University of Southern Queensland

Faculty of Health, Engineering & Sciences

Mansell Power Lifter Actuator Failure Analysis

A dissertation submitted by

Donald Bailey

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Engineering (Honours) (Computer Engineering),

Bachelor of Information Technology

Submitted: October, 2016

Abstract

The Mansell Power Lifter is a critical component of the Mansell Infant Retrieval System. It functions as an electrically powered stretcher which can be raised or lowered through the use of two independant DC-motor driven linear actuators. The Mansell Infant Retrieval System is a key piece of medical equipment used widely to transport critically ill infants. Over a number of years there have been reports of the linear actuators used in the Mansell Power Lifter failing unexpectedly. Due to the high reliability requirements of the Mansell Power Lifter as a medical device any failures are unacceptable, and as a result the manufacturer of the Mansell Power lifter has sought to identify the cause of the failures. The aim of this project was to attempt to identify the mechanism and cause of the actuator failures and if possible propose a potential solution.

In order to identify the failure mechanism and root cause, it was determined that data would need to be collected about the usage and operational characteristics of the linear actuators. After analysis of the situation, it was decided to construct an autonomous data collection device to collect real world usage data from the actuators. An integrated, microcontroller based, data logger device was designed and developed which could be placed on a Mansell Power Lifter and record operational data without interfering with normal operation. Analysis of the collected data was performed to identify any operational characteristics which might lead to failures. While an actual failure of the linear actuators was not observed during the testing period, the data obtained showed that the actuators were being subjected to a large number of short, sharp movements on a regular basis, which could be contributing to the limit switches in the actuators jamming. Since the definitive cause of the actuator failures could not be confirmed, given the data obtained, it was not possible to propose a guaranteed solution to the problem. Despite this, a number of potential solutions were proposed, which given further testing and analysis, could be implemented to help prevent actuator failures in the future.

University of Southern Queensland
Faculty of Health, Engineering & Sciences

ENG4111/2 *Research Project*

Limitations of Use

Dean
Faculty of Health, Engineering & Sciences

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

DONALD BAILEY

0061032854

Acknowledgments

I would like to thank my supervisors Dr John Leis and Dr John Grant-Thomson for their support and guidance throughout this project. Your advice has been invaluable in ensuring the success of this project.

I would also like to thank Wenross Holdings Pty. Ltd. for allowing this project to be conducted on the Mansell Infant Retrieval System and providing the resources and support required to make it happen. My thanks also goes to Paul Priebbenow and Anthony Vadalma at Wenross Holdings Pty. Ltd. for their continued advice and assistance throughout the project.

Finally, I would like to thank my family for their support throughout the course of undertaking this project and assistance with proof reading tasks.

Donald Bailey

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The Mansell Power Lifter is one of the three primary components of the Mansell Infant Retrieval System, which allows infants in a critical condition to be transported in a variety of medical vehicles along with all of the required medical equipment for their care. The Mansell Infant Retrieval System was invented by Dr John Grant-Thomson in the year 2000 and is currently manufactured by Wenross Holdings Pty. Ltd. in Toowoomba. The system has seen great success in the industry and is currently in use across Australia and also in Norway and Sweden. The Mansell Infant Retrieval System consists of the Mansell Power Lifter, an electrically powered stretcher, which will be the focus of this project, the Mansell Neosled which is an integrated system for mounting and powering various pieces of medical equipment, and the Mansell Neocot, which is a temperature controlled humidi-crib in which the infant is transported (Grant-Thomson 2014). A picture of a complete Mansell Infant Retrieval System can be seen in Figure 1.1.

The Mansell Power Lifter and Mansell Neocot both contain dedicated on-board battery packs to allow for their operation in the field, and are capable of being powered from a variety of road vehicle, aircraft, or mains power sources depending on the location that the system is in use. The Mansell Power Lifter is designed to allow for the Neosled and Neocot containing the patient to be raised or lowered to the most appropriate height for the medical staff and alleviate the physical strain involved in loading the system into an ambulance or medical aircraft. The system contains two linear actuators which can be

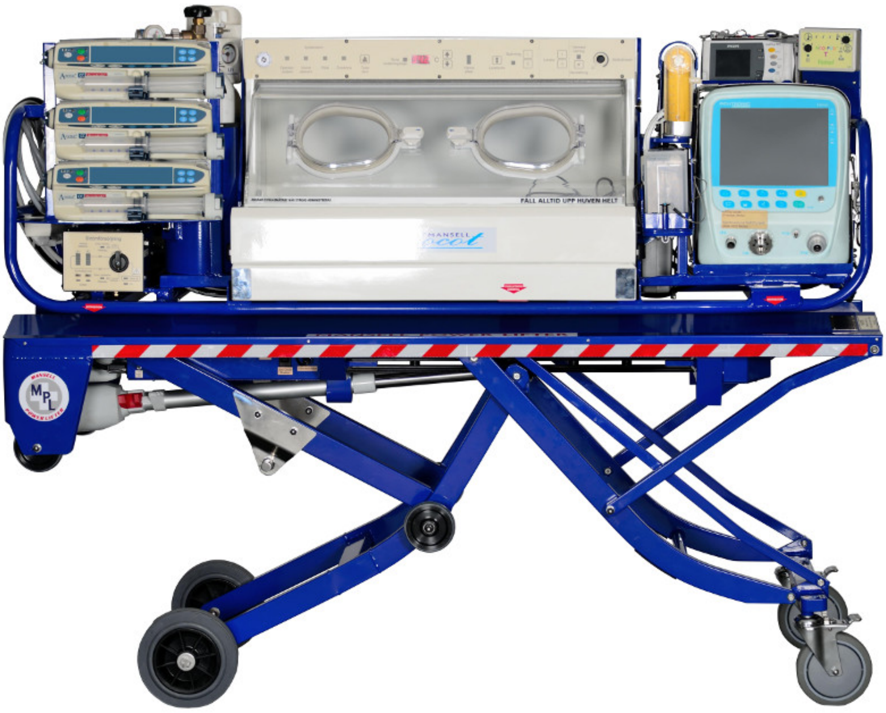


Figure 1.1: An example of a complete Mansell Infant Retrieval System (Grant-Thomson 2014).

operated independently in order to facilitate loading into an ambulance, where one set of wheels must be retracted before the other in order to allow the system to slide into the vehicle (Grant-Thomson 2014).

While the Mansell Infant Retrieval System has been in successful use for many years across Australia and overseas, there have been a number of reported occurrences of the linear actuators in the Mansell Power Lifter failing in the field. Prior to this project being conducted very little information has been known about the circumstances or characteristics of these failures and as a result Wenross Holdings Pty. Ltd. have not been able to identify the cause of the problem. As with any medical system, a malfunction of any kind is unacceptable and can put the life of the patient at risk. Even though the Power Lifter does not have a direct impact on the condition of the patient, a malfunction could significantly hinder the ability of the medical personnel to transport the patient to the destination. As such, Wenross Holdings Pty. Ltd. has sought to investigate the circumstances surrounding the actuator failures on the Mansell Power Lifter, in order to identify the cause and therefore determine an appropriate solution, to ensure the safe and effective operation of the Mansell Infant Retrieval System.

1.2 Project Motivation, Aims, and Objectives

The Mansell Infant Retrieval System is an essential tool used by many hospitals throughout Australia and overseas. The continued reliable operation of the system and all of its components, including the Mansell Power Lifter, is of utmost importance to ensure that the system can be relied upon to transport critically ill infants when required. In response to the reports of linear actuators in the Mansell Power Lifter failing, the aim of this project is to identify the root cause of the failures and propose a solution which can be implemented to ensure continued safe operation of the system. It is planned that the aims of the project will be met by working towards the following objectives:

- Design and construct an autonomous data collection device which integrates with the Mansell Power Lifter and can be deployed to an operational system in a hospital.
- Deploy one or more data collection devices to Mansell Power Lifters which are in regular use by Australian hospitals for one or more months.
- Perform stress testing of a test system at Wenross Holdings Pty. Ltd. in Toowoomba to identify conditions under which actuator failures may occur and the characteristics of such faults.
- Analyse the data obtained to identify the characteristics of actuator faults and determine the failure mechanism and cause.
- Using the information gathered propose a solution which could be implemented to prevent further actuator failures on systems in the field.
- If time and resources allow the proposed solution will be designed and an initial prototype built.

1.3 Project Methodology and Timeline

In order to achieve the objectives and outcomes of this project a methodology will need to be developed and followed. Additionally, each component of the methodology will need to take place on a timeline where certain project objectives will need to be completed before others can begin. Each of the major components of the project methodology and how they fit in to the project timeline are summarised in the following sections.

1.3.1 Analysis of Current Situation

In order to identify the cause of the linear actuator failures in the Mansell Power Lifter an understanding of how the system is used, and the conditions that it is used in, will need to be gained. It is also important to gain as much information as possible about the circumstances surrounding actuator failures and how the system behaves when a failure occurs. This component of the methodology will need to be completed first so that there is a good understanding of the problem before development and testing begins.

1.3.2 Data Collection

Though it will be necessary for data to be gathered from Mansell Power Lifters in use in the field, it will not be practical for this to be done in person or by medical staff, both due to time constraints and the nature of the system as a medical device, where the medical staff are focussed on the patient. Additionally, it would be an unreasonable and impractical task to ask the medical personnel using the system to accurately log its use on a day to day basis to the level required. Therefore, an automated system for gathering data will be required. An autonomous data logging device has been chosen to perform this task due to its ability to record numerous different parameters of operational data from the use of a Mansell Power Lifter in the field. Such data logging devices are highly suited to this application due to their flexibility and lack of impact on the operation of the system (Suzdalenko, Lazdans & Galkin 2012). The design and development of the data collection system and its ultimate deployment to the field will be the second component of the project methodology to be completed.

1.3.3 Data Analysis

Finally, the methodology for analysing the data gathered needs to be considered. Once the stress testing is complete and the data logger has been retrieved from the field, all data gathered will need to be analysed and compared in an attempt to determine the cause and characteristics of the actuator failures. Initially this analysis will be concerned with identifying portions of the data collected which correspond to a failure. Following this, the data relating to failure events can be analysed for potential clues to the failure cause. If it turns out that it is not possible to identify any failures from the data obtained by the data logger, it may be possible to identify operational characteristics from the data which are out of the norm and may point to a possible cause of the failures. This element of the project methodology will be conducted after the data collection process and will be an important step in determining what factors can be addressed to prevent the failures.

1.3.4 Solution Proposal

Once the analysis of the data is complete and informed conclusions have been drawn about the cause of the actuator failures, a potential solution will be proposed based on the knowledge obtained about the design and operation of the system. If possible this solution will be aimed at preventing actuator failures in the field with only minimal modification to the system, such that the solution could be easily implemented. Also, if it is identified that actuator failures could be prevented through modifications to their control scheme, it may be possible to build upon the work of Keys (2014) who developed an digital power control system for the Mansell Power Lifter which can be reprogrammed via software. This flexible digital control could be utilised to implement a better control scheme for the linear actuators, which could be incorporated into future versions of the Mansell Power Lifter. This will be the final stage of the project methodology to be completed. It is likely that the extent of the solution design and development will be dependent on the amount of time available.

1.4 Project Safety

In order to ensure that all elements of this project are undertaken in a safe manner, both in terms of personal safety and the safety of any modifications made to the Mansell Power Lifter, a risk assessment has been undertaken. The full risk assessment for the project is listed in Appendix F.

1.5 Dissertation Overview

This dissertation is organised as follows:

Chapter 2 contains an analysis of the Mansell Power Lifter systems and explores the design requirements of the data collection system.

Chapter 3 outlines the design considerations involved in the design of the data collection system.

Chapter 4 outlines the design and development of the hardware aspects of the data collection system.

Chapter 5 describes the design and development of the software aspects of the data collection system.

Chapter 5 considers the analysis of the data obtained from the data loggers and contains a discussion of the results. It also explores potential designs for a solution to the actuator failure problem.

Chapter 7 concludes the dissertation and suggests further work which could be conducted to explore the actuator failures further and implement the potential solutions proposed.

Chapter 2

System Analysis and Design Requirements

2.1 Chapter Overview

This chapter will provide an overview of the design and operation of the Mansell Power Lifter and Mansell Infant Retrieval System as a whole. An analysis of the current information known about the linear actuator failures will be conducted and the relevant literature pertaining to their failures reviewed. The analysis of the Mansell Power Lifter system design and actuator failure literature will enable the data collection system to be designed such as to maximise the usefulness of the data obtained. The objective of this chapter is to determine which operational and environmental parameters should be monitored by the data collection system in order to best diagnose the cause of the failures.

2.2 System Overview

2.2.1 Mansell Infant Retrieval System

The Mansell Infant Retrieval System consists of three primary components, the Mansell Neocot, the Mansell Neosled, and the Mansell Power Lifter. The Mansell Neocot is at the heart of the system and is where the infant under care is placed. The Neocot is a heated, temperature controlled, capsule which keeps the infant being transported at the optimal temperature. The Neocot itself is mounted within the Mansell Neosled along with all of the medical equipment required by the medical staff. The Neosled is a customisable frame which allows for a wide range of medical equipment to be securely mounted in proximity to the patient in the Neocot. It also contains a battery based power supply system which can be used to power the Neosled and all of the other equipment in use. The third component of the Mansell Infant Retrieval System and the main focus of this project is the Mansell Power Lifter. The Mansell Power Lifter is a purpose built, electrically operated, adjustable height stretcher, which is designed to carry the Neosled and Neocot (Grant-Thomson 2014).

2.2.2 Mansell Power Lifter

The Mansell Power Lifter features two electrically powered linear actuators which allow the entire system to be raised from 270mm to 1000mm with zero operator effort. While a typical load is only about 60kg the Mansell Power Lifter is capable of lifting up to 200kg. The linear actuators can be operated independently allowing for different heights between the front and rear of the system. This is necessary in order to allow the system to be loaded into a road ambulance where the front of the system needs to be slid into the vehicle while the rear is still on the ground. The linear actuators in use in the Mansell Power Lifter are Linak LA34 units with a short variant being used to actuate the front end of the system and a long variant being used in the rear (Grant-Thomson 2014).

The deck of the Mansell power lifter features a Lifeport Clipdeck locking system which allows for the Neosled to be loaded into helicopters and fixed wing aircraft which use the same locking system. It is necessary for the Neosled and Neocot to detach from the Mansell Power Lifter since there is often not enough room in some transport vehicles to accommodate the full Mansell Infant Retrieval System. In such cases the Mansell Power Lifter must be left behind and the Neosled loaded onto another Mansell Power Lifter or compatible stretcher at the destination. A Mansell Power Lifter without a Neosled attached can be seen in Figure 2.1 (Grant-Thomson 2014).

Figure 2.1: The Mansell Power Lifter without the Mansell Neosled and Mansell Neocot attached (Grant-Thomson 2014).

The Mansell Power Lifter comprises of several main components, the control box, power box, battery packs, and two linear actuators. A block diagram overview of the system can be seen in Figure 2.2. The control box consists of the individual operator controls for the actuators, a power on/off switch, and an array of status indicator lights. The control box also contains a safety sensor which will prevent the operation of the Mansell Power Lifter while it is locked into an ambulance or other medical vehicle (Keys 2014).

The power box is the main hub of the electrical system and performs both actuator control and battery charging functionality for the dual 12V battery packs. The actuator control circuitry is capable of powering the actuators from either 12V or 24V by changing the configuration of the battery packs from parallel to series respectively. By automatically switching from 12V to 24V operation when the load on the power lifter allows, the speed at which the actuators can raise or lower the system can be greatly increased. The control circuitry will switch the actuators from 12V mode to 24V mode if one or both of the actuators have been running for at least two seconds and the total combined current flowing through them is less than 11A. It should also be noted that the Mansell Power Lifter will not operate in 24V mode when an AC power source is connected (Keys 2014).

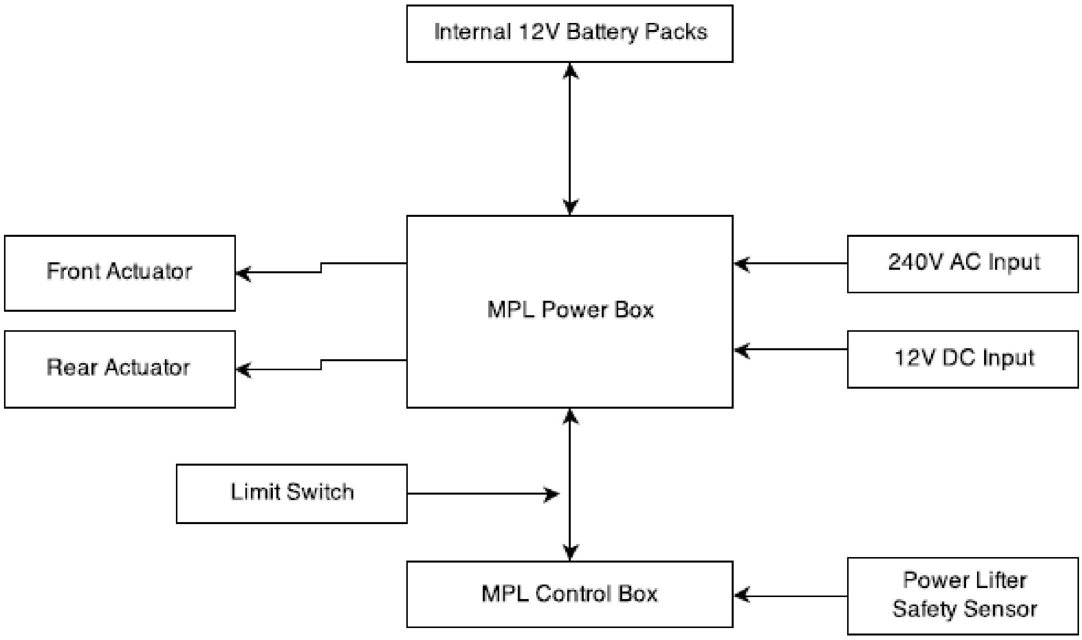


Figure 2.2: Block diagram overview of the Mansell Power Lifter electrical systems (Keys 2014).

