could easily be applied in reverse to give the desired effect

The third possible implementation of a moving average is an Exponential Moving Average, or EMA. This method functions similarly to the weighted moving average, but instead of fixed steps in the weighting, as is the case in weighted moving average, an exponential weighting in applied to the data series. This has the effect of a filter on the data where the weighting of the older data decreases 'infinitely' whist never reaching zero, practically the exponential moving average can be applied to all previous data (with the oldest data never being 'omitted' from the average. A major drawback of such a method is that large or small values in old data will still have an impact in the overall result regardless of their weighting.

The final method for implementing a moving average is to perform a simple moving average across the entire data series. This is known as a Cumulative moving average, this method takes all data points in the series and treats them with equal weighting. As with a simple moving average this method smooths out any peaks or troughs in the data.

A comparison of the effects of simple and cumulative moving average are contained within Figure 37. This demonstrates the difference between the two methodologies, the simple moving average (with a period of 10), shown below as a blue line, rapidly 'forgets' the previously 'programmed' data. Rapidly forming an average in the region of 45, averaging over the newly added data. The Cumulative average, shown below as red dots, reacts much more slowly to the new data, however on a larger scale once the number of 'new' data points outnumbers the 'programmed' data, the average would begin to fall, as is already apparent in the graph.

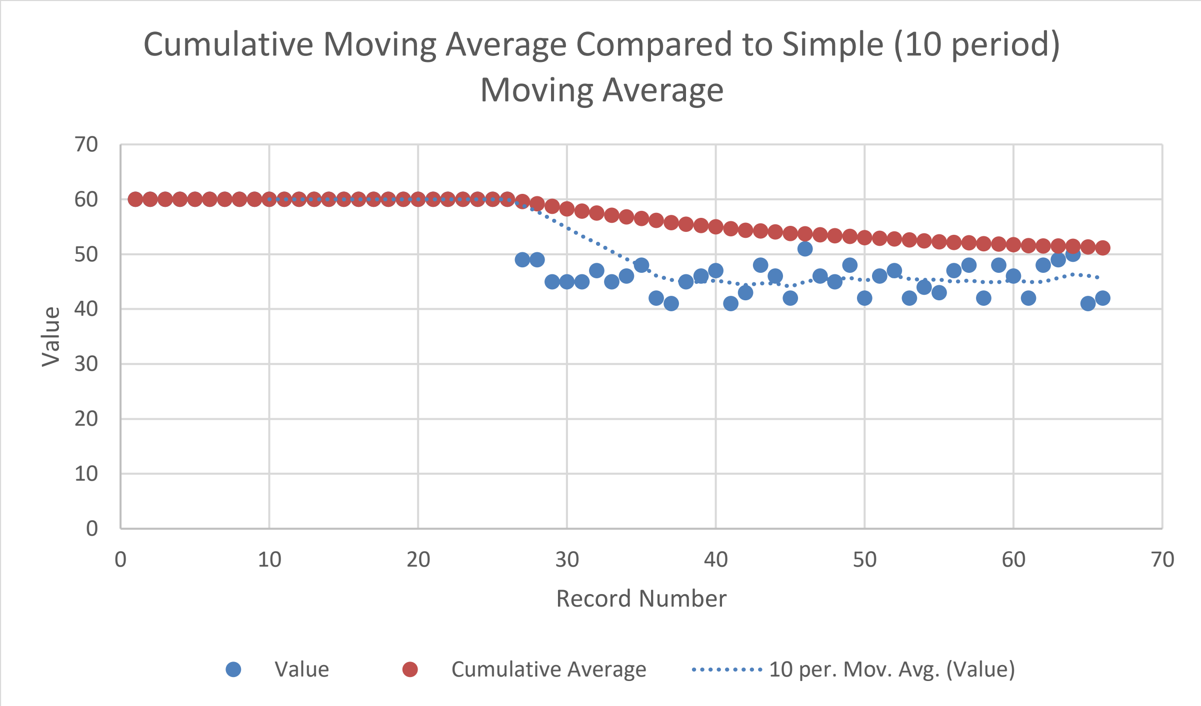


Figure 37 - Moving Average Comparison

If a cumulative moving average were to be used without any weighting a large existing data set would be required to 'protect' their input into the data. However, using a weighted methodology might be of more use if the protection of the original data is determined to be the priority.

The outcome from the averaging is fed back into the model to give a new 'fixed' value for the calculations in the model. The theory behind this approach is to reduce the number of data points in each calculation. There are two contrasting approaches to solve this problem, the first approach is the create a data repository which stores all data recorded by all loggers, this data set could then be used to more accurately predict the future energy use of the system but taking averages of all readings or by analysing the data in other ways. The drawback of this method is that as the dataset gets larger the computation time for the analysis will increase significantly impacting on the usability of the utility. The second alternative is to only store a discrete amount of data for a quicker analysis, this method works by running the analysis of the new data and applying a weighting to the existing stored value, the new 'average' value is stored as the new stored value with the remainder of the data discarded. This method retains the quick response from the data but risks losing valuable data, for example over a long term deployment the values in the data may vary due to seasonal impacts or other external influences.

A halfway house between these methods is to use techniques similar to those implemented in web applications for data presentation. These techniques revolve around having two databases, the larger database or historian contains all of the historic data and can be called upon when searches across a longer time base are required, but such calculations come with the processing delay previously discussed. The second smaller database is used for rapid data presentation to the user, this database is a copy of the most recent X values where X is customisable to preserve speed of display of a useful amount of data. For example, if the interface only requires the single most recent value for display, this can be stored and almost instantly retrieved, however it may be desirable to store the most recent hour, day or weeks data for rapid display (rather than searching a database of months and years of data).

7.4. Business Case Analysis

This section focusses on the impact the development of a power model can have on the development of a business case both for the hardware development of a new product as well as for the product itself when attempting to deploy a new or existing hardware platform into a new project. This section addresses the third hypothesis of this work.

7.4.1. Product Development

When developing a new hardware product there are many considerations which must be addressed, for example. The market need, i.e. if people will buy the products, the technical feasibility of the product, i.e. will it do the job it is required to, and finally if the product will make money.

The development of a power model answers the second of these questions. A power model can be implemented at two separate stages in the product development cycle. The first time it could be implemented is during an initial feasibility study of the newly proposed product. At this stage of product development, the aim of generating a high level power model is to determine at that high level, if the proposed approach is remotely feasible, this can be achieved by extracting data from the manufacturer’s datasheets as shown in section 4.3. At this early stage the power model can be formed of a brief overview of the proposed hardware focussing on the major components only, in the case of the energy harvesting data logger this would be focussed primarily on the main processor and 3G module. The accuracy of such a model would be less than shown previously in this work, but could prevent further development work into a product which is fundamentally impossible or impractical.

The main advantage of this method is to reduce the commercial risk in the early stages of a project, in most projects this preliminary investigation stage is not funded by a customer but in fact funded by the company so carries the greatest amount of commercial risk to the company, for every project which is taken forward from this stage there will be other projects which are not taken forwards.

The second and more detailed application of a power model in the product development cycle is during the detailed design stage. At the detailed design stage, a preliminary schematic is developed, and similarly to this work the schematics can then be analysed using the technique detailed previously in this work. By performing the design and implementation of the power model at this detailed level it is possible to highlight unknown, high power usage components such as the interrupt channels (further detail is contained in section 8.1) as well as confirming the high power usage components when taken into account with their surrounding components, by highlighting such components at this stage it is possible to re-design the circuits or select different components to overcome the high energy use. By performing this task at a design stage the cost to the business is significantly reduced.