The actuators used in the Mansell Power Lifter are LA34 DC-motor driven linear actuators from Linak. They are designed to be powered from either 12V or 24V and can provide 5500N of thrust in the push direction with a duty cycle of six minutes of operation per hour. Internally the actuators feature a printed circuit board containing two limit switches and associated components. The limit switches are mounted in parallel on the circuit board facing each other with a plastic slide sitting between them. When the shaft of the actuator reaches its limit in either direction, a small lever connected to the plastic slide is forced forwards or backwards and causing it to push on the button of the corresponding limit switch, thus disconnecting the power from the DC motor (Grant-Thomson 2014). The Linak LA34 limit switch circuit board can be seen in Figure 2.3 and a reverse engineered circuit schematic for it can be seen in Appendix B.

2.3 Known Actuator Failure Information

Wenross Holdings Pty. Ltd. keeps a record of all reported failures relating to the components of the Mansell Infant Retrieval System for warranty purposes. It is therefore relatively easy to collate information about past linear actuator failures for the Mansell

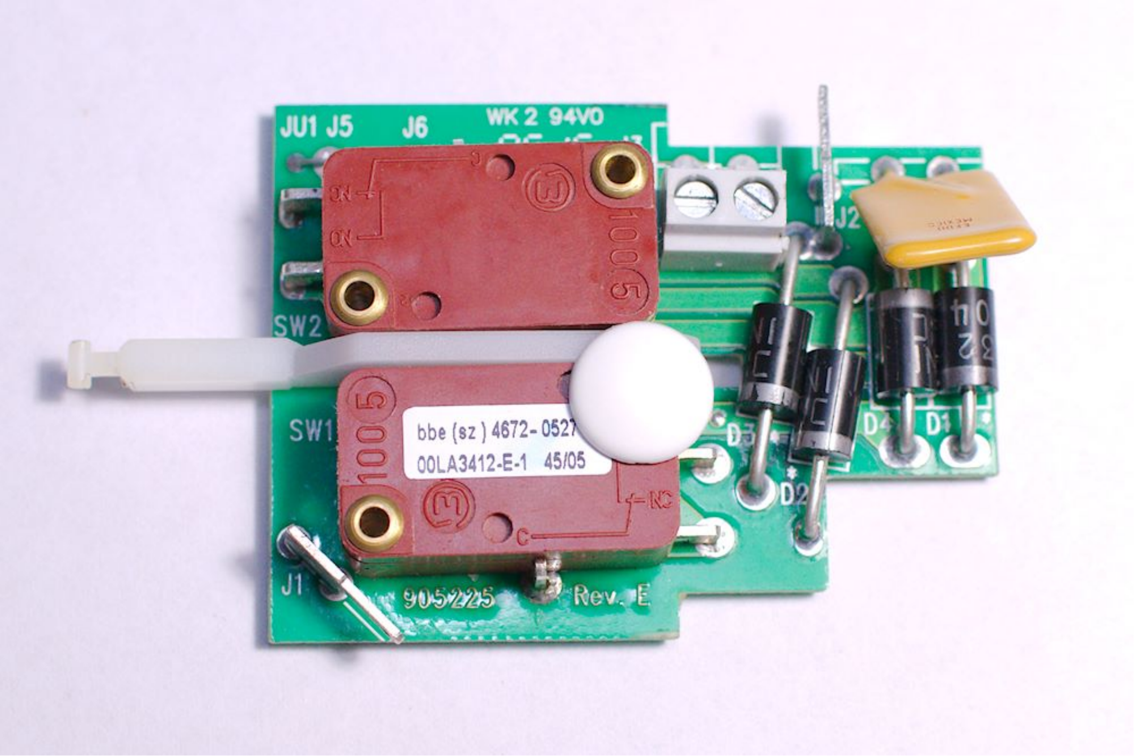


Figure 2.3: The internal limit switch circuit board from a Linak LA34 linear actuator showing the limit switch mechanism. (Photo: author)

Power Lifter and review the analysis and conclusions made at the time. After reviewing the warranty records for the period of 2007 to 2016 it was found that nine linear actuator failures had been reported for the Mansell Power Lifter out of about 100 full systems currently in use. Though the average actuator failure rate of 0.5% per year is relatively low, it is possible that the actual number of failures may be higher. Wenross Holdings Pty. Ltd. suspects that the actual number of failures may be higher than reported by users, since purchasing records show that locations with high system usage have been purchasing on average a couple of spare actuators each year. It should be noted that in all cases of reported failures the linear actuators were replaced as a safety precaution, regardless of whether or not the fault could be replicated.

2.3.1 Reported Failure Symptoms

The end user reports of actuator failures can be summarised as follows:

- Linear actuator failures only occur to one actuator at a time while the other actuator remains fully operational.

- Linear actuator failures are not linked to a particular actuator. Both the long and short actuators have been reported as failing roughly the same number of times.
- In a number of cases, when the failure occurred the actuator would operate in one direction but not the other.
- In a number of cases the actuator failure occurred when it was at one of its limits and was attempted to be moved in the opposite direction.
- In some cases hitting the faulty linear actuator with a large object (a hammer) resulted in normal operation being restored. However, records show that even after doing this the failure can reoccur at a later time.
- In some cases the failed actuator would start working again and the fault could not be reproduced.
- In one case it was noted that the failure occurred on a hot day while outside for an extended period of time.

2.3.2 Conclusions on Cause of Failures

The conclusions reached at the time of diagnosis by Wenross Holdings Pty. Ltd. in regards to the actuator failures can be summarised as follows:

- In the majority of cases the failure mechanism was attributed to problems with the actuator's internal limit switches. However, the exact failure causes recorded varied from limit switch damage to excessive lubrication of the limit switch mechanism. While this strongly points to the limit switches being the failure mechanism it is not clear what the actual cause of these failures may be.
- A number of recorded failures could not be replicated and the actuator was replaced as a safety precaution.

Since failures only occur to one actuator at a time, and both types of actuators have been reported as failing, it seems unlikely that the failure mechanism is a component of the Mansell Power Lifter power box and control systems. Additionally, given that hitting the affected actuator with a hammer can sometimes resolve the problem, and that failures have been reported to happen when the actuators are at their limits, there is a strong case for the limit switches being the failure mechanism. If there was a problem with the limit switches in the actuators jamming, it makes sense that this is most likely to occur at the end of their stroke, when the limit switches are activated. It is also conceivable that hitting the actuator with a hammer could result in a jam condition being resolved through the vibration of the impact.

The potential conclusion of the limit switches in the actuators being the failure mechanism, is backed up by the fact that during testing of a spare development Mansell Power Lifter, for the purposes of this project, a similar limit switch jam was encountered. The limit switch jam which was encountered matched the characteristics reported in the warranty claims where the affected actuator would only move in one direction. It was also confirmed that upon disassembling the actuator and unjamming the limit switch, normal operation was returned. However, it was also observed that in this case the actuator was in the middle of its travel and not at one of the limits when the limit switch jam occurred. While there is considerable evidence that the internal limit switches in the linear actuators may be the failure mechanism, it is still unknown as to what the failure cause may be. In addition to this, there continue to be inconsistencies in the reported failure symptoms of the linear actuators and little is known about the operating conditions under which the failures occurred.

2.4 System Reliability

2.4.1 Reliability Engineering

The Oxford Dictionary (2016) defines the word *reliable* as to be “Consistently good in quality or performance; able to be trusted.” It should be the aim of every engineer to develop systems that perform consistently and can be trusted by their users. However, this is particularly important in the field of medical engineering and other disciplines where the systems being worked on have critical safety implications if a failure were to occur. It should therefore be no surprise that reliability engineering plays a significant role in the design and development of systems and products in the modern engineering world.

In engineering terms, reliability can be generally accepted to mean the ability of a product or system to perform its required functions in an acceptable way over a certain time period. In addition to this both the designer and user of the system must concede that there will be certain operating conditions which must be met in order for the system to perform reliably. Finally, it must also be accepted that a system cannot be designed to work forever, and due to wear caused by normal use, will have an expected lifetime, only over which the reliability of the system should be a given. It should also be considered that excessive use, incorrect operating conditions, and a lack of maintenance may result in the actual lifetime being less than the expected lifetime (Kapur & Pecht 2014).

2.4.2 Reliability Quantification

It is often useful to quantify the reliability of a system or product via statistical means so as to determine the likelihood that one of those systems will fail. This information can then be used to review and determine whether or not changes need to be made to its design or construction to improve reliability. The most commonly used term in the statistical analysis of reliability is MTBF (Mean Time Between Failures) or MTTF (Mean Time To Failure). These terms are often used interchangeably, however the technical difference between the two is that MTBF is used for products where after being repaired the product will then be in good-as-new condition, and therefore have a constant failure rate even after being repaired (Birolini 2010). In the case of the Mansell Power Lifter, since the situation of a failed actuator can be rectified simply by replacing the actuator with a new one before returning the system to normal use, Mean Time Between Failure is the more appropriate term.

If reliability data regarding failures were to be gathered from a large population of products, which were statistically independent, the resulting plot of number of failures versus time would typically show a large number of initial failures, followed by a period of relatively few failures, and a final period towards the end of the expected life where the number of failures once again increases. This characteristic “Bathtub” curve shows that while the rate of normal product failures may be low, manufacturing defects and other inherent flaws in components will cause a certain number of systems to fail shortly after production. As time progresses and certain systems are subject to excessive wear and ageing the number of failures later in the product life-cycle will also increase (Birolini 2010).

While the "Bathtub Curve" is a good visual representation of the way in which system failure rates change over time, a probability density function is required in order to determine the probability of a system or component failing at a particular time. While there are a range of probability density functions which could be used, the Weibull distribution is particularly suited to reliability engineering applications due to the fact that it can be used to model a variety of other distribution shapes by simply changing the value of its parameters. The Weibull probability density function can be seen in Equation 2.1 and graphically in Figure 2.4. When the hazard function (Equation 2.2) of the Weibull distribution is considered it can be seen that $\beta < 1$ will represent the initial failure period from the "Bathtub Curve", while $\beta = 1$ and $\beta > 1$ respectively will represent the stable failure rate period and the end of life wear-out period. The Weibull hazard function can be seen in Figure 2.5 (Verma, Srividya & Karanki 2010).

$$f(t) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1} e^{-\left(\frac{t}{\alpha}\right)^{\beta}}, t > 0 \tag{2.1}$$

$$H(t) = \frac{\beta t^{\beta-1}}{\alpha^{\beta}} \tag{2.2}$$

2.4.3 Standards for Medical Equipment

In many if not all fields of engineering, standards play an important role in ensuring that products are designed to be safe, functional, and reliable. These product qualities are particularly important in the field of medical engineering due to the potentially life threatening consequences of poorly designed and unsafe equipment. As a result, it is a generally accepted practice in most countries that medical equipment must be certified to meet all applicable standards before it can be put into use (Backes 2007). While standards vary from country to country, the main standard that applies to the Mansell Power Lifter is IEC 60601-1 and its local variant AS/NZ 3200-1. The scope of IEC 60601-1 covers general requirements for basic safety and fundamental performance of electrical medical equipment. Due to the general nature of the standard it must in some cases be

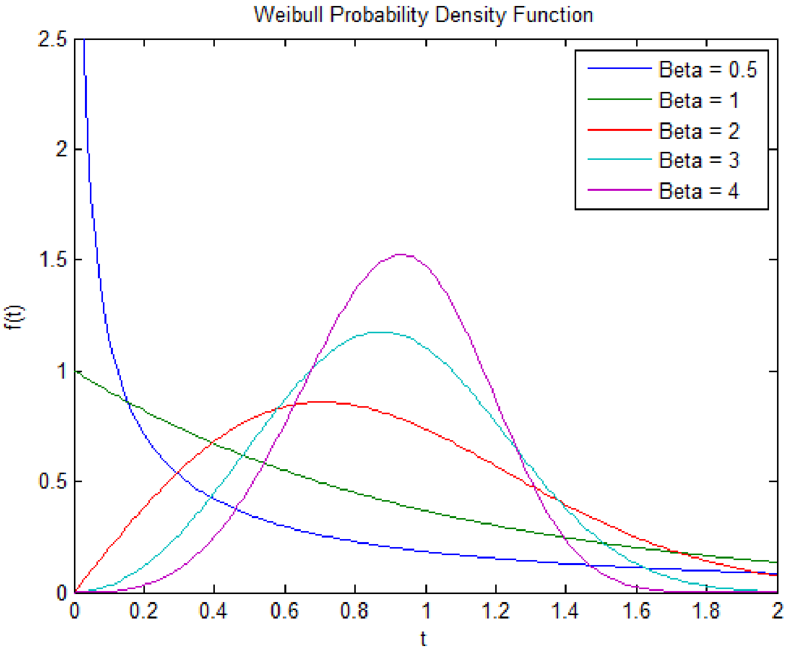


Figure 2.4: The Weibull PDF with a number of different β values representing its ability to form other PDF's (Source: author).

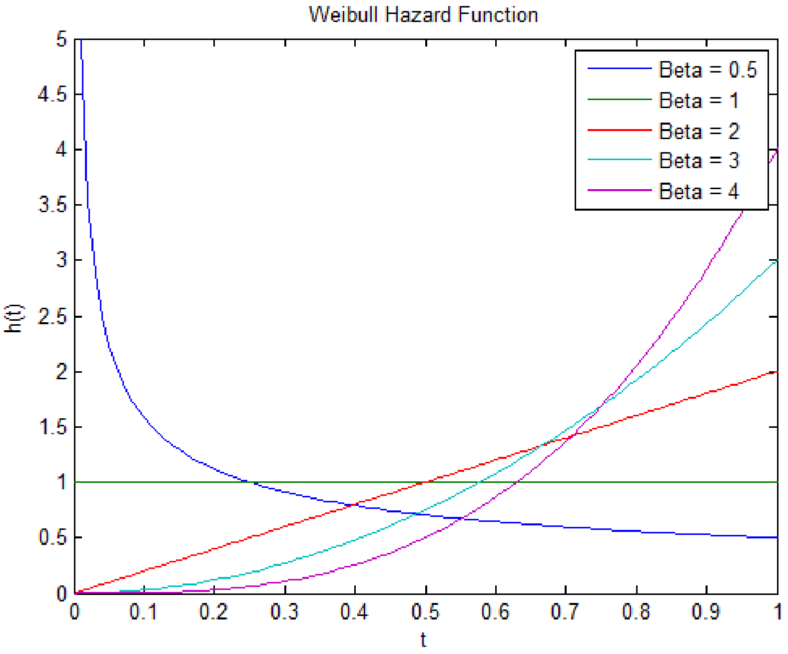


Figure 2.5: The Weibull hazard function with a number of different β values showing its ability to represent the three distinct stages of the "Bathtub" failure curve ($\beta < 1, \beta = 1, \beta > 1$) (Source: author).

supplemented by more specific standards relating to particular types of electrical medical equipment (International Electrotechnical Commission 2012).

IEC 60601-1 outlines that the basic safety and essential performance of a medical system are key factors which must exist in order for it to be considered reliable. Basic safety is defined as the system being designed in such a way that its normal operation does not harm the patient, and essential performance as the ability of the system to perform in such a way that it does not create a hazard for the patient (International Electrotechnical Commission 2012). In regards to the actuator failure scenario on the Mansell Power Lifter, it is the essential performance of the system which is being invalidated, since the inability to raise or lower the Mansell Power Lifter could result in delays or difficulties in transporting a critically ill infant, thus creating a hazard.

2.4.4 Electro-Mechanical Actuators

Electro-mechanical actuators, such as the Linak LA34 units used in the Mansell Power Lifter, are popular alternatives to traditional hydraulic actuators due to their ease of integration with electronic control systems and system power-to-weight benefits. However, as with any complex system there are a number of failure modes that they may exhibit. These failure modes can be grouped into three categories, mechanical or structural faults, motor faults, and electrical or electronic faults (Balaban, Bansal, Stoelting, Saxena, Goebel & Curran 2009).

Mechanical and structural faults can be considered to include problems with gear systems and bearings, or structural component failures caused by excess load or wear and neglect. Motor faults include loss of performance or function due to improper operating conditions or excess load. They can manifest themselves as faults such as shorts, open circuits, or shaft eccentricities. Electrical and electronic faults include component failures in power supply and control systems or wiring breakages and deterioration (Balaban et al. 2009).