The prototype stage of any project carries possibly the largest amount of commercial risk through the entire project, the reasoning behind this statement is that whilst prototyping a product, a significant investment is made into components, materials and assembly which will not form part of the delivered product which the company is being paid for. Whilst there are significant benefits in prototyping, hardware prototypes are the single most expensive units in the product development cycle, by identifying flawed designs prior to prototype manufacture the reduction in the number of 'prototype cycles' can more than justify the time taken to develop the power model to identify such risks.

A rough calculation is presented below to show the savings which could be made through this approach: It should be noted that the prices are approximations for a mid-sized project, with a prototype run formed of 2 manufactured units made of a single PCB (and 3 spare PCBs and other components).

Item	Cost
Components (£100 per unit, fixed for normal production)	£500
Printed Circuit Board (Approximately £20 per PCB at this size of production run)	£100
Manufacturing (Including Kitting, Manual Manufacturing process and Testing)	£400
Development (This is the time spend developing software, hardware tests and other engineering time which would be dedicated to discover the flaw in the design – this could be highly variable, but assuming the flaw is discovered in the early stages – additionally the cost of redevelopment of the hardware must be factored in	£1800

The calculation above results in a prototype cost of £2800, this saving could be made multiple times over if the design flaw is not discovered during the first iteration. The cost saving above could easily be over a week of Engineering resource dedicated to the power model and once the process for implementing the power model is in place the time to generate the model for a new hardware platform would be significantly reduced in comparison to the time taken in this work as the structure would already be in place.

7.4.2. Product Deployment

The second part of the usage of the power model in a business case is the usage of the power model in determining the situational feasibility of the logger.

Knowing if a novel product is going to work in a new environment is a process often fraught with risk, although type testing of the product can provide a level of comfort when it comes to low power systems for which the only power source is the embedded battery, the low power usage of the logger is fundamental to the ability of the logger to perform its purpose. When performing initial on site testing is it critical for the logger to perform to meet the specification laid down or confidence in the products ability to meet the specification could be significantly reduced. A major part of this is the implementation of the power model to assist in the definition of the operating parameters of the logger. By using the power model to determine the optimal scan rate and data connectivity rate the low power data logger could be deployed with system parameters capable of achieving the required functionality. Usage of the power model in this way could be used to increase confidence in the product and following successful deployments be used to further develop the model to give more accurate predictions. The model could also be used as part of specifying the maximum capabilities of the system, i.e. scan rate, battery life and other parameters. Furthermore, the model could be further developed, see section 8.4, to enable a user to enter parameters such as energy harvesting source, required scan rate etc. and the power model could recommend how to configure the logger to achieve the required functionality.

Section 7.4 has provided an overview of how the implementation of a power model could be integrated into making a business case both in the preliminary stages as well as the detail design phase. Additionally, it has been shown how the model could in integrated into the on-going support of the product.

Section 7 has focussed on the utilisation of the model generated in the previous sections, as well as investigating the techniques employed to validate the accuracy of the model against real world experience and how to iteratively improve the model to increase the accuracy of future deployments.

8. Future Work

This chapter covers some of the topics which have been alluded to previously in the Thesis in more depth, the sections below give more detail on how changes could be implemented to the model to expand its capability or usability.

8.1. Interrupt Channels

One topic which hasn't been covered in much depth previously is that of the interrupt channels in section 5.3.1 the energy usage for a single interrupt channel was calculated. This value of 127 mW, was removed from the model in section 6.1, This was due to the significant power draw of the channel. If an interrupt channel were added to the model and enabled, the standard battery would drain in 7 days, this is impractical for a remote monitoring application. This is partially due to the high current draw of the channel, but also due to the perpetual nature of the channel, i.e. All other high current functions are periodic so the average power usage is significantly reduced, however because the interrupt channel is always on the average power usage is large.

The solution to this issue is not necessarily a software issue, as the software functionality of the interrupt channel cannot change the power consumption of the channel, to understand how to best rectify this situation it is required to look at the hardware.

From a hardware perspective certain things can be changed. The channel is designed to be rugged against EMC, but this came with significant drawbacks. The most significant issue with the hardware design is the inclusion of the DCDC convertor, this convertor is very inefficient when only powering a single channel, although the same DCDC is used for the other channels the inefficiency is spread across multiple channels. The implications of the hardware design were not fully considered when requirements for the inputs were specified. Any functionality which operates at all times must use a minimal amount of power. The power usage of the interrupt channel is significantly above this arbitrary threshold, in that the power drawn by a single interrupt channel can drain the battery in a matter of hours (the calculated power usage of the interrupt channel was 127 mW = an alkaline PP3 battery has a capacity of 565 mAh giving a battery life of 4.4 hours.

To reduce the consumption of the logger and particularly the interrupt channel there are two proposed methods, one is an installation based solution and the second is a hardware redesign. It is also noted that the changes detailed below can equally be applied to the digital input channels of the logger to significantly reduce the power consumption of the channels as discussed in section 7.1.1.

8.1.1. Installation Based Solution

The circuitry for the interrupt channel works as follows. The Interrupt channel is enabled from the processor by driving an output pin high, then enables the power supply output from the channel to the outside world (through the DCDC mentioned in the previous section 8.1). This voltage is then passed to a sensor. This sensor then drives a single voltage back to the Interrupt channel of the data logger. This is the 'input' part of the channel.

The choice of sensor will affect the power usage of the logger. The most basic type of sensor which could be used is a switch, the physical implementation of switches can differ but functionally they act in a few simple modes – Single Pole Single Throw (SPST) or Single Pole Double Throw (SPDT). These two switch configurations can be configured as either 'Push to Make' or 'Push to Break'. By selecting various options from the options above it is possible to configure the data passed back to the logger. The following paragraph details some of the theory behind this

A digital input has two states, High (1) or Low (0). When a data logger reads data it has no understanding of the data that it is reading. Logically when a system is 'High' this is a synonym for 'On', however this may not always be the case. In safety critical applications such as control circuits, having 'Off' as a value of zero is not a viable option, in such as cases it may not be possible to distinguish between a wiring fault (open circuit) or the meaningful data. A well designed safety circuit will use a 'high' reading as the normal operation, i.e. Brakes released. Then use the 'low' reading as the 'active' state i.e. Brakes Applied. Such a system means that the system must energise into the healthy state; this means that the healthy state is distinguishable from a powered down system. Ideally a feed could be taken from two separate contacts (one active high, one active low) – meaning that a positive high signal is given in either state. In such a system two low values signals a fault.

To preserve power in a low current application such as this there are sensor configurations which lend themselves to low power consumption. For example, a tilt sensor could be configured in two different ways, both giving the same information but one using more power than the other.

- If the sensor is configured as 'Normally Closed' i.e. a Push to Break switch, then it would consume power from the data logger supply continuously until the tilt sensor was operated. When the sensor is 'tilted' the contacts would break apart, no longer conducting power, dropping the output from the sensor (input to the logger). This configuration is known as 'Active Low'

- If the sensor is configured as 'Normally Open' i.e. a Push to Make switch, then it would only consume power when the fault condition is present. When the sensor is 'tilted' the contacts would make, start conducting power, giving a high output from the sensor (input to the logger). This condition is known as 'Active High'

If the interrupt channel were to be configured with Active High sensors, the power supply drawn from the input would be reduced unless the channel were functional. This configuration would allow the logger to act as an 'early warning system' but if the condition monitored by the logger was a latched condition (i.e. the tilt sensor had fallen over). The power usage would increase causing an excessive drain on the battery.

8.1.2. Hardware Redesign Solution

The alternative solution is a hardware re-design; this is centred around reducing the power used by the hardware itself. The majority power draw of the interrupt channel is from the DCDC convertor (98%). This is partially due to a poor choice of component. The power loss comes from the following section of the datasheet:

Output

Output Voltage	• See table
Minimum Load	• 10%(6)

Figure 38 - XP Power ISF Datasheet [77]

The calculations have been made with the assumption that the DCDC is operating at minimum load (in most cases this would involve placing a resistor in series as the sensor is likely to be 'Open Circuit' or a 'Short Circuit'). For the component selected (ISF0505A) the minimum current is 200 mA. Taking into account the inefficiencies of the DCDC this is the major contributor to the power usage of the interrupt channel. If the DCDC were capable of being ‘short circuited’ it would be possible to significantly reduce the power consumption of the interrupt stage.

To calculate the current usage in this setup it requires measuring the input impedance of the digital input as the output power draw of the DCDC convertor. The digital input stage of the logger is configured for 5V inputs as shown in Figure 39.

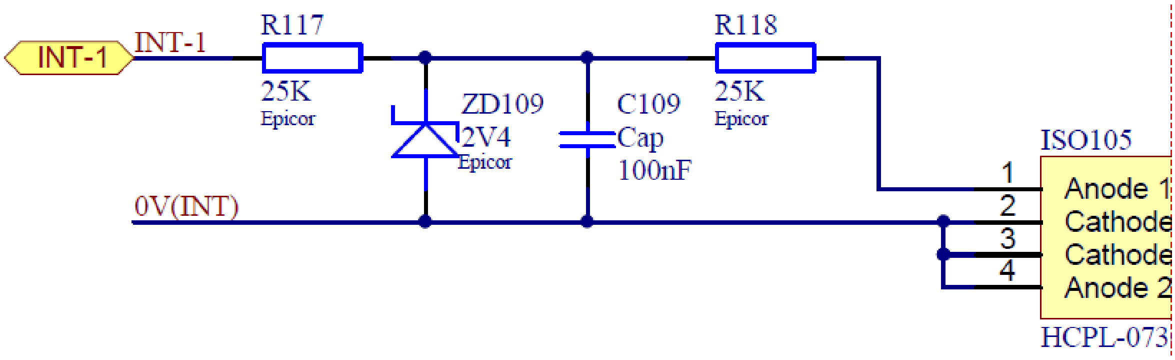


Figure 39 - Interrupt Channel Input Stage

The input impedance of the digital input is 50 kΩ. This gives a current draw from the logger power output of 100 µA, which at 5 V gives a power draw of 500 µA. If a DCDC which could functionally support such a low current draw whilst still performing correctly could be found, the power drawn from the logger would be reduced to a more acceptable level.

8.2. Data Receipt

In its initial phases the power model could not support the addition of the functionality of a data download (section 6.1). The main reason for this is that it is hard to predict the frequency at which the functionality would be required. At a worst case the function could be added to occur each time the logger connected to the remote server, this is not however a realistic representation of the application functionality. The variability of the function is based heavily on external factors, during a testing or initial deployment phase of a project it is possible that the logger could have its configuration or even firmware updated over the air. However, this then leaves the question of how often this function would be used once the system is at a stage which could be considered ‘stable’. It is reasonable to treat a firmware up-date as a very rare occurrence and a configuration update as an infrequent occurrence. For a stable installation, the frequency of a firmware update is often defined by how often a new function in the firmware is required – it is likely to be in the region of once a year if not less frequent. A configuration update is more likely as this can be the addition or removal of recording individual channels or changing the recording parameters. Additionally, a configuration update may be required if the details of the receiving server change, however this can be mitigated through other means (such as DNS addresses). The predicted frequency of a configuration is either quarterly or bi-annually.

The functionality should be preserved but would not be built into the model – primarily because the application of firmware and configuration updates is not a cyclical function and the periodicity of it is difficult to predict.

8.3. Modelling of Harvested Power Sources

An expansion to the model is an amount of simulation. This would be a significant expansion to the project, the details of the work are detailed below.

The Power model in its current guise calculates the battery life of the data logger from the maximum charge capacity of the logger to the point where it stops working. The doesn’t accurately describe the lifecycle of the data logger. In reality the logger is powered from energy harvested power sources such as solar panel, wind turbines or any other 5V source. This work can be divided down into three sections. Firstly, a study of the power input characteristics of the data logger, secondly a study of the conversion of weather data to logger battery charging and finally implementing the findings into the model as an additional module.

At present the logger is configured to generate power from three separate sources with specific functions in mind, however there is no reason this could not be further extended to modelling of vibration and thermal harvesting sources as the voltage output characteristic of the generators will likely be compatible with the existing inputs on the logger. To maximise the harvesting potential from other source it may be pertinent to add additional or different input components which are better suited.

8.3.1. Power Input Characteristics

The data logger has various methods of recharging the batteries, these have a logical priority based on the likelihood of the power source being present and the amount of power available. The PowerPath controller manages the sources to ensure that components are not damaged by cross feeds from the different supplies. The three types of power supply are USB, A wind turbine and a solar panel

The first power source which has the highest priority is the USB power supply, the reasoning for this priority is that if a USB power source is present it is assumed to be ‘unlimited’ whilst this is obviously not true, there is a limit to the amount of power that can be drawn via USB so this is unlikely to drain the external power supply. The current limit on a USB supply is significantly higher than the power requirement of the logger.

The second power source is a 12 Vdc Wind Turbine, this has a priority of 2, which means it is favoured over the Solar Panel but not the USB. If the wind turbine is generating power, it will produce more power than the solar panel, but is likely to generate power less frequently.

Finally, the solar panel is the lowest priority, the theory taken here is that the solar panel is most likely to have a small but more frequently present power generating capacity. In normal operation the logger would charge off the solar panel unless the wind turbine is generating power or USB power is connected.

The study of the hardware would need to confirm that the harvesting sources selected are sufficient to charge the batteries when in real operation. i.e. the components were functionally tested to confirm that, if a voltage were provided to the circuit, the battery charged (but this testing was conducted using a bench top power supply). The components built into the circuit are designed to harvest from the specified sources, but to correctly integrate the model further information is required.

8.3.2. Conversion of weather data into energy harvesting forecast

The second part of the modelling of the harvested energy is the most involved of the sections. A detailed plan is provided below for how the next stage of the modelling could take place.

The ambition of this work is to take previous weather data from a certain area and use the data to predict the amount of energy which could be generated. There are two parts to this, firstly importing and manipulating the weather data and secondly converting the data from weather data to a charging potential.

Historic weather data is readily available, but not necessarily in a useable format. A quick internet search for data gives the following information for the weather in Birmingham (UK).

Table 25 - Historical Monthly Averages for May [78] and November [79] .

Variable	May	November
Temperature	Average High Temperature = 16°C Average Low Temperature = 7°C	Average High Temperature = 9°C Average Low Temperature = 4°C
Relative Humidity	73%	87%
Precipitation	Average Monthly = 27mm	Average Monthly = 57mm
Hours of Sunshine Per Day	8 Hours	3 Hours
Wind	Average Wind Speed = 14 km/h (8 mph)	Average Wind Speed = 14 km/h (8 mph)
Fog	2 Days	5 Days

For a practical PC based application, a more practical method than the above is required, but the information required for such a system is easily available.

The second part of this system is converting real weather into physical charge on the batteries. This requires physical measurement of real world conditions, and to get meaningful data an extended study is required.

To conduct this study, the following results would be desired:

- An equation for converting the average temperature, average hours of sunlight and number of days with fog into an energy harvesting value for a solar panel.
- An equation for converting the average wind speed into an energy harvesting value for a wind turbine.

The application of such values is detailed in the following section (8.3.3).

The proposed methodology for finding this information is as follows. An experimental setup would need to be devised, the energy harvesting sources must be configured in such a way as to be as close to a deployment scenario as possible, a suggestion for this is that the harvesting source be connected to the power supply input of the logger and the power generated at the battery measured. This would be best done by measuring the voltage generated across the battery terminals as well as the current flowing into the battery.