In relation to the actuator failures experienced by the Mansell Power Lifter all three of these categories will need to be considered during the analysis and solution design phases. While there is no evidence of physical damage it is possible that jams or internal structural failures falling into the mechanical and structural category could be contributing to the issue. Motor faults such as open or short circuits will also need to be considered, and in the electrical and electronic category, it is possible that there are faults occurring in the limit switch electronics in the actuators themselves, or in fact with the control circuitry in the power box itself.

2.5 Design Requirements

While the recorded user reports of linear actuator failures on the Mansell Power Lifter point towards the internal limit switches as being the failure mechanism, even if this is the case it is still not clear what the failure cause is. It will therefore be necessary to perform an analysis of the operational characteristics of the linear actuators, preferably under normal operating conditions to take into account any usage factors which may contribute to the failure. The collection and analysis of this data will allow for a better understanding to be gained regarding the circumstances under which failures occur and their characteristics from a technical perspective.

Given that the reported linear actuator failures appear to occur rather infrequently, it may eventuate that it is not possible to capture data pertaining to a real actuator failure. In this case it may be necessary to analyse data from the normal usage of the Mansell Power Lifter, over a period of time, to identify any usage trends or component operation characteristics which may point towards component damage leading to failure. These findings could then be analysed to identify potential failure causes. Since the capture of real world usage data is required, and given the need to not hinder medical personnel in their use of the system, an autonomous data collection device will need to be developed to capture the desired parameters. The design considerations for this device will be considered in the following chapter.

Based on the information known about the actuator failures and the type of failures typical of electromechanical actuators, a general list of parameters which the data collection system should monitor can be determined. Actuator usage metrics such as total usage time and duty cycle will assist in determining whether or not the actuators are being operated outside of their specifications, resulting in wear, and actuator current and voltage will provide a means for determining if motor related faults are developing or if there is a problem with the actuator control system. In response to the past actuator failure information it will be beneficial to record the number of limit switch activations to assist in determining if they are indeed the failure cause, and also ambient temperature to establish if harsh thermal conditions are causing mechanical or electrical faults.

2.6 Chapter Conclusion

This chapter has provided an overview of the design and operation of the Mansell Infant Retrieval System and specifically the Mansell Power Lifter. The existing information known about the actuator failures which are being experienced was summarised and analysed to determine some of the key operational parameters which may provide evidence of the cause of failure. The relevant reliability and safety literature for medical systems and electro-mechanical actuators was reviewed, and finally the general design requirements for the data collection system were outlined. The next chapter will further explore a number of aspects of the design of this device.

Chapter 3

Design Considerations

3.1 Chapter Overview

This chapter will discuss the design considerations which were taken into account during the design phase for the autonomous data collection device. The data collection methodology to be used will be discussed and related to the design of the data collection device, any restrictions on its design due to the design and usage of the Mansell Power Lifter will be outlined, and alternative design options will be analysed.

3.2 Data Collection Methodology

3.2.1 Stress Testing

Stress testing is a technique often used in industry to help improve the reliability of products by identifying product defects or faults in advance of them being experienced by a customer in the field. This is achieved by subjecting the product under test to conditions outside of that in which they normally operate, in order to aggravate design issues or weak components, which may result in reduced performance or complete failure in real world use. Stress testing may also be used to determine an estimate of the expected life of a product in a much shorter period of time than the actual expected life, which is likely to be many years (Chan & Englert 2001).

The extent of stress testing can be considered to fall into three general categories; stress screening (Chan & Englert 2001), accelerated degradation testing, and accelerated failure testing (McLinn 2016). Stress screening is essentially a form of quality control where all or a subset of the products produced are subjected to stress outside of their normal operating conditions after production in order to ensure that there are no manufacturing defects or weak components. This is intended to identify products with obvious issues and prevent them from entering service, in order to reduce the number of initial failures that are typically experienced. This corresponds to the first section of the “Bathtub curve”. However, the limitation of stress screening is that since it is performed on a large portion of the marketable products, the tests performed must not be severe enough to cause a deterioration in performance over the product’s rated life. Conversely though this means that there is a trade-off in the number of potentially defective items which are detected (Chan & Englert 2001).

Accelerated degradation and accelerated failure testing are in contrast much less mild. In both cases the goal is to induce a degradation of the product in order to identify design inadequacies or component faults which do not manifest immediately under shorter test procedures. The primary difference between the two is that accelerated degradation testing aims to provide quicker results and shorter test durations by only looking for degraded performance and then extrapolating to draw conclusions about the reliability of the product. In contrast, accelerated failure testing waits for an actual failure to occur before conclusions are drawn. Accelerated degradation and failure testing can also be used to estimate the expected life of a product (McLinn 2016).

In the case of the Mansell Power Lifter, given that there are already a large number of systems in use in the field, stress screening techniques to assist in determining the cause of the actuator failures is out of the question. However, accelerated degradation and failure testing could be used to assist in replicating actuator failures in order for more in depth analysis to be performed. Since it would be beneficial to perform such stress tests using an automated system, these stress testing techniques could be readily integrated with the proposed data collection device.

3.2.2 Data Logging

Data logging is a useful tool for gathering in depth data on the operation of a system over a certain time period. While there are traditional measurement and data acquisition systems which operate in a laboratory style environment, and may be connected to a personal computer for direct access to data, such systems are not capable of easily gathering real world data from the operation of a system by its end users in the field, and are therefore only suitable for controlled or short term testing. Due to the limitations of such systems, an autonomous data logging device which stores data locally or transmits it though a communications channel to a remote server is more appropriate for gathering real world data in the field over long periods of time. However, due to their autonomous nature, a greater level of design effort is needed in order to ensure that autonomous data logging devices can perform the intended function and not hinder the use of the system under test in its normal fashion (Suzdalenko 2011).

Since an autonomous data logger is likely to be installed in a remote system for a period of time in order to gather data, there will be limited or no opportunities for the design or operation of the logger to be modified after deployment. It is therefore important that the design phase is thoroughly carried out and all of the design requirements are considered. While the operation and functionality of a data logging device can vary significantly based on the specific application and requirements, all autonomous data logging systems will have four main components, a power source and supply circuitry, sensors to interface with the system under test, a processor to interface with the sensors and consolidate raw data, and a data storage or transmission medium (Suzdalenko et al. 2012).

3.2.3 Testing and Data Collection Plan

Though stress testing of the Linak LA34 linear actuators used in the Mansell Power Lifter would be reasonably easy to undertake, given that Wenross Holdings Pty. Ltd. has a test Mansell Power Lifter and a number of spare actuators, it is not a sufficient alternative to real world testing. Firstly, the spare Linak LA34 units available for use are used and have an unknown age and usage history, and as such it is possible that they may have existing wear that could lead to particular fault tendencies, and therefore any results obtained may not be indicative of real world units. Additionally, there is only a minimal amount of information known about the usage patterns of the Mansell Power Lifters in use in the field, and as a result it would be difficult to accurately replicate a real world usage scenario in the stress testing environment. Therefore, the best approach to data collection at this stage is likely to be through the application of the data logger system to Mansell Power Lifters in use in the field, with stress testing used as a supplement where it is deemed necessary to test particular circumstances.

3.3 Hardware Design Considerations

Before the development of the data collection system can begin there are a number of hardware design considerations to be explored in order to ensure that the data logger meets the design requirements. These are as follows.

3.3.1 System Integration

Since the Mansell Power Lifter is part of a medical system, it is important that the safety and usability of the system is not affected in any way as a result of the field testing. Attaching a data collection device to the Mansell Power Lifter which interferes with its normal operation could result in life threatening consequences for the patient. In order to prevent such a situation from occurring it will be important to design the data logger in such a way that even if it was to fail, there would be little or no impact on the normal operation of the Mansell Power Lifter. It is planned that this requirement will be achieved through two design approaches.

The first design approach is that where possible data logger components will be placed in parallel with Mansell Power Lifter components such that if a failure was to occur in the data logger it would not prevent the operation of the entire circuit that it is attached to. The second design approach is that for any data logger components which must be placed in series with Mansell Power Lifter circuits, and could cause the system to stop working if a data logger failure occurred, the associated data logger components should be able to be easily removed to revert the system back to its default working configuration within a very short period of time. It is hoped that by taking these design approaches into account the data logger will be safe enough to be confidently used on a Mansell Power Lifter being used to transport actual patients.

3.3.2 Power Supply Considerations

Very few real world data logging devices have the luxury of access to a continuously available mains power source. Therefore, much thought needs to be put into the design of the power supply and the overall power consumption of the device to ensure that it is capable of operating in the desired application. Often this will involve the use of a battery to allow for completely autonomous operation and may be augmented by infrequent access to external power for charging or some form of renewable energy such as solar power. As a result of this limited power capacity, it is important to ensure that the power consumption of the data logger is minimised (Suzdalenko 2011).

Since the proposed data logger for the Mansell Power Lifter only needs to be powered when the linear actuators are being operated, it is possible that the data logger could be powered from the linear actuator power cables themselves, thus eliminating the need for a primary battery contained within the data logger or a direct connection to the Mansell Power Lifter main batteries. This will be possible, and will not have an impact on the operation or performance of the actuators, since the current consumed by the data logger is likely to be in the order of tens of milliamps (Suzdalenko 2011) compared to the multiple amps that the Linak LA34 actuators typically require (Linak 2015). The plan to power the data logger directly off of the Mansell Power Lifter actuator power cables fits within the design approach of connecting data logger components in parallel, since power can be tapped off each of the actuator power wires, and a fuse can be placed in series with the data logger to prevent short circuit conditions across the actuator power cables.

It should be noted that because the actuators are driven directly by the Mansell Power Lifter power box, the voltage on and current in the actuator power cables could be reversed depending on the direction that the actuator is being driven. As a result, it will be necessary to design a power supply for the data logger which is capable of rectifying the input voltages which could be of any polarity. In addition to the voltage rectification, the data logger power supply will need to contain voltage regulation in order to drop the operating voltage of the linear actuators, which could be either 12 or 24 volts depending on the operating conditions (Grant-Thomson 2014), and reduce it to the 5 or 3.3 volts required for the microcontroller and sensor systems.

3.3.3 Sensors and Data Acquisition

Sensors and data gathering circuits for data logging devices can be varied and will depend on the specific goals of the data logger which is being designed. Since the data logger for the Mansell Power Lifter is concerned with monitoring linear actuators containing DC motors, it can be assumed that the logger will at least contain voltage and current sensing circuitry in order to monitor the operational parameters of the motors (Suzdalenko et al. 2012). Voltage can usually be measured through the analogue to digital conversion functionality of modern microcontrollers with external resistor divider networks in order to translate the voltage being measured into the correct range for the converter (Holoubek 2013). Current can be measured through several techniques including measuring the voltage across a series resistor, current transformers, and Hall Effect devices. However, all of these techniques still require an analogue-to-digital voltage conversion to be performed, either in the current sensor module itself or via a microcontroller (Yarborough 2015).

While there are several different methods for measuring current it should be noted that different current sensing techniques have different levels of invasiveness on the circuit in question. For current measurement through a series resistor, the very act of measuring the current requires that the current being measured flow through a resistor, causing a voltage drop and therefore power dissipation. This may not be an appropriate technique if the resulting voltage drop could affect the operation of the system. Additionally, if very high currents are being measured then very low resistance values will be required in order to minimise the voltage drop and power dissipation in the resistor (Yarborough 2015).

Magnetic current measurement techniques on the other hand are considered to be non-invasive since the measurement component of the sensor has no direct electrical connection to the circuit under test. This is possible due to the magnetic fields created when current flows through a wire. By coupling this magnetic field into the detection circuit it is possible to detect the current flow. Current transformers achieve a voltage output by passing the current generated by the magnetic field in a secondary coil through a resistor. Hall Effect devices on the other hand use the difference between a local magnetic field and the perpendicular magnetic field created by the current in the wire to generate a voltage, this is known as the Hall Effect. Hall Effect devices have the advantage over current transformers of consuming less power due to not dissipating power in a resistor (Yarborough 2015).

Either a current transformer or Hall Effect current sensor circuit would be suitable for implementation in the data logger for the Mansell Power Lifter due to their non-invasive current sensing ability, which lines up with the design objective of minimising direct interference with system circuitry. However, a Hall Effect sensor has been chosen over a current transformer for this application, since analysis of the available sensor modules showed that Hall Effect sensors were more readily available in small integrated packages, which will help to keep the overall size of the data logger down. Specifically, a current sensor module containing the ASC712-30A current sensor integrated circuit from Allegro MicroSystems LLC was chosen to be used in the data logger. It has a current measurement range of ± 30A (Allegro Microsystems LLC 2012) which should be sufficient for the maximum current through the actuators of approximately 25A when the system is fully loaded (Keys 2014), and can withstand any motor current transients generated thanks to a transient tolerance of 100A for 100ms (Allegro Microsystems LLC 2012).

While the voltage and current measurement will make up the main sensor components required, it will also be necessary to have some form of temperature sensor to assist in determining if the temperature of the actuators is contributing to the failures, as theorised in one of the actuator failure occurrences. Ideally, temperature measurement would be performed by placing a temperature sensor on a point of interest on the actuator, such as the motor or internal circuit board. However, since it is a design objective that the data logger be as non-invasive to the Mansell Power Lifter as possible, and should be easily removed if problems are experienced, placing a temperature sensor inside the actuators will not be feasible.

It was decided to use a single temperature sensor to measure the ambient temperature at the data logger itself. While potentially not as useful as a localised temperature sensor, the ambient temperature could be used to identify any trends which could be contributing to the actuator failures. In particular a DS1631 integrated temperature sensor with I2C communications interface was chosen to be used. A number of DS1631's are already in use in the Mansell Neocot and are therefore readily available and known to work reliably.

3.3.4 Microcontroller Selection and Considerations

The next component that needs to be considered is the system processor. The system processor is responsible for the task of retrieving the data measured by the various sensors, performing any calculations that are required to get the data into a usable state, and then recording the data to a storage medium or transmitting it to a remote server. The requirements for a system processor can vary greatly based on the requirements of the data logger, however the most important considerations are the number of sensor inputs (generally analogue to digital channels), the variety and number of supported communications protocols, power consumption, and the ease of implementation (Suzdalenko et al. 2012).

While there are a large number of low power microcontroller processors on the market which contain numerous analogue inputs and various communications protocols, one important consideration which can set them apart is the ease of implementation. The ease of implementation can be considered both in terms of physical hardware implementation and software development. The hardware implementation of a microcontroller is usually not as simple as just purchasing a single integrated circuit chip. Most microcontrollers on the market today will require a number of external components in order to properly function, such as crystal oscillator clock circuits, low voltage power supply circuitry, and programming interfaces. These additional requirements will need to be considered when choosing a microcontroller platform.

The software development environment for potential microcontrollers will also need to be considered as different products may require the use of different programming languages and have varying levels of pre-existing software library support. One such microcontroller platform which is becoming increasingly popular due to its ability to support rapid prototype development is the Arduino family of microcontroller development boards, based on Atmel ATMega microcontrollers. These microcontroller development boards provide an integrated microcontroller solution with all of the required external components on board and are available in various sizes. They also come with an integrated software development environment based on the C/C++ languages (Arduino 2016*a*) with a large number of libraries available for accelerated software development (Smith & Miller 2013).

In order to ensure that the data logger is able to spend sufficient time in the field to collect a useable data set, it is important to prevent the development process from taking too long. If a microcontroller solution can be obtained which does not require extensive hardware development in order to be implemented it will be possible to reduce the amount of development time required. While there are many other microcontroller products on the market, very few of them can offer small sized development boards which are ready to use out of the box, while at the same time providing free and easy access to a large range of software libraries.

Due to the ease of development, both in terms of hardware and software, and the fact that a number of communication software libraries will be required to interface with the planned hardware, it was decided to use an Arduino Nano as the microcontroller platform for the data logger. The Arduino Nano features the Atmel ATMega328P microcontroller running at 16MHz with 14 digital I/O pins, 8 analogue inputs, 32KB or flash memory, 2KB of RAM, and 1KB or EEPROM. The Arduino Nano board contains all of the required microcontroller support circuitry and a USB interface for programming all within a small 45mm by 18mm form factor (Arduino 2016*a*). This small size will assist in keeping the overall size of the data logger low.

3.3.5 Data Storage and Accessibility

Data storage is the final key component of a data logging device. It is necessary that a data storage solution be implemented so that all of the data gathered by the sensors can be recorded in a defined manner for later analysis. It is also important that the data can be easily retrieved at the end of the logging period, and that it is recorded in such a way that it is easy to manipulate and analyse once retrieved. While in some applications it is desirable that the data gathered by the data logger be transmitted in real-time to a remote server for immediate analysis (Smith & Miller 2013), in the case of the Mansell

Power Lifter, immediate retrieval of data is not of critical importance and will not be pursued due to the additional cost and complexity involved.

The alternative to remote data transmission is to store data locally in the data logging device. However, depending on the period over which the data logger is expected to be autonomous, and the frequency and quantity of data sampling, the required storage space may be very large. Also, since typical microcontrollers suitable for this application only have up to several kilo-bytes of on board non-volatile memory, an external data storage solution will be necessary (Suzdalenko et al. 2012).

Secure Digital or SD cards are one such, flash memory based, data storage solution which is popular for use in data logging devices. These memory cards are widely available and are typically used in digital cameras among other things. SD cards have the advantages of small physical size with standard and micro form factors, large data capacities of tens to low hundreds of gigabytes, are widely available, and use the Serial Peripheral Interface (SPI) bus for data communications (Ibrahim 2010). The SPI bus is available on the majority of microcontroller platforms including the Atmel ATMega328P used in the Arduino Nano (Arduino 2016*a*). SD cards can also be easily connected to a personal computer to extract data files located on them (Ibrahim 2010).

For data logging applications, values recorded to file-based flash memory such as SD cards are typically written in Comma Separated Value (CSV) format where each data field is separated by a comma character. This format is widely used and allows for data to be quickly imported into software such as Microsoft Excel or MATLAB for analysis (Suzdalenko et al. 2012). In order to keep the overall size of the data logger small it was decided that a micro-SD card and associated microcontroller interface hardware would be used to record data from the various sensors.

While writing to the micro-SD card is a suitable technique for storing the large amounts of real time data generated by the data logger's sensors, for some parameters such as the total elapsed operational time and limit switch activation counter, which need to be read back and updated regularly, extracting their previous value from the micro-SD card may prove unnecessarily cumbersome. For these variables, in addition to storing them on the micro-SD card, it will be beneficial to keep an up to date copy in the non-volatile EEPROM (Electrically Erasable Programmable Read Only Memory) of the ATMega328P microcontroller, where they can be easily accessed and updated as necessary.

3.4 Software Design Considerations

In addition to the hardware design considerations, there will also be a number of software design considerations to be made. These are outlined as follows.

3.4.1 Software Reliability

Real-time systems are often thought of as computer controlled systems which must respond immediately to ensure the safety of the process that they are controlling. However, Laplante (2004) defines a real-time system as a system which must be able to perform not only the correct operation but also execute it within a timely manner, such that the system specification is not breached. Real-time systems may be further classified as soft, firm, or hard depending on the consequences and severity of a failure to meet the time requirements. Soft real-time systems have minimal consequences for failure and hard real-time systems have catastrophic consequences for failure (Laplante 2004).

In the case of the Mansell Power Lifter data logger, there are several reasons for considering it to be a real-time system. Firstly, it is important for the reliability of the data obtained that sensor and time measurements are performed and processed at regular intervals. For example, in order to record an accurate representation of the system parameters at a particular point in time it is important that all sensor readings be performed within the shortest timeframe possible.

Another factor that testifies to the data logger being a real-time system is the need to detect a poweroff condition and write any remaining data to the micro-SD card before the microcontroller loses power. Since the data logger is powered solely from the linear actuator power cables, when the user lets go of the actuator drive buttons on the control panel and the actuator is switched off, there is only a very short period of time until the capacitors in the data logger power supply are drained and the microcontroller turns off. The microcontroller must be able to detect this condition as quickly as possible and write out any remaining data to the micro-SD card immediately to prevent data loss.

Since the failure of the data logger to meet its time requirements will at worst result in a partial loss of sensor data, and not have any impact on the operation of the Mansell Power Lifter, it can be considered to be a soft real-time system. While a failure to perform within the necessary time requirements would not result in a critical failure for either the data logger or the Mansell Power Lifter, it will be important to design the software of the data logger in such a way that the risk of a time related failure occurring is minimal. This can be achieved through careful specification analysis and software structure design along with the potential application of real-time system design techniques. In addition to the real-time system reliability requirements it will be necessary to perform an appropriate amount of in depth testing of the data logger software to ensure that there are no errors in its operation.

3.4.2 Program Structure

As with many embedded system designs the Mansell Power Lifter data logger software will take the form of a finite state machine. Finite State Machines are systems which can be thought of as operating in one of a number of defined execution states at any particular time. The system will have an initial starting state and one or more potential terminal states. Additionally, for each state there will be particular events which will trigger a transition to a different state of operation (Laplante 2004).

Finite State Machines can be classified into two main types, Moore machines and Mealy machines. The difference between the two is that Mealy machines can define outputs associated with the transition between states while Moore machines can only have outputs associated with machine states themselves (Laplante 2004). Finite State Machines can be represented mathematically as a five-tuple for Moore machines and a six-tuple for Mealy machines as seen in Equations 3.1 and 3.2 respectively. Where S is a set of all possible states, i is the initial state, T is the set of terminal states, Σ is an alphabet of symbols which mark the state transitions, Γ is the set of outputs, and δ is the transition function for determining the next state (Laplante 2004).

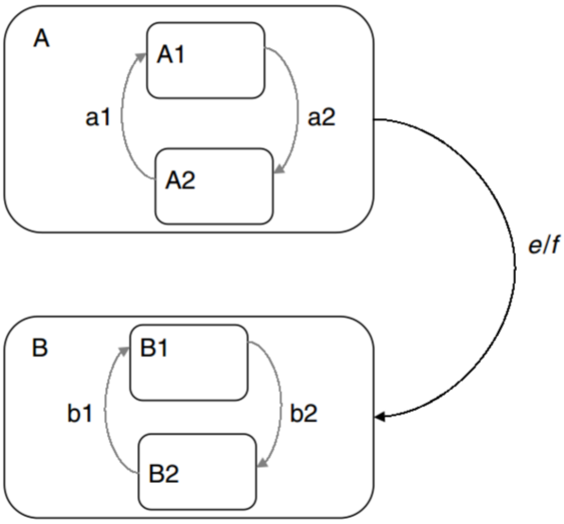


Figure 3.1: An example of a Finite State Machine statechart showing nested states and state transitions (Laplante 2004).

$$M = \{S, i, T, \Sigma, \delta\} \tag{3.1}$$

$$M = \{S, i, T, \Sigma, \Gamma, \delta\} \tag{3.2}$$

One of the advantages of thinking about the software design of a system as a Finite State Machine is that statecharts can be used to easily represent the operation of the system and its movement between various operating states. Statecharts can also be used to represent common software design features such as nesting and multi-tasking. An example of a Finite State Machine statechart showing state nesting can be seen in Figure 3.1 (Laplante 2004).

While initially the data logger may seem to only have one normal operational state, that of obtaining sensor data and recording it at regular intervals, upon further analysis it can be seen that there are also a number of states relating to the start up of the device and the pre-shutdown actions. It can also be seen that within the primary operational state there will be a number of sub-states for each actuator depending on whether or not it is running and if it has hit one of its limits. It is therefore appropriate to consider the software design of the data logger as a Finite State Machine and use the associated techniques in the design process.

Table 3.1: List of parameters to be recorded by the data collection device.

Full Log File	**Summary Log File**
Front actuator current	Average front actuator current
Front actuator voltage	Average front actuator voltage
Front actuator current run time	Front actuator total run time for current operation
Front actuator total elapsed time	Front actuator total elapsed time
Front actuator total limit switch activations	Front actuator total limit switch activations
Rear actuator current	Average rear actuator current
Rear actuator voltage	Average rear actuator voltage
Rear actuator current run time	Rear actuator total run time for current operation
Rear actuator total elapsed time	Rear actuator total elapsed time
Rear actuator total limit switch activations	Rear actuator total limit switch activations
Ambient temperature	Ambient temperature
Time and date that current operation was initiated	Time and date that current operation was initiated

3.5 Chapter Conclusion

This chapter has discussed the design requirements for the construction of a data collection system for the Mansell Power Lifter which will take the form of an autonomous data logging device. Specifically, a number of hardware considerations relating to data collection, processing, and storage were covered as well as the necessary implementation considerations relating to these functions. The relevant design concepts for the software side of the data logger were also considered.

After working through all of the design considerations in this chapter and the requirements outlined in Chapter 2, the following list of parameters have been identified for the data logger to monitor and record as can be seen in Table 3.1. Additionally, it was decided that as well as the full record of all data gathered by the data logger, a second summary log file would be created to provide an overview of the operating parameters of the Mansell Power Lifter for each operation. The motivation behind the summary log file is to provide the ability to get a good overview of the operation of the system quickly without having to perform time consuming processing and analysis of the detailed log data.

Chapter 4

Data Collection System Hardware Design

4.1 Chapter Overview

This chapter will discuss the design and development of the hardware aspects of the data collection system. The design considerations explored in the previous chapter will be applied to the final design of each of the aspects of the device. Each of the four main data logger component categories will be covered, power supply, sensors and data sources, the microcontroller, and data storage, as well as a discussion on the physical construction of the data logger. A block diagram of the data logger hardware can be seen in Figure 4.1 and a full circuit schematic can be seen in Appendix B. The software design of the microcontroller Arduino code will be discussed in Chapter 5.

4.2 Power Supply

As outlined in the design considerations, the data logger for the Mansell Power Lifter will be powered entirely from the linear actuator power cables. However, the data logger circuitry cannot be directly connected to the actuator cables since the Arduino Nano and various sensors require a supply voltage of 5V and the voltage supplied to the linear actuators can be either 12V or 24V depending on the mode of operation. Additionally,

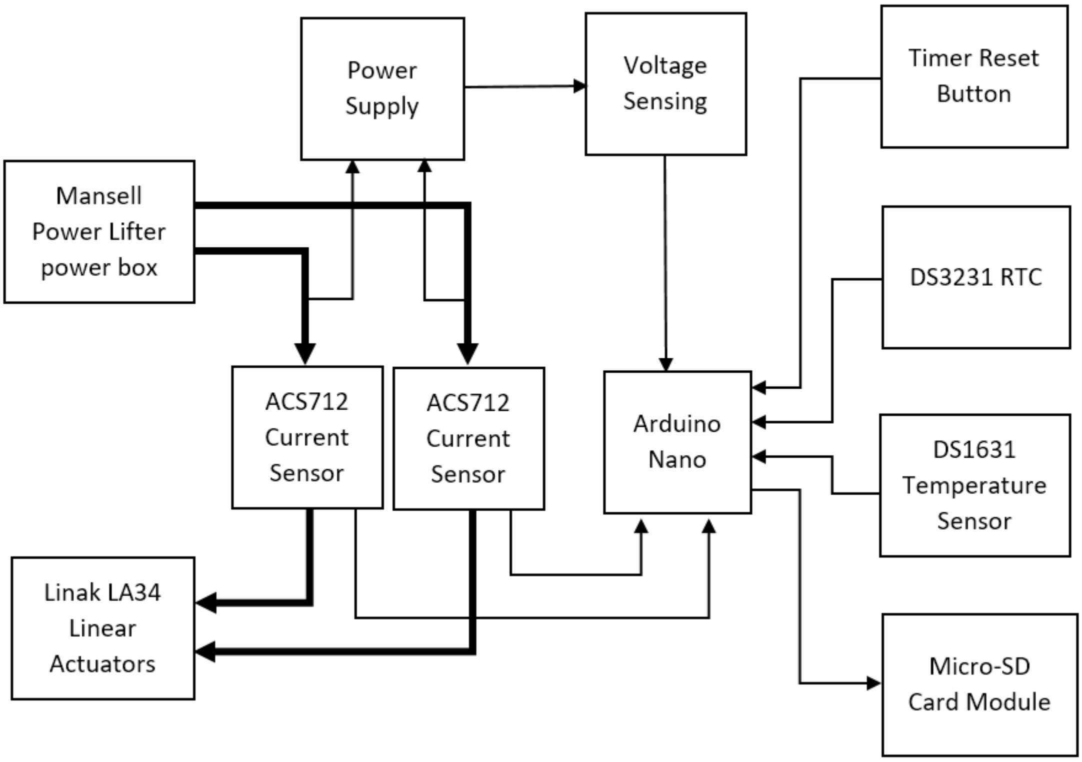


Figure 4.1: Block diagram of the Mansell Power Lifter data logger showing major system components and connections. (Source: author)

since the linear actuators are directly driven by the Mansell Power Lifter power box, the voltage on the actuator power cables could be either polarity depending on the direction the actuators are being driven (Keys 2014). Finally, since it is desired that the data logger monitor the operation of both of the linear actuators it will be necessary to obtain power from either or both of the linear actuator's power cables depending on whether the front actuator, rear actuator, or both actuators are running. It will therefore be necessary for a power supply to be designed which can meet these requirements.

The first power supply requirement to be designed is the ability for the data logger to be powered from 12V or 24V with variable polarity. In order to achieve this it was decided that a full-wave bridge rectifier would be implemented. Full-wave bridge rectifiers are typically used in AC power supply designs to convert the sinusoidal input voltage about zero into an AC waveform with only positive voltages (Davide Giacomini 2008). A full-wave bridge rectifier can be implemented using four discrete diodes, however they are also readily available in compact integrated circuit form. For the Mansell Power Lifter data logger a number of KBP205G integrated full-wave bridge rectifiers were salvaged from the mains AC circuitry of a number of old battery chargers which were no longer in use. While these rectifiers are significantly over-rated for this application with mains voltage tolerance and a maximum current of 2A (Taiwan Semiconductor n.d.) they were readily available and have a compact size which will assist in keeping the footprint of the power supply small. While there will be a small voltage drop across the diodes in the bridge rectifier this should not have any effect on the operation of the power supply since the input voltage is significantly higher than the required output voltage.

In order to allow for the data logger to be powered from either the front or rear linear actuator power cables or both, depending on how the Mansell Power Lifter is being operated, it was decided that two of the full-wave bridge rectifiers chosen would be connected to each of the actuator power cables and their outputs would be connected together. If one actuator was running but not the other, then it would not be possible for power to flow back into the non-powered actuator cables, due to the diodes in the bridge rectifier being reverse-biased. Additionally, there would be no risk of different voltages being applied to each of the actuator power cables, and therefore producing a short circuit, since when the Mansell Power Lifter power box switches between 12V and 24V mode it switches both actuators simultaneously. To provide additional protection for the Mansell Power Lifter against faults in the data logger power supply, fuses were placed in series with each input to the power supply. If a short circuit condition within the power supply were to occur the corresponding fuse would blow and ensure that the actuators could continue to function normally.

The final power supply design requirement to be considered is the need to reduce the 12V or 24V supplied to the linear actuators down to the 5V which is required for the Arduino Nano and the various sensor modules. There are two common voltage regulation techniques used to reduce high supply voltages down to low digital logic voltages, linear regulators and switching regulators. The operation of these voltage regulation circuits will not be discussed here in detail. Linear voltage regulator circuits require very few components but can dissipate a large amount of power ($P_{Loss} = (V_{IN} - V_{OUT}) \times I$) and are therefore quite inefficient. As a result linear voltage regulators used in high power applications will require significant heatsinking. In comparison, switching voltage regulators require a number of additional components but have the advantage of being very efficient due to their discrete switching nature of operation. (Zhang 2013) Due to the low power requirements of the data logger and simplicity of design it was decided to use a linear voltage regulator in the power supply.

Specifically the LM7805 was chosen due to its wide availabilty and wide input voltage range of up to 35V (Fairchild Semiconductor 2014*b*). Aside from the main voltage regulator integrated circuit, a linear voltage regulator also requires a bypass capacitor across both the input and output (Zhang 2013). Typical values of these capacitors are 0.33μF for the input and 0.1μF for the output (Fairchild Semiconductor 2014*b*). However, it was identified that for the application of the data logger it would be necessary for the Arduino Nano microcontroller to remain powered on for a short time after the power was removed from the linear actuator power cables. This is necessary in order to ensure that any remaining data can be written to the micro-SD card successfully before the data logger turns off. Since the additional run time was only likely to be very short it was determined that this requirement could be achieved by increasing the values of the input and output capacitors on the voltage regulator.

While both the input and output capacitors were increased in size it was determined that increasing the size of the output capacitor significantly would not be appropriate since this would result in a significantly increased voltage rise time at power on. Instead it was decided to significantly increase the size of the input capacitor since the higher 12V or 24V applied to it would minimise the effects of the capacitor charge time at power on. An additional advantage of increasing the size of the input capacitor instead of the output capacitor is that voltage regulation will be retained for a significant time period instead of the immediate voltage drop initiation which would result from only increasing the output capacitor size.

In order to determine appropriate values for these capacitors a test scenario was set up where, upon detecting a power-off condition, the Arduino Nano would pull one of its digital outputs to ground, before performing the necessary power-off operations and writing any remaining data to the micro-SD card. Once all power-off operations were performed the Arduino would once again output a high on the digital output. By monitoring the data logger input voltage and the digital output of the Arduino using an oscilloscope it was possible to determine if the power supply capacitor configuration was suitable by observing whether or not the Arduino had time to complete its power-off operations and output a digital high before power was lost.