These results must then be compared to the weather data for the day, week and month where possible. By measuring the data across these different time bases gives a better picture of how the current weather conditions convert into power.

The study should ideally be conducted across multiple locations and across several months, this should give a better picture of the effect that weather and other parameters have on the generated power.

It is also worth considering the seasonal variation, the amount of energy harvested in summer months will differ to that in the winter months, by harvesting both from a solar panel as well as a wind turbine gives the opportunity to harvest energy at any time.

8.3.3. Implementation of Weather Data into the Power Model

The final stage of this further work would involve integrating the findings of the previous two sections into an enhanced version of the existing power model. This enhanced model would require the following information to be input by the user:

- Energy Harvesting sources to be used from the work described above; this would be a selection from the two default power sources, but the code should be written in a way as to support the addition of further energy harvesting sources
- The Location of the installation, be that the name of a location, a post code or a GPS reference – these three options are fairly interchangeable although the location could be ambiguous. The specific accuracy of the location will vary depending on the source for the weather data. If the data set is geographically precise it may be possible to drill closer, but it is likely that the weather data will be based on the nearest town or city.

This should allow the user to determine if the installation is viable or essentially if the energy harvested from the solar panel and wind turbine is greater than the energy usage of the logger, then the logger should be capable of working perpetually. If not, the rate of discharge of the battery would be reduced by the energy harvesting but not enough to recharge the battery.

8.4. Decision Support

The final addition to the power model is to generate a more user friendly GUI, this is more of a human factors issue but can have a major impact on the usability of the model. The proposed GUI is described below.

The proposed GUI takes inspiration from the current trend of mobile applications and web application used for a wide range of applications. Specifically, applications based around finance, the prime example is the homepage of the website Wonga.com. The website, shown in Figure 40, uses 'sliders' to configure the parameters of a short term loan, as the sliders are moved the figure for total repayment is automatically calculated. By adjusting the position of the sliders the application generates the new values.

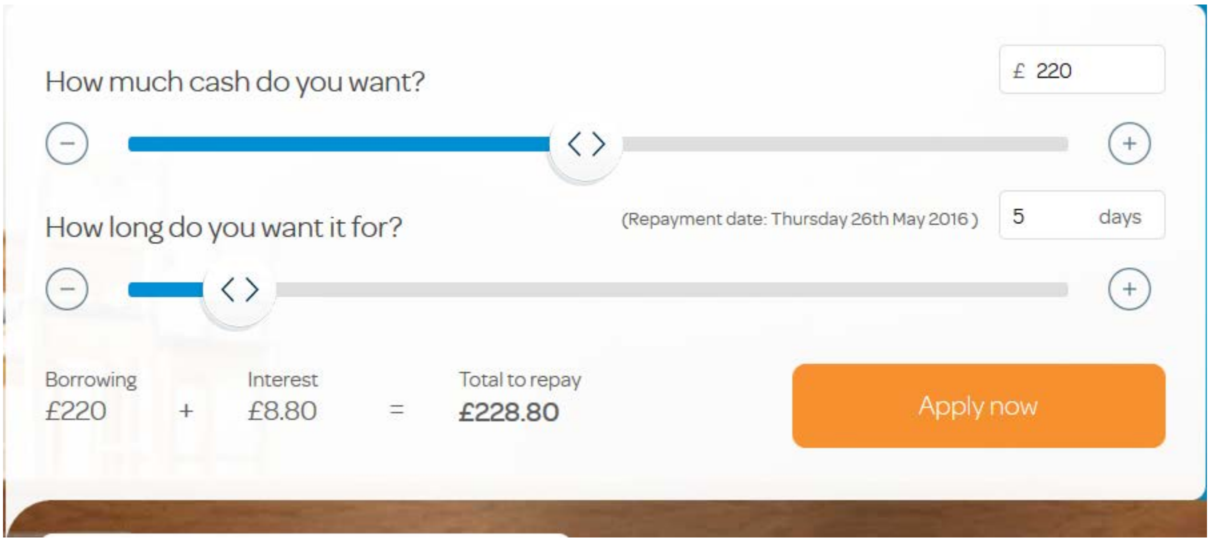


Figure 40 - Wonga.com Homepage showing sliders [80]

A setup similar to this is proposed to enable a dynamic configuration utility. The GUI should take all the values input by the user and return values for the calculated battery life, and the viability of the installation.

9. Conclusions

9.1. Findings

The objectives of this thesis were to investigate the use of a power model to assist the design of an energy harvesting data logger. The main themes to be covered are as follows:

1. Is it possible to generate a power model for specific electronic hardware and use the model to determine the operational feasibility of the hardware?
2. Is it possible to use a power model to feedback into the design of the hardware to develop more efficient hardware?
3. Can adopting a power model as a part of the design approach reduce the overall system cost of a hardware design?

For each of the main themes the findings are covered below

9.1.1. Is it possible to generate a power model for specific electronic hardware and use the model to determine the operational feasibility of the hardware?

This thesis has proved that through studying the components chosen for a circuit design that it is possible to develop a functional model for the operation of the hardware. By breaking the system down into functional sections, in a similar method to that commonly used for an FMECA, the overall task of measuring the power usage of the logger can be broken down into manageable sections, in this case study this was done by making a calculation for each part of the modular system.

Secondly, hardware design is a dynamic system, often driven by embedded firmware (or other embedded drivers), because of this the power consumption of the hardware is not constant and therefore needs to be calculated for each operating mode.

With all energy modes covered and an understanding of the system it is possible to generate a model determining the battery life of the logger. This information can then be used to practically influence the configuration of future deployments by providing more accurate predictions.

9.1.2. Is it possible to use a power model to feedback into the design of the hardware to develop more efficient hardware?

The power model revealed that the first predictions of which components would draw the most power and energy are not always accurate. Section 7.1.1 investigated the balance in energy usage between the loggers DAQ and Data Transmission modes. It was shown that even though the individual card scans for DAI and SGA cards only take 100 ms and 50 ms respectively, that with a 1 seconds scan rate these actions take place 3600 times an hour – giving an on time of 360 seconds per hour, per DAI card. An On-Time which dwarfs the 75s on-time of the Data Transmission mode even if the Data Transmission mode occurs every 15 minutes.

Furthermore, performing the task of generating the power model displayed very quickly that certain parts of the design were unfeasible for a deployment of the logger in its current format. The design intent of the interrupt channels was to allow an external trigger to start the logger recording data, however if the interrupt channels drain too high a current from the battery they could not operate functionally.

From this finding it was possible to make suggestions as to how this could be improved in a further iteration of the design. This finding also displayed the risk of adopting the policy of 'we've always done it that way', e.g. the DCDC convertor was chosen due to its known EMC performance but attention was not paid to certain functional characteristics which fundamentally damage its operation.

9.1.3. Can adopting a power model as a part of the design approach reduce the overall system cost of a hardware design?

One significant risk of any design project is the cost of building up hardware and then needing to re-design the hardware due to errors, changes in scope or if the hardware doesn't meet the design criteria. The requirement for the logger to meet a power requirement can be added to this list.

When a software project has a problem, the changes can be applied in a short time and the software 're-compiled' at a minimal time and cost. If a hardware project has a problem this often cannot be rectified in a short timescale. The calculations in section 7.4 show that the time spent developing the model can easily detect flaws in both the design as well as confirming the designed schematic/hardware can meet the design specification.

The output from this thesis is that there is definitely value in performing such a task. It would be a risky option to build a new hardware project with no knowledge that it either does or doesn't comply with the system scope.

9.2. Recommendations

The findings of this Thesis point to the following recommendations:

- Creating a power model for a new design of electronic hardware is a viable and worthwhile undertaking, if the scope of a project has a requirement to operate on battery power or to use energy harvesting this task is a necessary undertaking.
- As discussed in section 7.4, the power model can create significant financial savings for a business in the riskiest part of the development cycle.
- In projects which are not looking to be powered from either a battery or energy harvesting source, this approach may be onerous for only marginal gains, however there is a case for implementation on high power projects if there is a stringent power limit from a permanent power source
- To take this work forwards, integration of the expansion work detailed in section 8.3 would be recommended. The addition of the modelling of energy harvesting sources could significantly increase the functionality of the decision support mechanic detailed in section 8.4.

9.3. Review of Approach

The approach taken in this work has focussed on making a power model for a specific hardware platform, the positive outcome of this approach is the accuracy of the model generated. The model has been designed in such a way that it can be extended to add additional cards to the existing platform.

The negative side of the model is that the transferability of the model to other hardware platforms. The approach taken can be transferred without a significant change, the stages of extractive data from schematics and datasheets remains the same as well as the data input to the model. The biggest issue is that once the data has been entered into the model the software which drives the model within excel would require changes and this would require a level of staff competence at utilising Visual Basic and the required functionality.

As part of the development of the work, significant effort was put into the technical side of the model, the extraction of data and preliminary work on the power model. As a result of this certain sections have a greater representation than others, however the level of detail undertaken has proved that a reduced or similar level of effort could produce significant savings to a business.

A. Appendix A - References

[1] Rail Accident Investigation Board, “Derailment and collision, Watford tunnel,” RAIB, 26 September 2016. [Online]. Available: https://www.gov.uk/government/news/derailment-and-collision-watford-tunnel. [Accessed 03 October 2016].

[2] H. Zervos, P. Harrop and R. Das, “Energy Harvesting and Storage 2014-2024: Forecasts, Technologies, Players,” Energy Harvesting Journal, Cambridge, 2014.

[3] Solar Impulse, “Solar Impulse Clean Technologies to Fly Around the World:,” 08 October 2016. [Online]. Available: http://www.solarimpulse.com/. [Accessed 08 October 2016].

[4] C. Weber and A. Perrels, “Modelling lifestyle effects on energy demand and related emissions,” *Energy Policy,* vol. 28, no. 8, pp. 549-566, 2000.

[5] É. Mata, A. S. Kalagasidis and F. Johnsson, “A modelling strategy for energy, carbon, and cost assessments of building stocks,” *Energy and Buildings,* vol. 56, pp. 100-108, 2013.

[6] V. A. Balogun and P. T. Mativenga, “Modelling of direct energy requirements in mechanical machining processes,” *Journal of Cleaner Production,* vol. 41, pp. 179-186, 2013.

[7] J. R. Bumby, P. H. Clarke and I. Forster, “Computer modelling of the automotive energy requirements for internal combustion engine and battery electric-powered vehicles,” *IEE Proceedings A - Physical Science, Measurement and Instrumentation, Management and Education - Reviews,* vol. 132, no. 5, pp. 265-279, 2008.

[8] J. Kiviluoma and P. Meibom, “Methodology for modelling plug-in electric vehicles in the power system and cost estimates for a system with either smart or dumb electric vehicles,” *Energy,* vol. 36, no. 3, pp. 1758-1767, 2011.

[9] A. Carroll and G. Heiser, “An Analysis of Power Consumption in a Smartphone,” in *2010 UNSENIX conference on USENIX Annual Technical Conference*, Boston, MA, 2010.

[10] M. Deruyck, E. Tanghe, W. Joseph and L. Martens, “Modelling and optimization of power consumption in wireless access networks,” *Computer Communications,* vol. 34, no. 17, pp. 2036-2046, 2011.

[11] IMB Corp, “IBM Archives: DPD chronology - page 2,” 30 August 2013. [Online]. Available: http://www-03.ibm.com/ibm/history/exhibits/dpd50/dpd50_chronology2.html. [Accessed 21 December 2015].

[12] IBM Corp, “IBM Archives: IBM 1800 data aquisition and control system,” 30 August 2013. [Online]. Available: http://www-03.ibm.com/ibm/history/exhibits/vintage/vintage_4506VV4022.html. [Accessed 21 December 2015].

[13] National Instruments, "CompactRIO Software-Designed Performance Controller - National Instruments:," 22 April 2010. [Online]. Available: http://www.ni.com/compactrio/performance-controller/?icid=HP_FG_en-GB_hil_10281402_png. [Accessed 21 December 2015].

[14] S. Etchell, D. Phillips and C. Blackburn, "Remote condition monitoring of London Underground track circuits utilising Commercial Off-The-Shelf hardware and software platforms," in *The 6th IET Conference on Railway Condition Monitoring (RCM 2014)*, Birmingham, 2014.

[15] National Instruments, "cRIO-9063 - National Instruments:," 2015. [Online]. Available: http://www.ni.com/pdf/manuals/376330a_02.pdf. [Accessed 18 August 2016].

[16] Institute of Physics, "Energy Harvesting," 24 July 2014. [Online]. Available: http://www.iop.org/resources/energy/. [Accessed 24 July 2014].

[17] Pelamis Wave Power, "Pelamis Wave Power," 30 July 2014. [Online]. Available: http://www.pelamiswave.com/. [Accessed 30 July 2014].

[18] R. Ahmadian, R. Falconer and B. Lin, "Hydro-environmental modeling of proposed Severn barrage, UK," in *Proceedings of the ICE - Energy, Volume 163, Issue 3, 01 August 2010 , pages 107 –117*, 2010.

[19] www.vintagecalculators.com, "Teal Photon," 26 July 2014. [Online]. Available: http://www.vintagecalculators.com/html/teal_photon.html. [Accessed 30 July 2014].

[20] The AA, "AA Shop | Solar Panel Car & Caravan Battery Charger:," 30 July 2014. [Online]. Available: http://shop.theaa.com/store/home/car-battery-solar-charger. [Accessed 30 July 2014].

[21] Marlec Engineering, "Remote Power Supply - Professional Applications," 30 July 2014. [Online]. Available: http://www.marlec.co.uk/remote-power-supply/professional-applications/. [Accessed 30 July 2014].

[22] rimstar.org, "Working out your power needs for an off-grid solar system," 30 March 2014. [Online]. Available: http://rimstar.org/renewnrg/calculate_your_power_needs.htm. [Accessed 7 August 2014].

[23] Linear Technology, "SmartMesh_Power_and_Performance_Estimator," Linear Technology, 2013.

[24] PV Education.org, "Solar Cell Efficiency Records," [Online]. Available: http://www.pveducation.org/pvcdrom/appendices/solar-cell-efficiency-results2. [Accessed 10 09 2016].

[25] R. Aloulou, P. O. Lucas De Peslouan, J. Armand, H. Mnif, F. Alicalapa, M. Loulou and J. D. L. S. Luk, "Improved Circuit Model Of Photovoltaic Cell For Energy Harvesting Application," *2014 17th IEEE Mediterranean Electrotechnical Conference (MELECON),* pp. 548-552, 2014.

[26] Microchip, "AN1521 - Practical Guide to Implementing Solar Panel MPPT Algorithms," Microchip, 2013.

[27] N. Sharma, J. Gummeson, D. Irwin and P. Shenoy , "Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems," *2010 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON),* vol. ., no. ., pp. 1-9, 2010.

[28] A. Cammarano, C. Petrioli and D. Spenza, “Pro-Energy a novel energy prediction model for solar and wind energy harvesting wireless sensor networks,” *2012 IEEE 9th International Conference on Mobile Adhoc and Sensor Systems (MASS),* pp. 75-83, 2012.

[29] J. Lu, S. Liu, Q. Wu and Q. Qiu, “Accurate modeling and prediction of energy availability in energy harvesting real-time embedded systems,” *2010 International Green Computing Conference,* vol. ., no. ., pp. 469-476, 2010.