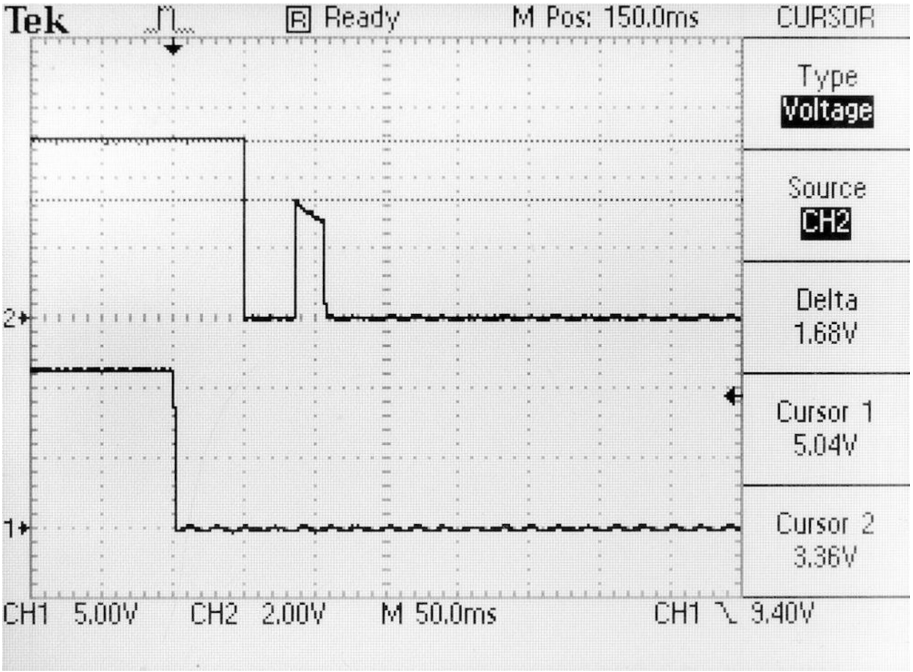


Figure 4.2: Mansell Power Lifter data logger power supply test for performing power-off operations with 12V input voltage. (Photo: author)

To perform this test, channel one of the oscilloscope was connected to the actuator power cables to measure the input voltage and channel two was connected to digital pin four of the Arduino. The ground wires for both channels were connected to the data logger's circuit ground. The oscilloscope was set to trigger on the falling edge of channel one when the power was disconnected. Screen captures of the oscilloscope while performing these tests can be seen in Figures 4.2 and 4.3 for 12V and 24V input voltage respectively. It can be seen that in both scenarios the Arduino Nano is able to complete its power-off tasks before the input voltage drops sufficiently low that it can no longer operate. After testing with various capacitor values for the voltage regulator input and output capacitors, it was determined that an input capacitor of 1000μF and an output capacitor of 10μF would be appropriate and provide enough power for the Arduino to perform its power-off operations.

The final schematic for the data logger power supply can be seen in Figure 4.4 and the full schematic for the entire data logger can be seen in Appendix B.

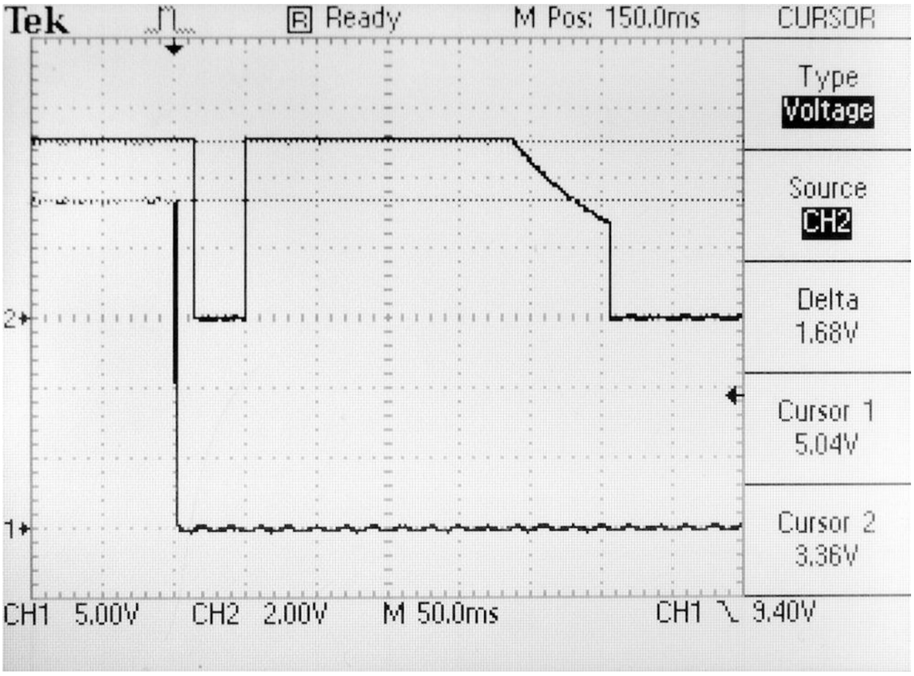


Figure 4.3: Mansell Power Lifter data logger power supply test for performing power-off operations with 24V input voltage. (Photo: author)

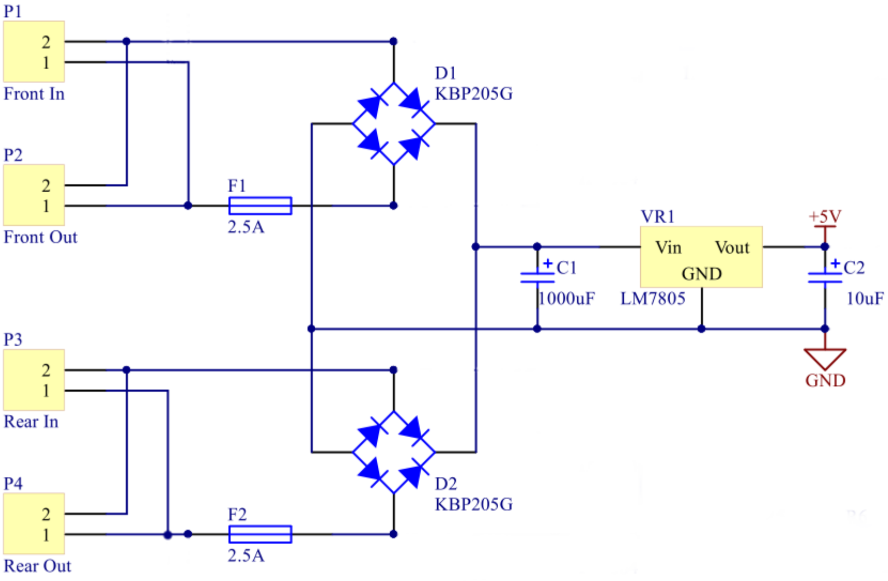


Figure 4.4: Mansell Power Lifter data logger power supply schematic. (Source: author)

4.3 Sensors and Data Sources

4.3.1 Voltage Sensing

Since the operating voltage of the actuators has been identified as an important parameter for the data logger to measure, it will be necessary to implement some form of voltage sensing. While the ATMega328P in the Arduino Nano has ten analogue voltage inputs it will not be possible to measure the voltage on the linear actuator cables directly since they are operating at a much higher voltage than the 5V used by the microcontroller. Not only would the ATMega328P not be able to measure voltages this high it would likely be severely damaged. It was decided that the high voltages on the linear actuator power cables could be converted to an appropriate voltage level for the microcontroller by implementing a resistor divider network. This technique is commonly used to scale down large voltages to a readable level, with the value read by the analogue-to-digital converter being scaled up in software using the calculated ratio of the resistor divider network (Holoubek 2013).

However, the resistor divider network is not the only consideration which needs to be made in the case of the data logger. Since the polarity of the voltage on the actuator cables could be reversed depending on which direction it is being driven, simply connecting a resistor divider network to the actuator cables will only work when the voltage is in a certain polarity. If the actuator is driven in one direction the voltage will be able to be measured, but if it is driven the opposite direction the resistor divider will be connected to ground. One way around this problem would be to implement four resistor divider networks, one for each actuator power cable wire for both actuators. However, this would require twice the number of microcontroller inputs and twice as many resistors.

It was identified that one way to overcome this limitation was to place two diodes in series with the voltage measurement line for each wire of the actuator power cable, essentially forming a half-wave bridge rectifier. This would allow current to flow from whichever wire was positive at any point in time through the resistor divider, but at the same time also block it from flowing directly back into the negative wire, due to the other diode being reverse-biased. Unlike a full-wave bridge rectifier a half-wave bridge rectifier only allows positive parts of the input waveform to appear on the output (National

Instruments 2013). These two diodes are placed before the first resistor in the resistor divider network and will result in a small voltage drop which must be accounted for in order to obtain accurate voltage measurement. The half-wave bridge rectifier was implemented using standard 1N4004 diodes since they were readily available.

The resistor values of the resistor divider network were calculated based on a maximum possible actuator cable voltage of 25V representing a full scale 5V input to the microcontroller analogue-to-digital converter. A reduction in voltage from 25V to 5V represents a division ratio of 5 for the resistor divider network. In order to prevent excessive current from being wasted in the resistor divider networks it was decided to keep the overall impedance at around 1MΩ. Thus in order to obtain a voltage division ratio of 5 it was decided to use a 1MΩ resistor followed by two 100KΩ resistors in series. The input to the microcontroller analogue-to-digital converter was taken from the point between the 1MΩ resistor and the two 100KΩ resistors.

While the resistor divider network designed would have approximately a uniform division factor of 5, the effects of the half-wave bridge rectifier diodes placed in series need to be considered. The 1N4004 is specified to have a 0.6V forward voltage drop at 10mA (Fairchild Semiconductor 2014*a*), though given the very high impedance of the resistor divider network the actual current is likely to be much less. Additionally, since diodes are non-linear in their forward voltage behaviour, as their forward current changes it is likely that the voltage across the diode will be variable depending on the input voltage. In order to test the linearity of the voltage dividers with the rectifier diodes, various input voltages were fed into the data logger power supply, and the output of the resistor divider was measured and recorded. The results were imported into MATLAB and both a first and second order polynomial fit was performed and compared to the original data. A graph showing the results of the test can be seen in Figure 4.5.

As expected, the addition of the bridge rectifier diodes in series with the resistor divider network resulted in the division factor of the resistor divider network being increased. The resistor divider network testing also shows that the non-linearity of the diodes used in the bridge rectifier does have a noticeable effect on the division factor over the input voltage range. Additionally, it can be seen that the second order polynomial fit results in a very good match to the actual measured data while the first order polynomial fit is a less optimal fit. While it would be reasonable to simply find the average resistor

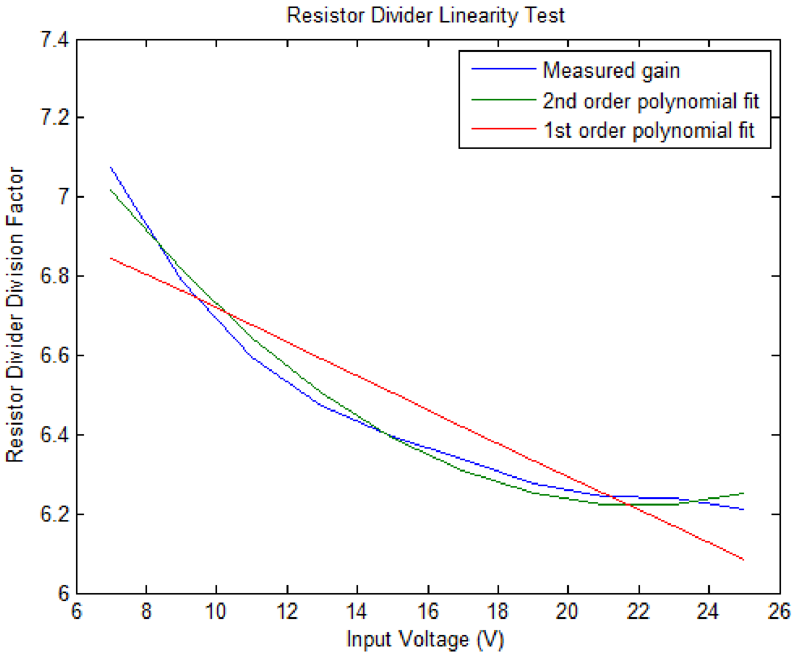


Figure 4.5: Mansell Power Lifter data logger input voltage resistor divider network division factor linearity testing. (Source: author)

divider network gain and use it as a constant across the entire voltage input range, it was decided that it would not be computationally unreasonable to implement the second order polynomial, in order to achieve the most accurate voltage measurement results. The results from the polynomial fit performed in MATLAB showed that the coefficients of the division factor polynomial would be as shown in Equation 4.1.

$$V_{IN} = 0.1155V_{Measured}^2 - 0.8175V_{Measured} + 7.6267 \tag{4.1}$$

The final schematic for one of the data logger voltage sensing circuits can be seen in Figure 4.6 and the full schematic for the entire data logger can be seen in Appendix B.

4.3.2 Current Sensing

As noted in the design considerations in Chapter 3 the data logger will use two ACS712-30A Hall Effect current sensors to measure the current being supplied to each actuator. The ACS712-30A is capable of meeting the electrical requirements of measuring up to the maximum actuator current of approximately 25A under full load (Keys 2014) in either polarity, and being able to withstand transients caused by the DC motors of up to 100A for

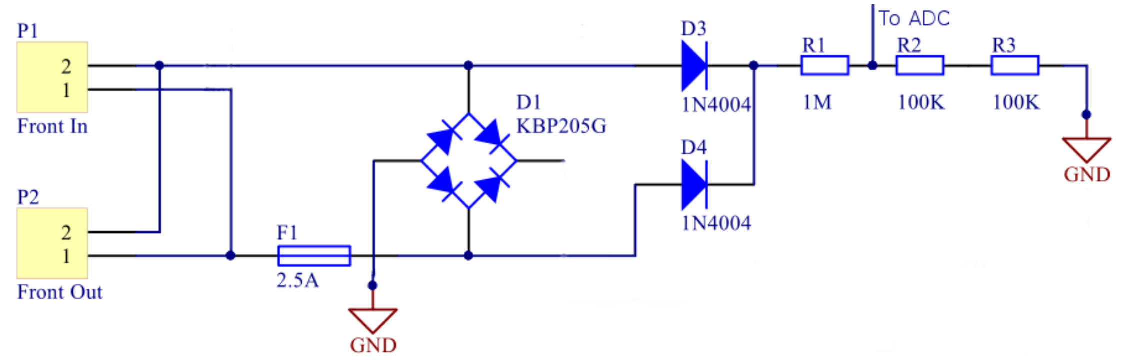


Figure 4.6: Mansell Power Lifter data logger input voltage resistor divider network schematic. (Source: author)

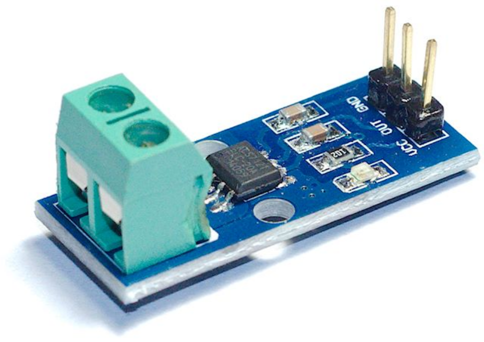


Figure 4.7: An Allegro ACS712-30A Hall Effect current sensor module as used in the data logger. (Photo: author)

100ms (Allegro Microsystems LLC 2012). One additional benefit of using the ACS712-30A is that it is available from a number of sources as a low cost pre-made module on a printed circuit board, with screw terminals for easily connecting to the current source being measured. This will reduce the development time required to implement the ACS712-30A significantly since it is only available as a surface mount component which would otherwise require a custom printed circuit board to be manufactured. A picture of the ACS712-30A current sensor module can be seen in Figure 4.7.

The ACS712-30A current sensor modules will be placed in series with one of the linear actuator power wires in each linear actuator power cable. Since current could flow in either direction, and in different directions for each actuator at any particular time, care will need to be taken to ensure that the current carrying cables are connected in a consistent manner across both actuators. In other words if both actuators are being driven in the same direction at the same time the output of their corresponding ACS712-30A current sensors should be of the same polarity to ensure that the results can be interpreted correctly. The output of each ACS712-30A current sensor will be an analogue voltage from 0V to 5V corresponding to a current range of -30A to +30A with 0A being represented by $\frac{1}{2}V_{CC}$ or approximately 2.5V (Allegro Microsystems LLC 2012). Each ACS712-30A voltage output will be connected to an analogue-to-digital converter input on the Arduino Nano's ATMega328P.

4.3.3 Temperature Sensing

As discussed in the design considerations in Chapter 3 the Mansell Power Lifter data logger will feature a single temperature sensor located within the data logger itself to monitor the ambient temperature that the system is operating in. The DS1631 temperature sensor from Maxim Integrated was chosen over a simpler thermistor temperature sensing arrangement due to the higher precision and reliability of an integrated solution straight out of the box. The DS1631 has a measurement range of -55°C to 125°C with an accuracy of ±0.5°C from 0°C to 70°C. It has a configurable conversion resolution of 9 to 12 bits and communication is performed over the I2C bus. Since the temperature conversion time for the DS1631 increases as the number of bits of resolution increases (Maxim Integrated 2015*a*) it was decided to use the lowest resolution of 9 bits to ensure that the ambient temperature could begin being logged as soon as possible after the data logger was powered on. With 9 bits of resolution the maximum temperature conversion time is 93ms and the temperature output will have a step size of 0.5°C (Maxim Integrated 2015*a*).

The DS1631 is already in use by Wenross Holdings Pty. Ltd. for temperature measurement in the Mansell Neocot and it was therefore readily available for integration into the data logger. The DS1631's in use in the Mansell Neocot are each soldered onto small printed circuit boards. These temperature sensor modules were transferred unchanged to the data logger design. A picture of the DS1631 temperature sensor modules used can be seen in Figure 4.8.