[30] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava and V. Raghunathan, “Adaptive Duty Cycling for Energy Harvesting Systems,” *Proceedings of the 2006 International Symposium on Low Power Electronics and Design, 2006. ISLPED'06. ,* vol. ., no. ., pp. 180-185, 2006.

[31] J. Holler, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand and D. Boyle, From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence , 1st Edition, Stockholm: Academic Press, 2014.

[32] ZF Electronic Systems, “Energy Harvesting Wireless Switch,” 20 August 2014. [Online]. Available: http://www.cherrycorp.com/english/switches/energy%20harvesting/index.htm. [Accessed 20 August 2014].

[33] P. Lee, Z. E. Eu, M. Han and H. Tan, “Empirical Modeling of a Solar-Powered Energy Harvesting Wireless Sensor Node for Time-Slotted Operation,” *2011 IEEE Wireless Communications and Networking Conference (WCNC),* pp. 179-184, 2011.

[34] L. Reynolds and A. Wasenczuk, “Energy Harvesting for Asset Condition Monitoring and Remote Sensing in the Rail Industry,” *The Institution of Engineering and Technology International Conference on Railway Condition Monitoring, 2006.,* pp. 194-197, 2006.

[35] Perpetuum, “Perpetuum - the global leader in vibration energy harvesting:,” 22 August 2014. [Online]. Available: http://www.perpetuum.com/. [Accessed 22 August 2014].

[36] K. K. Agrawal, G. N. Pandey and K. Chandrasekaran, “Analysis of the condition based monitoring system for heavy industrial machineries,” *2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC),* vol. ., no. ., pp. 1-4, 2013.

[37] C. Roberts, Writer, *Lecture on Methods for Improving System Performance (2012-11-29).* [Performance]. University of Birmingham, 2012.

[38] R. Ghosh, B. Chatterjee, D. Dey and S. Chakravorti, “Remote Condition Monitoring of High Voltage Insulators Employing GSM,” *2013 IEEE 1st International Conference on Condition Assessment Techniques in Electrical Systems (CATCON),* pp. 292-296, 2013.

[39] A. Carroll and G. Heiser, “An Analysis of Power Consumption in a Smartphone,” in *USENIXATC'10 Proceedings of the 2010 USENIX conference on USENIX annual technical conference* , Berkeley, CA, USA, 2010.

[40] Y. Wenxian, P. J. Tanver and M. Wilkinson, “Wind Turbine Condition Monitoring and Fault Diagnosis Using both Mechanical and Electrical Signatures,” *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2008. AIM 2008.,* pp. 1296-1301, 2008.

[41] E. Gustavsson, M. Patriksson, A.-B. Strömberg, A. Wojciechowski and M. Önnheim, “Preventive maintenance scheduling of multi-component systems with interval costs,” *Computers & Industrial Engineering,* vol. 76, no. 2, pp. 390-4000, 2014.

[42] B. S. Hauge, A. M. Stevens, R. J. Loomis and A. Ghose, Reliability-centered maintenance on the Space Shuttle Program, Los Angeles: IEEE, 2000.

[43] K. T. Meselhy, W. H. ElMaraghy and H. A. ElMaraghy, “A periodicity metric for assessing maintenance strategies,” *Industrial Product-Service Systems — IPS2,* vol. 3, no. 2, pp. 135-141, 2010.

[44] D. Hickson, Interviewee, *Remote Condition Monitoring.* [Interview]. 18 February 2013.

[45] C. Welford, Interviewee, *Remote Condition Monitoring.* [Interview]. 29 May 2013.

[46] A. Burchell, Interviewee, *Remote Condition Monitoring.* [Interview]. 23 July 2013.

[47] M. C. Herkes, “Condition monitoring drives organizational change,” *The Institution of Engineering and Technology International Conference on Railway Condition Monitoring, 2006.,* pp. 7-12, 2006.

[48] European Railway Agency, “Common safety methods for Risk assessment:,” [Online]. Available: http://www.era.europa.eu/Core-Activities/Safety/Safety-Management-System/Pages/Risk-Assessment.aspx. [Accessed 27 May 2016].

[49] Rail Safety and Standards Board Ltd, “CSM for risk evaluation and assessment:,” RSSB, [Online]. Available: http://www.rssb.co.uk/improving-industry-performance/the-legislative-framework/csm-for-risk-evaluation-and-assessment. [Accessed 27 May 2016].

[50] C. Busch, K. Gjønnes and M. Tveraaen, “Common Safety Method or Best Practice - A Practical Implementation of CSM-RA,” in *International Railway Safety Council*, Berlin, 2014.

[51] O. S. Krasheninin, The Effect of Locomotive Hours on Adjusting Periodicity of Maintainance Service and Current Repair, Khar'kov: Ukrainian State Academy of Railway Transport, 2015.

[52] H. Tan, P. W. Q. Lee, W. K. G. Seah and Z. A. Eu, “Impact of Power Control in Wireless Sensor Networks Powered by Ambient Energy Harvesting (WSN-HEAP) for Railroad Health Monitoring,” *WAINA '09. International Conference on Advanced Information Networking and Applications Workshops, 2009. ,* pp. 804-809, 2009.

[53] Network Rail, “Dawlish - Storm damage - Timetables and travel - Network Rail,” 22 August 2014. [Online]. Available: http://www.networkrail.co.uk/timetables-and-travel/storm-damage/dawlish/. [Accessed 22 August 2014].

[54] Network Rail, “Rail services to resume through Hatfield and Stainforth - Network Rail:,” 22 August 2014. [Online]. Available: http://www.networkrail.co.uk/news/2013/june/Rail-services-to-resume-through-Hatfield-and-Stainforth/. [Accessed 22 August 2014].

[55] BBC News, “BBC News - Rail services resume at Harbury Tunnel landslip site:,” www.bbc.co.uk, 17 February 2014. [Online]. Available: http://www.bbc.co.uk/news/uk-england-26223623. [Accessed 22 August 2014].

[56] BBC News, “BBC News - Hereford-to-Malvern rail line still closed after hole opened:,” www.bbc.co.uk, 25 September 2012. [Online]. Available: http://www.bbc.co.uk/news/uk-england-hereford-worcester-19710125. [Accessed 22 August 2014].

[57] BBC News, “BBC News - Trains from Hereford to Great Malvern disrupted over 'badger' hole:,” www.bbc.co.uk, 25 June 2012. [Online]. Available:

http://www.bbc.co.uk/news/uk-england-hereford-worcester-18578214. [Accessed 22 August 2014].

[58] RSSB / Future Railway Wessex Alliance, “£3-million Future Railway competition seeks remote condition monitoring solutions,” 7 October 2013. [Online]. Available: https://connect.innovateuk.org/documents/2800720/3676352/%C2%A33million%20EIT%20Remote%20Condition%20Monitoring%20Comp%20press%20release. [Accessed 22 August 2014].

[59] RSSB, “RSSB1709 - Remote Condition Monitoring Challenge - Invitiation to Tender,” RSSB, London, 2013.

[60] L. Zhong, *Power Consumption by Wireless Communication,* Houston: RICE University, 2018.

[61] E. Perla, A. Ó Catháin, R. S. Carbajo, M. Huggard and C. McGoldrick, “Realistic energy modelling for wireless sensor network environments,” *PM2HW2N '08 Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks,* vol. 3, no. 1, pp. 35-42, 2008.

[62] GKN Land Systems, “GKN Hybrid Power Description Video,” [Online]. Available: http://www.gkn.com/landsystems/brands/hybrid-power/technology-and-innovation/Pages/video-library.aspx. [Accessed 3 April 2015].

[63] The Electronic Storage Network, “Energy Storage Network Consultation Document 2014,” November 2014. [Online]. Available: http://www.electricitystorage.co.uk/files/6214/1719/4439/141103_ESN_consultation_document_V.1.0.pdf. [Accessed 3 April 2015].

[64] Energy Storage Association, “Hydrogen Energy Storage,” 3 April 2015. [Online]. Available: http://energystorage.org/energy-storage/technologies/hydrogen-energy-storage. [Accessed 3 April 2015].

[65] D. J. MacKay, Sustainable Energy - Without the Hot Air, Cambridge, England: UIT, 2009.

[66] Micrel Incorporated, “MIC5225 Ultra-Low Quiescent Current 150mA uCap Low Dropout Regulator,” Micrel Inc, San Jose, CA, 2007.

[67] Silicon Laboratories, “EFM32GG380F1024-QFP100 ARM Cortex-M3 32-bit Microcontroller | Silicon Labs:,” [Online]. Available: https://www.silabs.com/products/mcu/32-bit/efm32-giant-gecko/pages/EFM32GG380F1024-QFP100.aspx. [Accessed 06 August 2015].

[68] Cinterion, “PHS8-E Hardware Interface Description,” Cinterion a Gemalto Company, 2012.

[69] Cinterion, “Application Note 26: Power Supply for Wireless Applications,” Cinterion a Gemalto company, 2011.

[70] Farnell, “NTA2712 - GP BATTERIES - Rechargeable Battery, Lithium Ion, 4400 mAh, 7.4 V, Wire Leads | Farnell element14:,” 4 June 2016. [Online]. Available: http://uk.farnell.com/gp-batteries/nta2712/battery-li-ion-7-4v-4400mah-2s1p/dp/1848661. [Accessed 4 June 2016].

[71] Rapid Electronics, “Duracell MN1400B Plus C Cell Battery- Pack of 2 | Rapid Online:,” 4 June 2016. [Online]. Available: http://www.rapidonline.com/duracell-mn1400b-plus-c-cell-battery-pack-of-2-18-3265. [Accessed 4 June 2016].

[72] Tadiran Batteries, “LTC Batteries SL-2770 Size: C,” 4 June 2016. [Online]. Available: http://uk.rs-online.com/web/p/products/6684556/?grossPrice=Y&cm_mmc=UK-PLA-_-google-_-PLA_UK_EN_Batteries-_-Non-Rechargeable_Batteries&mkwid=sPCRuB4Ca_dc|pcrid|88056613443|pkw||pmt||prd|6684556&gclid=Cj0KEQjw-Mm6BRDTpaLgj6K04KsBEiQA5f20Eyd-xMZ3wwFJ-mRWgU3. [Accessed 4 June 2016].

[73] CPC Fanell, “ENERGIZER - 6LR61DP12 - BATTERY, ALKALINE PP3 9V PK12 | Premier Farnell CPC UK:,” 4 June 2016. [Online]. Available: http://cpc.farnell.com/1/1/35666-energizer-6lr61dp12-battery-alkaline-pp3-9v-pk12.html. [Accessed 4 June 2016].

[74] Senceive Ltd, “Wireless Sensor System - Senceive:,” Sattelite Creative, [Online]. Available: http://www.senceive.com/products/wireless-sensor-system/. [Accessed 20 August 2016].

[75] Linear Technology, “LTC3112 15V, 2.5A Synchronous Buck-Boost DC/DC Convertor,” Linear Technology Corporation, Milptas, 2010.

[76] A. Croft and R. Davison, Mathematics for Engineers: A Modern Interactive Approach (Third Edition), Harlow: Pearson Education Limited, 2008.

[77] XP Power, 1 Watt ISF Series DCDC Convertor Datasheet, XP Power, 2012.

[78] Myweather2.com, “May Climate History for Birmingham,” [Online]. Available: http://www.myweather2.com/Holiday-Destinations/United-Kingdom/Birmingham/climate-profile.aspx. [Accessed 20 May 2016].

[79] Myweather2.com, “November Climate History for Birmingham,” [Online]. Available: http://www.myweather2.com/Holiday-Destinations/United-Kingdom/Birmingham/climate-profile.aspx?month=11. [Accessed 20 May 2016].

[80] WDFC UK Ltd, “Short Term Loans | Credit for the real world | Wonga:,” [Online]. Available: https://www.wonga.com/. [Accessed 21 May 2016].

[81] I. Pohl, *A Practical Approach to Power Management for Harvesting Energy,* Cambridge: Linear Technology, 2013.

[82] R. J. Preece, T. M. Hanif, R. J. Amos and E. J. C. Stewart, “An energy management led approach to configuration and deployment of energy harvesting data loggers to monitor trackside assets,” in *6th IET Conference on Railway Condition Monitoring (RCM 2014)*, Birmingham, 2014.

[83] M. Mansour, “The suprising thing about Quality in Agile Projects,” 7 August 2013. [Online]. Available: http://agilebench.com/blog/the-surprising-thing-about-quality-in-agile-projects. [Accessed 3 April 2015].

[84] R. J. Preece, T. M. Hanif, R. J. Amos and E. J. C. Stewart, “Deploying a low energy power model within data loggers and sensors for the remote monitoring of assets,” in *The Stephenson Conference - Research for Railways*, London, 2015.

[85] RWE Innogy GmbH, “RWE Innogy opens third wind farm in Italy,” RWE Innogy GmBH, Essen, 2011.

B. Appendix B - Power Model Source Code

Code Module 1 – Hardware Type

```
Sub HardwareType()

    HWType = Application.Worksheets("Config File").Range("B1").Value

    'MsgBox HWType

    Select Case HWType

        Case Worksheets("Hardware Type").Range("A1").Value

            HT = 1

            For i = 1 To 7

            Temp = Worksheets("Hardware Type").Cells(HT, i + 1)

            Worksheets("Config File").Cells(i + 2, 2).Value = Temp

            Next i

        Case Worksheets("Hardware Type").Range("A2").Value

            HT = 2

            For i = 1 To 7

            Temp = Worksheets("Hardware Type").Cells(HT, i + 1)

            Worksheets("Config File").Cells(i + 2, 2).Value = Temp

            Next i

        Case Worksheets("Hardware Type").Range("A3").Value

            HT = 3

            For i = 1 To 7

            Temp = Worksheets("Hardware Type").Cells(HT, i + 1)

            Worksheets("Config File").Cells(i + 2, 2).Value = Temp

            Next i

        Case Worksheets("Hardware Type").Range("A4").Value

            HT = 4

            For i = 1 To 7
```

```
            Temp = Worksheets("Hardware Type").Cells(HT, i + 1)

            Worksheets("Config File").Cells(i + 2, 2).Value = Temp

            Next i

        Case Worksheets("Hardware Type").Range("A5").Value

            HT = 5

            For i = 1 To 7

            Temp = Worksheets("Hardware Type").Cells(HT, i + 1)

            Worksheets("Config File").Cells(i + 2, 2).Value = Temp

            Next i

        Case Worksheets("Hardware Type").Range("A6").Value

            HT = 6

            For i = 1 To 7

            Temp = Worksheets("Hardware Type").Cells(HT, i + 1)

            Worksheets("Config File").Cells(i + 2, 2).Value = Temp

            Next i

        End Select

End Sub
```