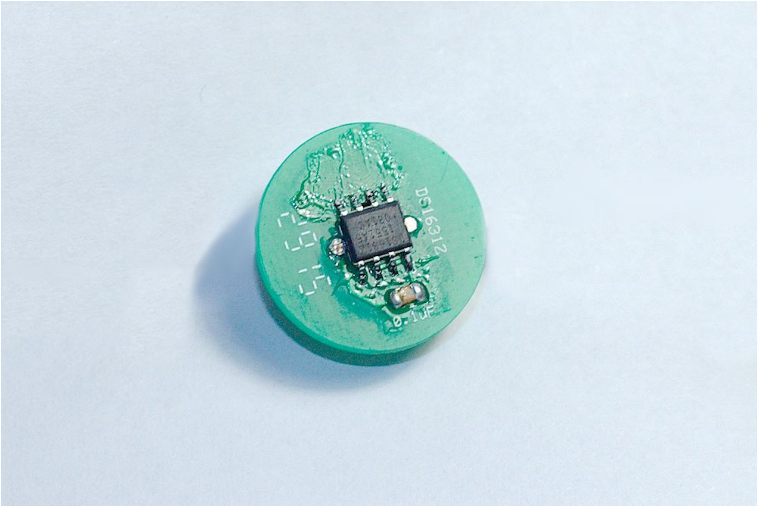


Figure 4.8: A temperature sensor module produced by Wenross Holdings Pty. Ltd. for the Mansell Neocot containing a Maxim Integrated DS1631 thermometer integrated circuit. (Photo: author)

4.3.4 Real Time Clock

While the duration of each operation of the Mansell Power Lifter can be easily determined using the internal timers of the microcontroller, in order to find out how frequently it is used it will also be necessary to work out the idle time between operations. Since the data logger will be powered off between operations of the system it will not be possible to use the microcontroller itself to measure this time. It was determined that an appropriate solution to this problem was to implement a real time clock module. Real time clocks are precision timer circuits which are able to keep track of time and output it in a human readable manner with hours, minutes, and seconds, and in many cases also keep track of the current date as well. They generally feature a dedicated oscillator of some kind and importantly have the provision for a battery backup source to keep the correct time even when the main system is powered off. (Bowman 2013) By recording a timestamp along with each data point written to the micro-SD card it will be possible to determine when each and every operation occurred and the time between each operation.

For the Mansell Power Lifter data logger it was decided to use a DS3231 real time clock module from Maxim Integrated. The DS3231 features an integrated temperature compensated crystal oscillator that enables it to be accurate to within ±2 minutes per year and features full calendar datekeeping. It features the provision for a 3.3V backup battery and communication is performed over the I2C bus (Maxim Integrated 2015*b*) which will

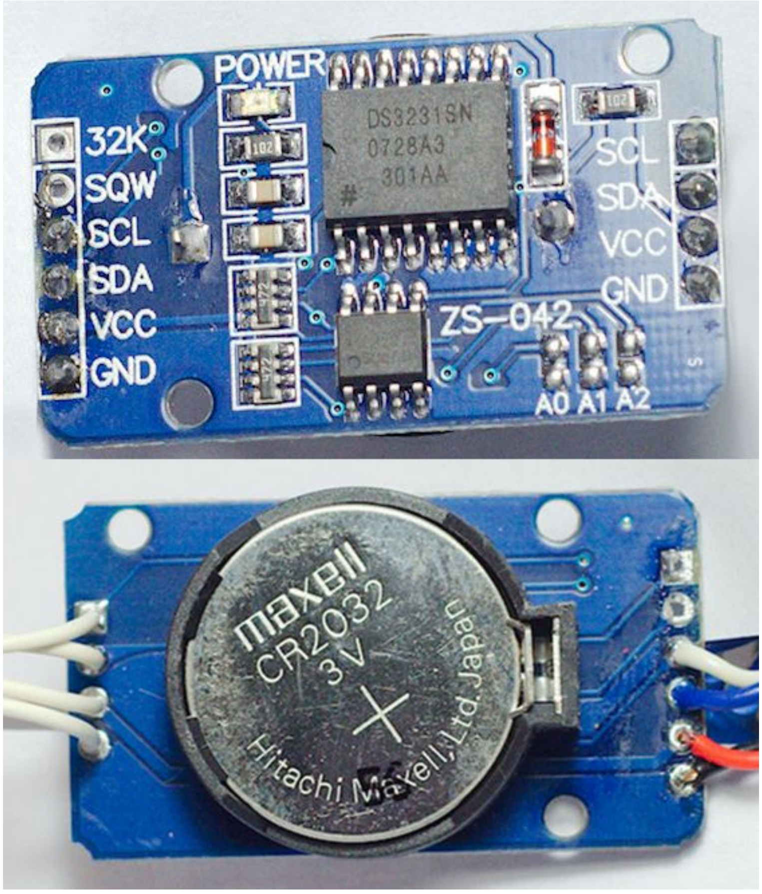


Figure 4.9: The real time clock module used in the Mansell Power Lifter data logger featuring the DS3231 real time clock integrated circuit. (Photo: author)

allow for it to be easily integrated with the DS1631 which also uses the I2C bus. One further important reason for choosing the DS3231 is that it was readily available on a small size and low cost printed circuit board module, with all of the required support circuitry, and a coin cell backup battery, thus reducing the development time required to implement it. A picture of the DS3231 real time clock module which was used in the data logger can be seen in Figure 4.9.

4.4 Micro-SD Card Data Storage

As outlined in the hardware design considerations section in Chapter 3, the data logger will use a micro-SD card to record the operational data gathered by the data logger's various sensors in CSV format. Micro-SD cards and SD cards in general use the SPI bus to send and receive data with their connected microcontrollers. While the ATMega328P used in the Arduino Nano does have the SPI bus built in, it will not be possible to directly

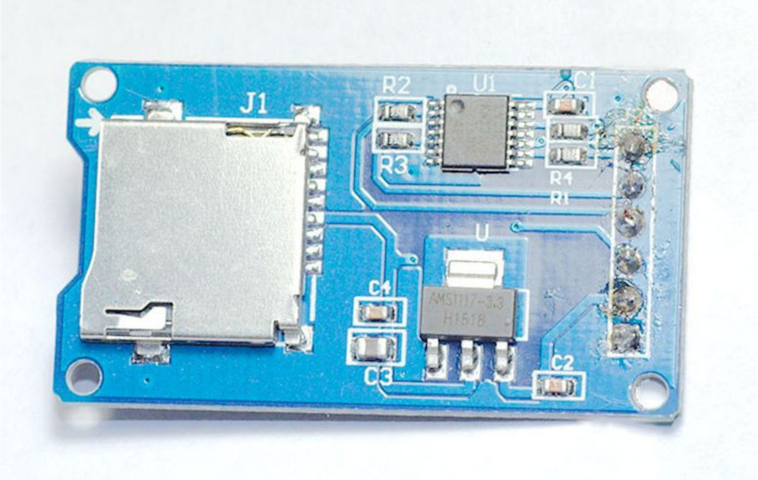


Figure 4.10: The CATALEX MicroSD Card Adapter used in the Mansell Power Lifter data logger to interface with the micro-SD cards used to store recorded data. (Photo: author)

connect it to the micro-SD card since they both operate at different voltage levels. The ATMega328P is powered from 5V and uses 5V digital logic while all micro-SD and SD cards are powered from 3.3V and use 3.3V logic. Connecting the two directly could result in failure to communicate properly or damage to either component. As a result it will be necessary to place some kind of logic level conversion circuitry between the two.

There are two main techniques commonly used for logic level conversion, series resistors and purpose built logic level translators. While either level conversion technique can work for the SPI bus, integrated circuit logic level translators are considered to be a better overall solution because they can guarantee that the voltage levels fed into the integrated circuits match the appropriate voltage levels, and due to the fact that they generally contain good overvoltage protection (Maxim Integrated 2004).

In addition to the need to translate the logic levels for the micro-SD card it will also be necessary to provide a local 3.3V regulated power source to supply power to the micro-SD card and a micro-SD card socket for the card to plug in to. Given these requirements it was determined that in order to minimise the hardware development time it would be necessary to obtain a pre-produced micro-SD card module complete will all required support circuitry. The CATALEX MicroSD Card Adapter module was identified as a suitable product and chosen to be integrated into the data logger design since it contains a dedicated logic level translation integrated circuit and the required power supply circuitry. A picture of the micro-SD card module chosen can be seen in Figure 4.10.

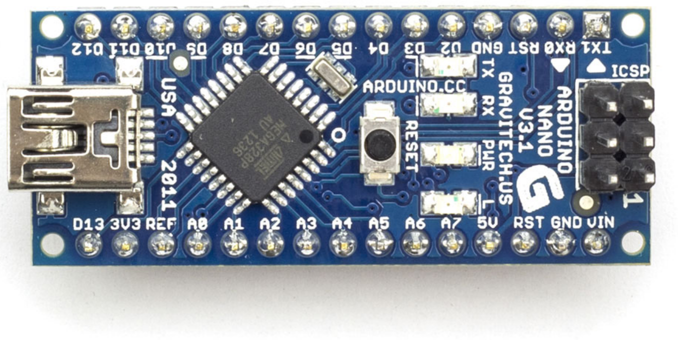


Figure 4.11: The Arduino Nano microcontroller development board featuring the Atmel ATMega328P microcontroller (Arduino 2016*a*).

The micro-SD cards chosen to be used in the data logger for recording the sensor data were SanDisk Ultra 8GB micro-SD cards. These cards were chosen because they were reasonably priced and were the smallest capacity readily available locally. Testing of the final design of the data logger showed that for every second that the Mansell Power Lifter was operating and the data logger was collecting data, approximately 7KB of data was written to the micro-SD card. At a rate of 7KB/s, and with an 8GB micro-SD card it would take well over 300 hours of operation for the micro-SD card to be filled. Given that it was only planned for the data loggers to be deployed to the field for a couple of months at a time it was decided that the 8GB micro-SD cards chosen would have more than sufficient capacity.

4.5 Arduino Nano

The Mansell Power Lifter data logger will use an Arduino Nano microcontroller development board featuring the Atmel ATMega 328P, the motivations for which were outlined in the design considerations in Chapter 3. A picture of the Arduino Nano can be seen in Figure 4.11.

In order to connect to the various sensors and modules which will be included in the data logger, a number of the Arduino Nano's input and output functions will be used. Two of the eight analogue-to-digital converter inputs will be used to connect to the two input voltage sensing circuits, and another two will be used to measure the output voltage of the ACS712-30A Hall Effect current sensors. The DS1631 temperature sensor and DS3231 real time clock will be connected to the I2C bus on the ATMega328P which is an alternate function for the A4 and A5 analogue inputs on the Arduino Nano. The micro-SD card module will be connected to the SPI bus pins of the ATMega328P through the four digital pins D10 to D13 on the Arduino Nano. Finally, the output of a pushbutton and resistor circuit will be connected to digital pin D2 of the Arduino Nano to a form physical counter reset function for the elapsed time and limit switch activation totals stored in the ATMega328P's EEPROM. A full summary of the pin connections of the ATMega328P in the Arduino Nano can be seen in Table 4.1.

The ATMega328P in the Arduino Nano comes out of the box with a bootloader installed, which allows for user code to be loaded onto the microcontroller through a USB cable, and included USB to serial converter on the printed circuit board. This default programming configuration has the benefit of enabling the microcontroller to be programmed out of the box with no additional hardware other than a USB cable (Arduino 2016*a*). Due to the ease and simplicity of programming, it was initially decided to program the Arduino Nano using the standard USB connection and bootloader, however after programming it a number of times it was discovered that the microcontroller would take a significant amount of time to start up and begin running user code. In order to quantify this time delay a test was performed using a digital oscilloscope to determine exactly how long it was taking for the microcontroller to start up.

The Arduino Nano was programmed to simply output a high voltage level on one of its digital I/O pins immediately after it began running user code. The Vcc supply was connected to channel one of the oscilloscope and the digital I/O pin in use was connected to channel two. Both channels were referenced to circuit ground. The oscilloscope was configured to trigger on a rising voltage on channel one. As can be seen in Figure 4.12 the results of the test showed that using the standard Arduino Nano bootloader it would take 1.5s for user code to begin running. The implications of this start up time on the data logger would be that every time the Mansell Power Lifter was used, 1.5s of data at the start of every operation would not be recorded.

In order to confirm that the Arduino Nano bootloader was in fact the cause of the start

Table 4.1: ATMega328P and Arduino Nano microcontroller pin connections.

Arduino Nano Pin No./Board Label	**Primary Pin Functions**	**Pin Usage**
1/TX1	Digital I/O & UART Tx	Not in use.
2/RX0	Digital I/O & UART Rx	Not in use.
3/RST	Reset	Not in use.
4/GND	Ground	Not in use.
5/D2	Digital I/O	Digital input connected to the pushbutton.
6/D3	Digital I/O	Not in use.
7/D4	Digital I/O	Not in use.
8/D5	Digital I/O	Not in use.
9/D6	Digital I/O	Not in use.
10/D7	Digital I/O	Not in use.
11/D8	Digital I/O	Not in use.
12/D9	Digital I/O	Not in use.
13/D10	Digital I/O & SPI SS	Connected to the slave select (SS) line of the micro-SD card module.
14/D11	Digital I/O & SPI MISO	Connected to the master in slave out (MISO) line of the micro-SD card module.
15/D12	Digital I/O & SPI MOSI	Connected to the master out slave in (MOSI) line of the micro-SD card module.
14/D13	Digital I/O & SPI SCK	Connected to the serial clock (SCK) line of the micro-SD card module.
15/3V3	3.3V	Not in use.
16/AREF	ADC reference voltage in	Not in use.
17/A0	Digital I/O & ADC	Analogue input connected to the rear actuator voltage sense circuit.
18/A1	Digital I/O & ADC	Analogue input connected to the front actuator voltage sense circuit.
19/A2	Digital I/O & ADC	Analogue input connected to the rear actuator ACS712-30A current sensor.
20/A3	Digital I/O & ADC	Analogue input connected to the front actuator ACS712-30A current sensor.
21/A4	Digital I/O & ADC & I2C data	Connected to the serial data (SDA) line of the I2C bus.
22/A5	Digital I/O & ADC & I2C clock	Connected to the serial clock (SCK) line of the I2C bus.
23/A6	ADC	Not in use.
24/A7	ADC	Not in use.
25/5V	5V voltage input/output	Connected to the 5V output of the power supply.
26/RST	Reset	Not in use.
27/GND	Ground	Connected to the ground of the power supply.
28/VIN	Built in voltage regulator input	Not in use.

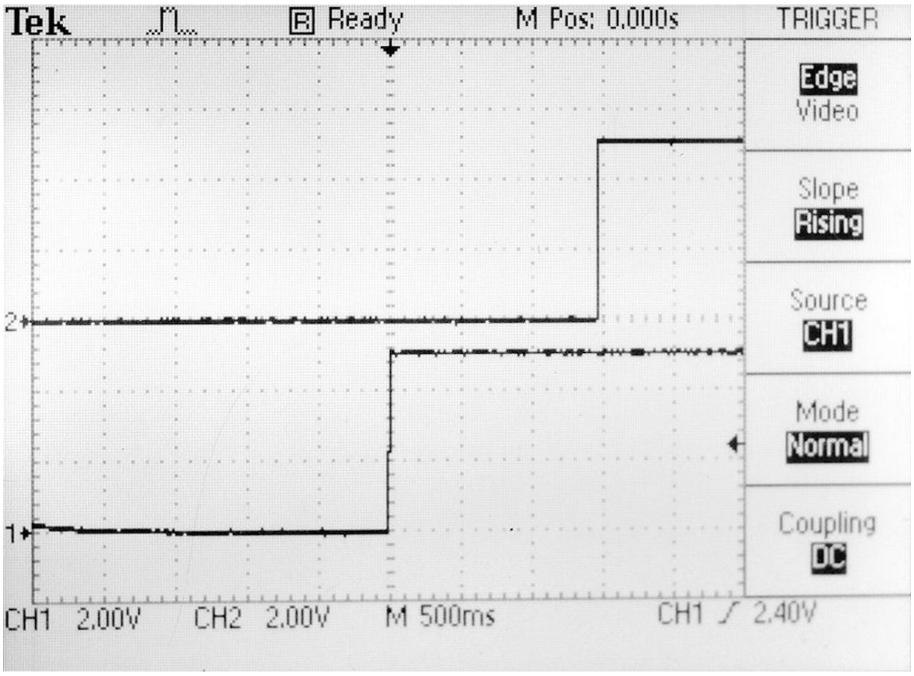


Figure 4.12: Arduino Nano start up time test with Arduino Nano bootloader. (Photo: author)

up delays the same test was performed except with the user code programmed directly to the ATMega328P using the In-System Programming (ISP) header on the circuit board. The results of the test can be seen in Figure 4.13 and show that by programming the microcontroller directly the start up delay is significantly reduced to approximately 75ms. The programming of the ATMega328P through the ISP header was achieved using an Arduino Uno development board which was programmed to operate as an ISP programmer for other Atmel microcontrollers such as the ATMega328P (Arduino 2016*b*). Since any loss of data is undesirable for the data logger, it was decided that the Arduino Nanos used in the data logger would be programmed directly through the ISP header in order to avoid the excessive start up delays experienced with the bootloader.