Code Module 2 – Time & Power Calculations

```
Sub TimeCalcs()

    With Worksheets("Config File")

    Card1ScanF = .Range("B12").Value

    Card2ScanF = .Range("B13").Value

    Card3ScanF = .Range("B14").Value

    Card4ScanF = .Range("B15").Value

    Card5ScanF = .Range("B16").Value

    Card6ScanF = .Range("B17").Value

    GSMCardF = .Range("B18").Value
```

```
CPUCard = .Range("B3").Value

End With


If Card1ScanF > 0 Then

    Multiplier1 = 3600 / Card1ScanF 'Number of Scans Per Hour

Else

    MsgBox ("DIV!0 - No Scan Frequency Detected")

End If

If Card2ScanF > 0 Then

    Multiplier2 = 3600 / Card2ScanF 'Number of Scans Per Hour

Else

    MsgBox ("DIV!0 - No Scan Frequency Detected")

End If

If Card3ScanF > 0 Then

    Multiplier3 = 3600 / Card3ScanF 'Number of Scans Per Hour

Else

    MsgBox ("DIV!0 - No Scan Frequency Detected")

End If

If Card4ScanF > 0 Then

    Multiplier4 = 3600 / Card4ScanF 'Number of Scans Per Hour

Else

    MsgBox ("DIV!0 - No Scan Frequency Detected")

End If

If Card5ScanF > 0 Then

    Multiplier5 = 3600 / Card5ScanF 'Number of Scans Per Hour

Else

    MsgBox ("DIV!0 - No Scan Frequency Detected")

End If
```

```
    If Card6ScanF > 0 Then

        Multiplier6 = 3600 / Card6ScanF 'Number of Scans Per Hour

    Else

        MsgBox ("DIV!0 - No Scan Frequency Detected")

    End If

    If GSMCardF > 0 Then

        Multiplier7 = 1 / GSMCardF 'Number of Scans Per Hour

    Else

        MsgBox ("No GPS Card Selected or DIV!0 - No Scan Frequency
Detected")

    End If

    Dim Multiplier(6)

    Multiplier(1) = (Multiplier1)

    Multiplier(2) = (Multiplier2)

    Multiplier(3) = (Multiplier3)

    Multiplier(4) = (Multiplier4)

    Multiplier(5) = (Multiplier5)

    Multiplier(6) = (Multiplier6)

    ' Reset Counters for Energy usage and Time

    T_DAQ = 0

    GSM_T = 0

    Sleep_Energy = 0

    CPU_Energy = 0

    DAQ_Energy = 0

    GSM_Energy = 0



    ' Calculate On Times for Each Card and Total
```

```
Dim OnTimes(6)

For i = 1 To 6

    OnTimes(i) = Time_Calc(Multiplier(i), i)

    T_DAQ = OnTimes(i) + T_DAQ

    'MsgBox ("Time for Card " & OnTimes(i))

    'MsgBox ("Time Total " & T_DAQ)

Next i

MsgBox ("DAQ Time Total: " & T_DAQ)

'Calculate 3G Module On Time

If GSMCardF <> 0 Then

    With Worksheets("3G Card")

        GSM_T = Multiplier7 * (.Range("C26").Value / (1000))

    End With

Else

    GSM_T = 0

End If

MsgBox ("3G Time Total: " & GSM_T)

' Calculate Total System On Time and Sleep Time

Total_Time = T_DAQ + GSM_T

MsgBox ("Total On Time " & Total_Time)

Sleep_Time = 3600 - Total_Time

MsgBox ("Sleep Time: " & Sleep_Time)

' Calculate Energy Usage in Sleep Mode

With Worksheets("CPU Card")

    Sleep_Energy = (.Range("B24").Value / 1000) * ((Sleep_Time) / 3600)

End With

MsgBox ("Energy Used By Sleep Mode: " & Sleep_Energy)
```

```
    ' Calcualte the Energy Usage in DAQ mode

    Dim Card_Energy(6)

    For i = 1 To 6

        Card_Energy(i) = Energy_Calc(Multiplier(i), OnTimes(i), i)

        DAQ_Energy = Card_Energy(i) + DAQ_Energy

        MsgBox ("Energy Use for Card " & Card_Energy(i))

        'MsgBox ("Energy Total " & DAQ_Energy)

    Next i

    'Add CPU Energy Usage during DAQ

    With Worksheets("CPU Card")

        CPU_Energy = (.Range("B25").Value / 1000) * (T_DAQ / 3600)

        MsgBox ("CPU_Energy: " & CPU_Energy)

        DAQ_Energy = CPU_Energy + DAQ_Energy

    End With

    MsgBox ("Energy Used by DAQ Mode: " & DAQ_Energy)

    ' Calculate 3G Module Energy Usage

    With Worksheets("3G Card")

        GSM_Energy = (.Range("B26").Value / (1000)) * (GSM_T / 3600)

    End With

    MsgBox ("Energy Used by 3G Module: " & GSM_Energy)

    ' Calculate Total Energy Usage

    Total_Energy = Sleep_Energy + DAQ_Energy + GSM_Energy

    'Write Energy Value to Excel

    Worksheets("Config File").Cells(21, 2).Value = Total_Energy

End Sub

Function Time_Calc(Multi, i)


    With Worksheets("Config File")
```

```
    Card1Type = .Range("B4").Value

    Card2Type = .Range("B5").Value

    Card3Type = .Range("B6").Value

    Card4Type = .Range("B7").Value

    Card5Type = .Range("B8").Value

    Card6Type = .Range("B9").Value

    End With

    Dim CardType(6)

    CardType(1) = (Card1Type)

    CardType(2) = (Card2Type)

    CardType(3) = (Card3Type)

    CardType(4) = (Card4Type)

    CardType(5) = (Card5Type)

    CardType(6) = (Card6Type)

    Select Case CardType(i)

        Case "6211.01"

            'MsgBox ("DAI Card")

            With Worksheets("DAI Card")

            CardTime = ((.Range("C25").Value) / (1000)) * Multi

            'Store Time In Seconds

            'MsgBox ("DAI On Time Per 1 Hour = " & CardTime)

            End With

        Case "6212.01"

            'MsgBox ("SGA Card")

            With Worksheets("SGA Card")

            CardTime = ((.Range("C25").Value) / (1000)) * Multi

            'MsgBox ("SGA On Time Per Hour = " & CardTime)

            End With
```

```
        Case "None"

            'MsgBox ("Slot Empty")

        Case "N/A"

            MsgBox ("Hardware Selection Invalid")

    End Select

    Time_Calc = CardTime

End Function


Function Energy_Calc(Multi, Card_on, i)

    With Worksheets("Config File")

    Card1Type = .Range("B4").Value

    Card2Type = .Range("B5").Value

    Card3Type = .Range("B6").Value

    Card4Type = .Range("B7").Value

    Card5Type = .Range("B8").Value

    Card6Type = .Range("B9").Value

    End With

    Dim CardType(6)

    CardType(1) = (Card1Type)

    CardType(2) = (Card2Type)

    CardType(3) = (Card3Type)

    CardType(4) = (Card4Type)

    CardType(5) = (Card5Type)

    CardType(6) = (Card6Type)


    Select Case CardType(i)

        Case "6211.01"

            'MsgBox ("DAI Card")
```

```
            With Worksheets("DAI Card")

            'Calculate Energy Consumed in 1 Hour (Wh)

            'MsgBox ("Card_on Time(s): " & Card_on)

            CardPwr = ((.Range("B25").Value) / (1000)) * (Card_on / (3600))

            'in amps / in hours

            'MsgBox ("Energy Usage (Wh): " & CardPwr)

            End With

        Case "6212.01"

            'MsgBox ("SGA Card")

            With Worksheets("SGA Card")

            'Calculate Energy Consumed in 1 Hour (Wh)

            'MsgBox ("Card_on Time(s): " & Card_on)

            CardPwr = ((.Range("B25").Value) / (1000)) * (Card_on / (3600))

            'in amps / in hours

            'MsgBox ("Energy Usage (Wh): " & CardPwr)

            End With

        Case "None"

            'MsgBox ("Slot Empty")

        Case "N/A"

            MsgBox ("Hardware Selection Invalid")

    End Select

    Energy_Calc = CardPwr

End Function
```