Though the ISP programming method was used for both of the data loggers developed, it should be noted that further experimentation after the development process was complete, discovered that by loading the bootloader for the more recently released Arduino Uno, which contains the same ATMega328P microcontroller, the boot delay originally experienced with the Arduino Nano bootloader is no longer present. Furthermore, the start up times were found to be comparable to directly programming the microcontroller through the ISP header to within a few milliseconds.

One final hardware design feature that was added to the Arduino Nano connections was

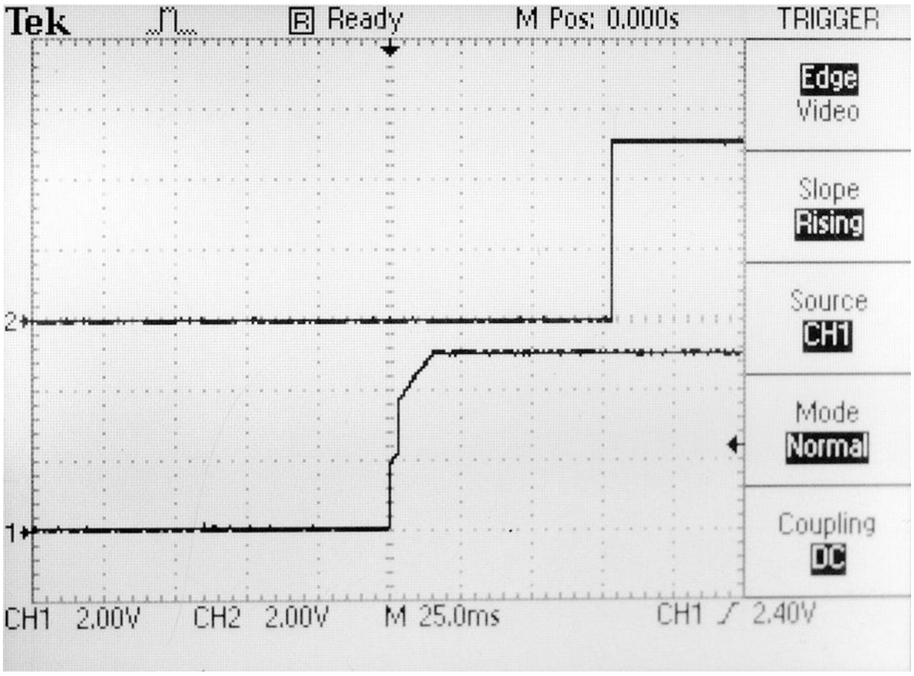


Figure 4.13: Arduino Nano start up time test with no bootloader. (Photo: author)

the addition of 1kΩ series resistors on the Master In Slave Out (MISO), Master Out Slave In (MOSI), and Serial Clock (SCK) lines of the SPI bus. The addition of these resistors was necessary because the Atmel ISP programming interface uses these three lines from the SPI bus to facilitate programming of the microcontroller. Since these three SPI bus lines are shared between SPI bus and ISP programming duties it is necessary to add the series resistors to prevent SPI devices from driving the MISO, MOSI, and SCK lines while programming of the microcontroller through the ISP interface is in progress (Atmel 2008).

4.6 Physical Construction

The physical construction of the Mansell Power Lifter data logger was an important consideration in the design and development process, since in order to meet the design requirement of being as unobtrusive to the normal operation of the system as possible, it is important to ensure that the data logger is able to integrate easily with the Mansell Power Lifter. After an analysis of the physical construction of the Mansell Power Lifter it was determined that the most appropriate place to attach the data logger was directly behind the power box underneath the power lifter's main deck. Installing the data logger in this position would put it out of the way of moving parts and user interaction areas.

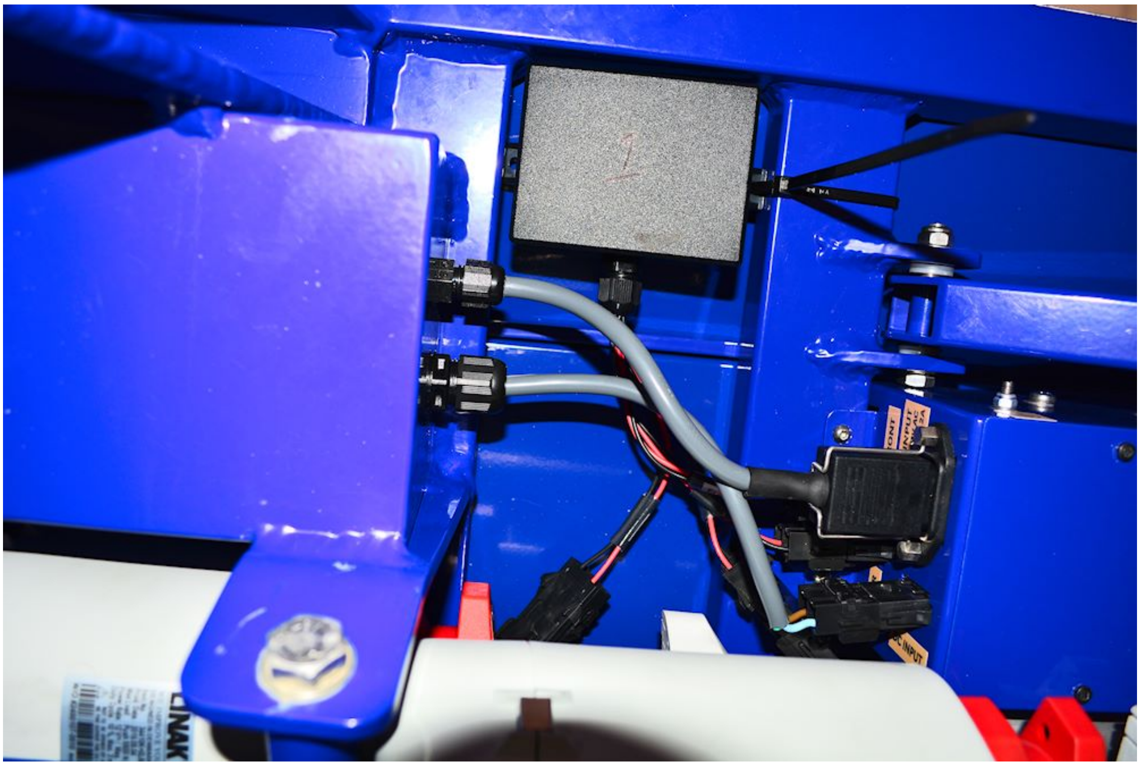


Figure 4.14: One of the data loggers attached to a Mansell Power Lifter. (Photo: author)

Additionally, installing the data logger so close to the power box which the linear actuator power cables connect to would minimise the cabling required to connect the actuators. In order to house the data logger components a black plastic electronics potting box (85x65x32mm) was obtained which would fit the desired location of the Mansell Power Lifter. The data logger was secured in place using cable ties passed through each of the brackets on the enclosure and around the structural bars of the Mansell Power Lifter. A picture of one of the data loggers fitted to a Mansell Power Lifter can be seen in Figure 4.14.

The modules which make up the final data logger are the power supply circuit board, which was constructed on a prototyping printed circuit board and includes power supply components, voltage sensing circuits, and the counter reset pushbutton, the two ACS712-30A current sensors, Arduino Nano, DS1631 temperature sensor, DS3231 real time clock, and micro-SD card module. Inside the data logger enclosure the components were arranged into two layers. The power supply, both ACS712-30A current sensors, and the Arduino Nano were placed on the bottom layer and soldered together onto a strip of prototyping board to keep them rigidly aligned. Meanwhile, the DS3231 real time clock and micro-SD card modules were soldered onto another strip of prototyping board to hold them rigidly in place and formed the top layer. Both layers were connected using individual wires which were cable tied together to keep them neat. Additionally, a piece of non-conductive card was cut out and placed between the two layers to prevent any short circuits which could form due to component movement.

Since the DS1631 shares the I2C bus with the DS3231 it was connected to the I2C pass-through headers on one end of the DS3231 module using short wires and attached to the side of the enclosure using double sided tape to assist in accurately measuring the ambient temperature outside the enclosure. It should be noted that although the I2C bus requires pull up resistors on both the SDA and SCK lines (Atmel 2009) it was not required for these to be added to the design since the DS3231 module obtained includes these resistors on board.

The data logger was connected to the Mansell Power Lifter power box and linear actuators using the same large gauge wire as is used internally to the power box and the same connectors as featured on the power box and actuator power cables. The data logger connects directly to the power box using two sets of cables each for the front and rear actuators respectively. The two sets of cables enter the data logger through the cable grommet. Inside the power box a portion of the red wires is stripped back and a small power wire is tapped off of each to go to each of the power supply inputs respectively before the wires are bent 180° to exit the data logger, once again through the cable grommet. Meanwhile, the black wires are cut and the stripped back ends inserted into one of the screw terminals of their respective ACS712-30A current sensors. As with the red wires a small power wire is tapped off from each of the black wires to connect to each input of the power supply, except here an inline fuse is soldered in series. The remaining terminals of the ACS712-30A current sensors are connected to by an additional two black wires which make their way out of the data logger through the cable grommet and are reunited with their matching red wire in the output connectors which will be attached to the actuator power cables.

An overall view of the data logger with the lid removed to show the top layer of modules inside the enclosure can be seen in Figure 4.15 and a close up photograph of the second layer of modules inside the data logger showing the internal acuator current carrying cables can be seen in Figure 4.16.

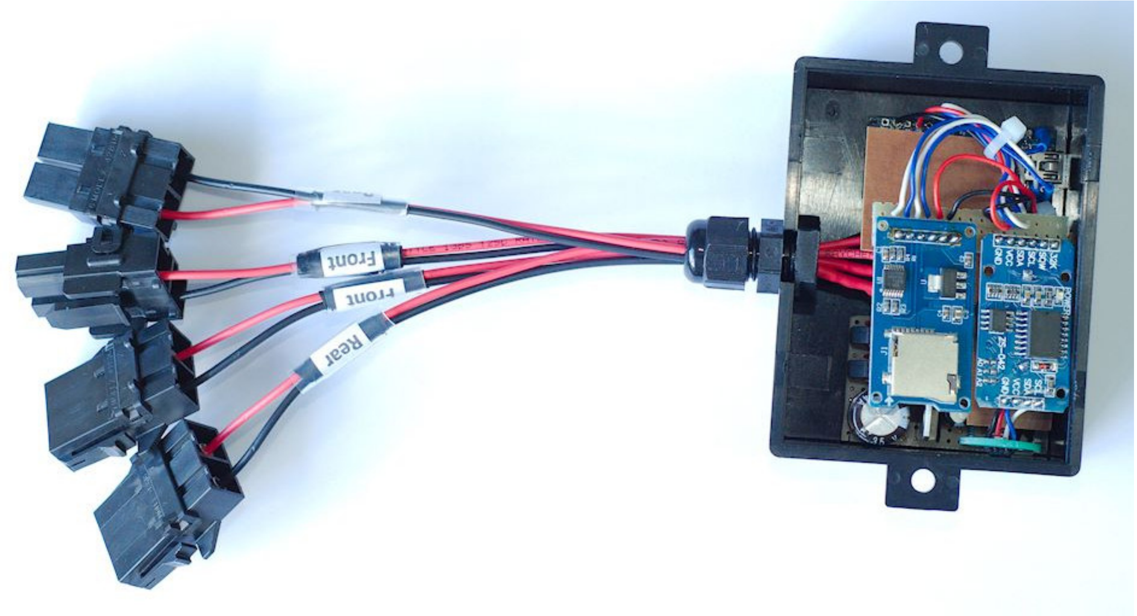


Figure 4.15: The final construction of the Mansell Power Lifter data logger showing the external cables and the top layer of internal modules. (Photo: author)

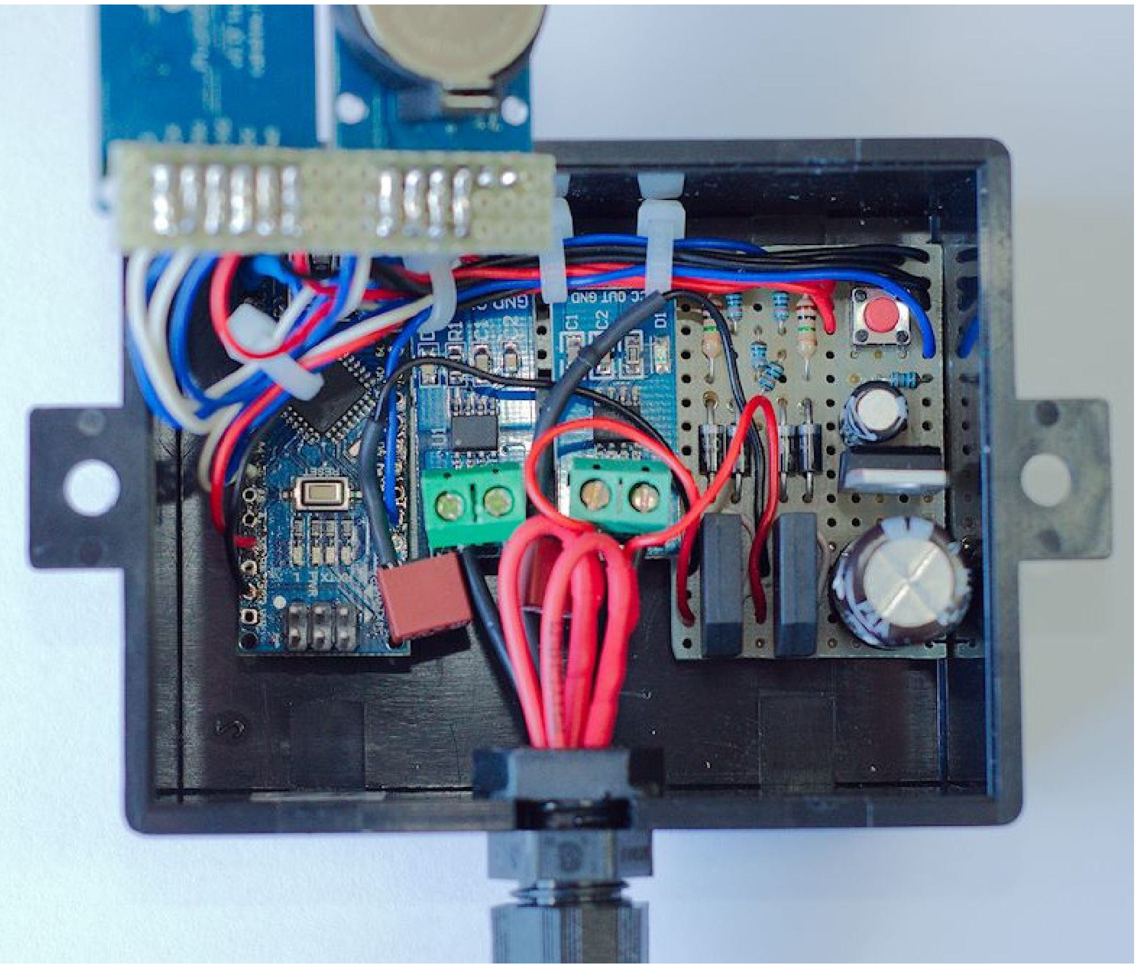


Figure 4.16: The final construction of the Mansell Power Lifter data logger showing the second layer of internal modules and the internal current carrying actuator wiring. (Photo: author)

4.7 Chapter Conclusion

This chapter has outlined the design and development decisions made regarding each of the major hardware components of the Mansell Power Lifter data logger in the categories of power supply, sensors and data acquisition, microcontrollers, and data storage. The physical construction of the data logger and its integration into the Mansell Power Lifter has also been discussed. The next chapter will focus on the software design and development aspects of the data logger.

Chapter 5

Data Collection System Software Design

5.1 Chapter Overview

This chapter will discuss the design and development of the software aspects of the data logger system. The design considerations explored in Chapter 3 will be applied to the final design of the data logger software. As outlined previously, the data logger software will be developed in the Arduino Integrated Development Environment (IDE) and will use a finite state machine program structure. Some of the software libraries and functions used in the data logger software were not written by the author and were either available freely as part of the Arduino IDE or from other sources. Where appropriate the authors of these functions have been indicated using comments in the code itself. The full Arduino C/C++ code which was developed for the data logger can be seen in Appendix C.

5.2 Program Structure

The Arduino IDE works with the concept of having two main functions in the source code `setup()` and `loop()`. The contents of the `setup()` function runs once upon the microcontroller starting up and the contents of the `loop()` function are executed repeatedly for the remainder of the time that the microcontroller is running. All user code other

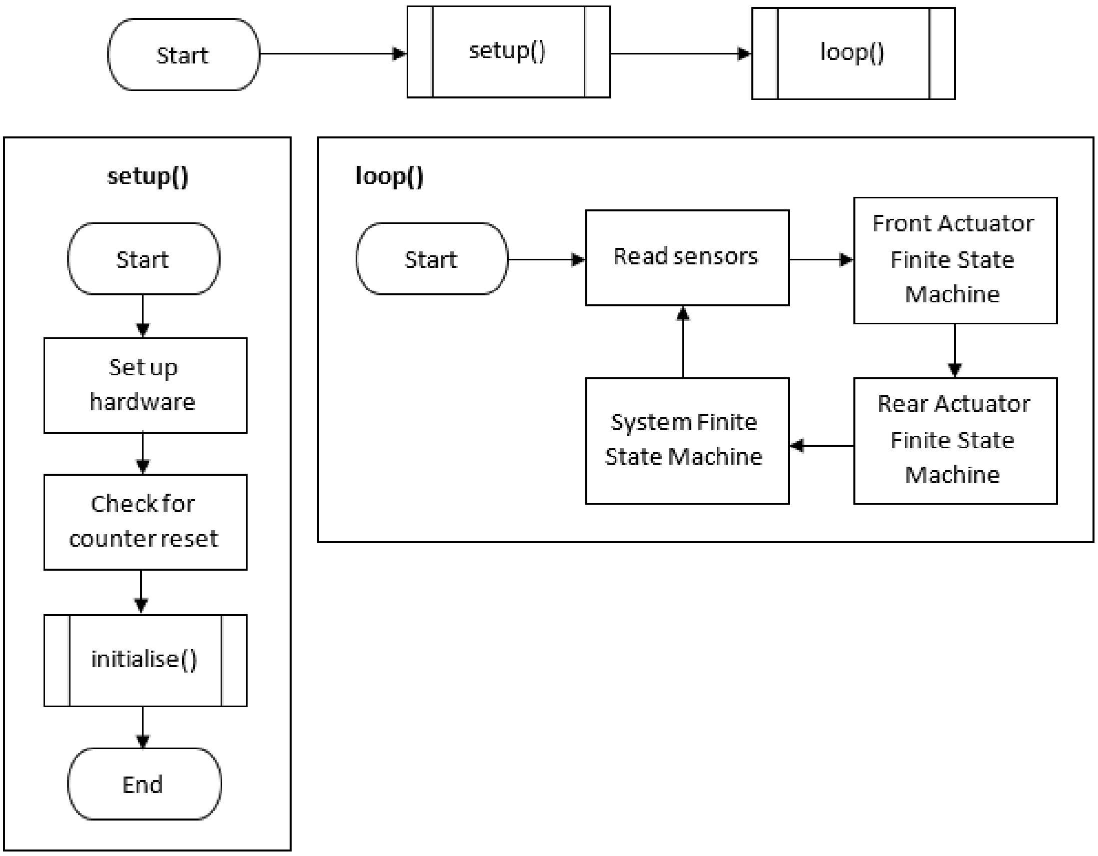


Figure 5.1: Mansell Power Lifter data logger software structure flowchart. (Source: author)

than functions and variable declarations must be inside one of these two main functions (Evans 2007).

For the data logger software, the `setup()` function will be used to perform hardware initialisation and counter reset functionality while a separate initialisation function (`initialise()`) takes care of initialising the system variables. A separate initialisation function is required since under some circumstances the data logger will need to initialise its variables outside of a power cycle event. The `loop()` function will be used to perform the collection and analysis of data from the various sensors as well as the finite state machine logic. Specific tasks such as reading sensors and outputting processed data will be handled by dedicated functions. An overview of the structure of the program can be seen in the form of a flowchart in Figure 5.1.

5.3 Initialisation

As stated previously, when the ATMega328P in the Arduino Nano starts up it first runs the code in the `setup()` function which in the case of the Mansell Power Lifter data logger is being used to perform hardware initialisation functionality and process the counter reset functionality. Upon entering the `setup()` function the first task that is performed is that the digital pin for the counter reset button is set as a digital input using the Arduino `pinMode()` function. Next, timer 1 in the ATMega328P is set up to interrupt the system for data collection every 100ms by updating the values of the relevant configuration registers, the rationale for which will be outlined later in this chapter.

The next hardware initialisation steps to be performed are the enabling of the I2C and SPI busses. The I2C bus is initialised using the Arduino `Wire.begin()` function and the SPI bus with the Arduino `SD.begin(SD_CS_PIN)` function, where `SD_CS_PIN` is a constant defined with the pin number of the SPI chip select line. The final hardware initialisation task that is performed is to set up the DS1631 temperature sensor. The DS1631 is configured to perform continuous 9 bit temperature conversion by sending `0xAC` and `0x80` to the address of the DS1631 on the I2C bus. Finally in order to start the temperature conversion process `0x51` is sent to the DS1631 on the I2C bus.

After the hardware initialisation process is complete the `setup()` function reads the state of the counter reset button input. If the user has been holding the reset button down while the data logger is starting up each of the actuator elapsed time counters and limit switch activation counters stored in the ATMega328P's EEPROM will be reset to zero as well as their matching working variables stored in RAM. The final task that the `setup()` function performs is to call the `initialise()` function which sets up all of the working variables ready for the data logger to begin monitoring the actuators.

5.4 Finite State Machine Elements

After startup and initialisation are complete there are three finite state machines which govern the operation of the data logger, the "overall system" state machine and individual state machines for each linear actuator. The front and rear actuator state machines are nested within the “overall system” state machine. Additionally, all of the state machines have been represented as Mealy machines since there are operations associated with the transitions between states.

5.4.1 Overall System State Machine

The “overall system” finite state machine takes control of the execution of the data logger once the start up hardware and software initialisation is complete. It ensures that the system is able to transition between normal operation and power-off cleanup conditions without losing any data which has been collected from the sensors. After initialisation, and when one or both actuators are running, the data logger is in the “normal operation” state where the two actuator state machines take care of monitoring the current status of each of the actuators. In the “normal operation” state actuator data that has been collected from the sensors is written to the `log_full` file on the micro-SD card.

If the data logger detects that the voltage on both of the actuator power cables falls below the threshold level of 9V then the user has released all of the buttons on the user control panel and the current “operation” is said to have finished. A voltage threshold level of 9V was chosen because the voltage supplied to the actuators should never fall below roughly 10V in normal operation. By placing the voltage threshold close to the theoretical minimum supply voltage, the affect of any voltage decay time can be minimised. Since neither of the actuator power lines are being driven in this case the data logger is not being supplied with power and is running off of the capacitor reserve in the power supply. When an operation ends it is essential that the “overall system” state machine perform the pre-poweroff operations and enter the “wait for poweroff” state before the supply voltage drops and the microcontroller shuts down. This will ensure that any remaining data is written to the `log_full` and `log_short` files on the micro-SD card before power is lost.

The “wait for poweroff” state consists of a single while loop which continuously polls the voltage on both of the actuator power cables. If the user presses one of the actuator buttons on the control panel before the microcontroller loses power, the data logger soft-

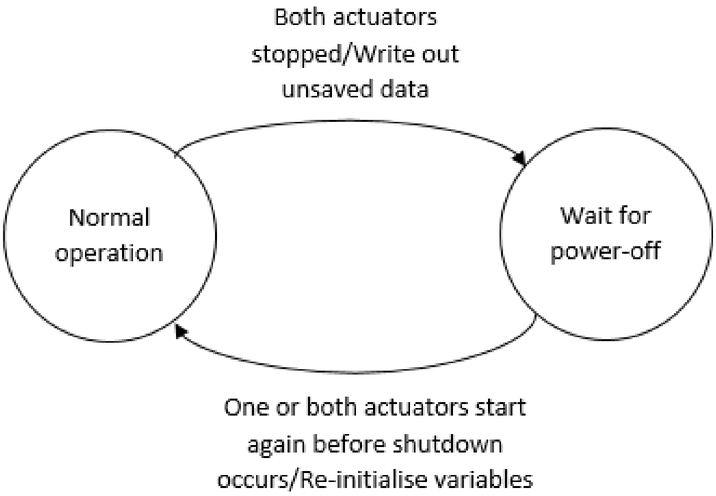


Figure 5.2: Mansell Power Lifter data logger “overall system” state diagram. (Source: author)

ware will perform an initialisation of its working variables and transition back into the “normal operation” state. If power is not restored to either of the actuator power cables the microcontroller will lose power shortly after entering the “wait for poweroff” state due to the capacitors in the power supply discharging and voltage regulation being lost. The state chart for the “overall system” finite state machine can be seen in Figure 5.2.

5.4.2 Front and Rear Actuator State Machines

The “front actuator” and “rear actuator” finite state machines operate within the “normal operation” state of the “overall system” state machine. The two state machines monitor the operation of each of the linear actuators and update several of the counter and timer variables as they transition between operational states. If one of the actuators was in the “actuator stopped” state and the voltage on its actuator cable rises above the threshold level of 9V the associated actuator state machine transitions into the “actuator running” state by resetting its associated operation timer variables. As the actuator is being operated, the “actuator running” states update the actuator’s timer and counter variables after each sampling period.

If an actuator state machine detects that its associated actuator is powered on but has a current below the operational current threshold of 0.5A there is a possibility that the actuator has hit one of its internal limit switches. An operational current threshold of 0.5A was chosen since due to inaccuracies in the measurement of the output of the ACS712-30A current sensors, even when no current is flowing through the actuators there is a small current reading present. Testing of the system showed that the no current error reading would never go above 0.5A and therefore this was chosen as the threshold.

Since it was decided to monitor the limit switch activations, this situation where the actuator is powered on but not running is one which the data logger must be able to detect. If the data logger detects that the actuator has been powered on but is without operating current for three sampling periods, it assumes that one of the internal limit switches has been activated and increments the associated actuator limit switch counter before transitioning into the “limit switch activated” state. It was determined that it was necessary to wait three sampling periods to confirm a limit switch activation, since in the case of actuator direction changes it is possible for the data logger to read close to zero current for one to two samples.

Since an actuator limit switch activation takes three sampling periods of 100ms each to be confirmed, it was theorised that it may be possible for the limit switch to be activated and the user to let go of the control button before the activation was registered. However, testing showed that this was unlikely to occur unless the user released the control button extremely quickly. One limitation of this limit switch detection method which should be noted is that it cannot distinguish between which of the two limit switches within the actuator was activated. It is feasible that this may be implemented by analysing the direction of the current flow just prior to the activation, however since this feature was added at the very end of the development period there was not enough time to implement it.

Once one of the actuator state machines is in the “limit switch activated” state it can either return to the “actuator running” state or transition to the “actuator stopped” state. The actuator state machine will enter the “actuator running” state if the current in the actuator returns to a level above the threshold. In the process of changing to the “actuator running” state the limit switch activation detection variables will be reset. If the voltage present on the actuator power cable drops below the voltage threshold the actuator state machine will transition into the “actuator stopped” state. In the process of transitioning the state machine will update the actuator’s elapsed time variable and associated value in the microcontroller’s EEPROM by adding to the existing value the run time of the current operation. Similarly, if the actuator state machine is in the "actuator running" state and the voltage on the associated power cable drops below the threshold, the state machine will transition into the "actuator stopped" state by updating the same timer variables. The state charts for both the front and rear actuator finite state machines can be seen in Figure 5.3.

5.5 Measurement Timing Considerations

5.5.1 Timers and Interrupts

In order to perform the collection of the sensor data at regular intervals it was necessary to implement some form of timer mechanism to set the delay between each collection point. While it would have been possible to achieve this using a delay function after each measurement the actual delay between each measurement would have been longer than anticipated due to the processing performed after each measurement. Instead it was decided to set up one of the system timers as an interrupt source for the microcontroller such that at the end of each collection interval a new measurement would be consistently triggered.

The timers and interrupts were configured by setting the prescaler for timer 1 in the ATMega328P to 1024 and enabling clear timer on compare match mode using the registers `TCCR1A`, `TCCR1B`, and `TCCR1C`, such that upon reaching a predetermined value the timer will reset back to zero. The `OCR1AH` and `OCR1AL` timer compare registers were configured with a combined value of 1562, which given the microcontroller clock frequency and timer prescaler would result in a interval of approximately 100ms. Finally the timer interrupt mask register for timer 1 (`TIMSK1`) was configured such that when the value of the timer matched the value in the timer compare register the microcontroller would be interrupted.

The way in which it was decided to handle the interrupts for data collection timing was that after a successful data collection and all associated processing was completed the microcontroller would enter a while loop based on the value of a volatile boolean variable `waiting`. Upon entering the while loop `waiting` was set to true and the microcontroller became stuck in the while loop. When the timer created an interrupt, the `ISR(TIMER1_COMPA_vect)` function was called and the value of `waiting` was set to false.

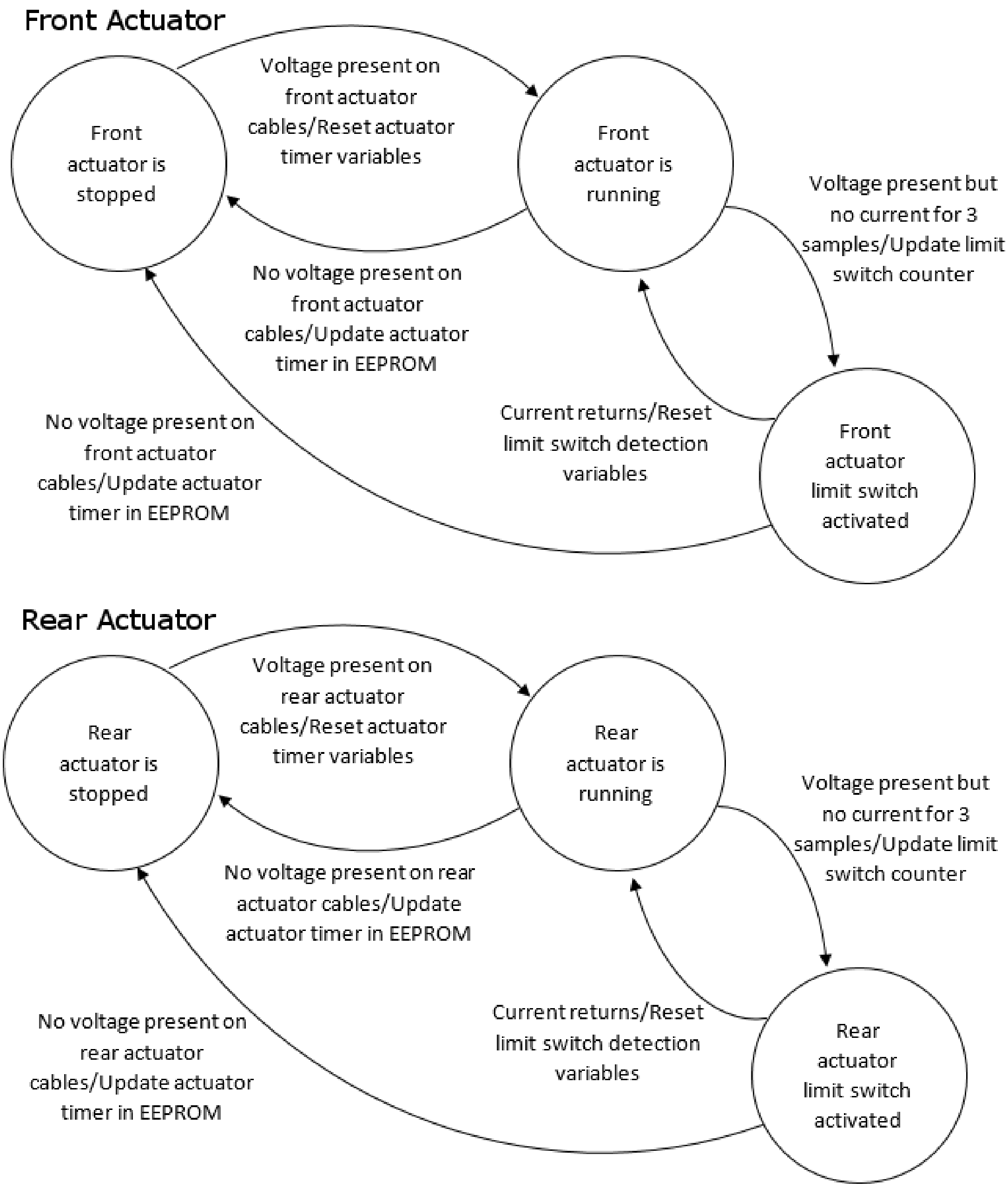


Figure 5.3: Mansell Power Lifter data logger software front and rear actuator state diagrams. (Source: author)

Upon returning to the while loop the microcontroller would detect that the value of `waiting` had changed and would break out of the while loop and initiate a new data collection cycle.

5.5.2 Data Storage Write Time

One concern that was identified during the development phase of the data logger was that the time taken to write the collected data to the micro-SD card might be greater than the data collection interval time due to the speed of the SPI interface on the ATMega328P. If this was the case it would be possible that the system might skip a data collection point due to the timer interrupt happening while the microcontroller was still writing to the micro-SD card. As a result data would only be collected from the sensors and written to the micro-SD every second data collection interval.

In order to determine if this was likely to be a problem a test was performed. The data logger code was modified such that one of the digital pins on the Arduino Nano would output a high level until the micro-SD data writing function was called at which point it would drop to a low level until all data had been written and it returned to a high level. By viewing the output of this digital pin with an oscilloscope it would be possible to determine how much time was required to write all of the required data to the micro-SD card and whether or not there would be any timing issues. A screen capture of the oscilloscope while performing one of these tests can be seen in Figure 5.4, although it should be noted that in this case the test was performed during a power-off operation where the amount of data written is slightly more than normal. As can be seen, even in the worst case scenario it will only take approximately 40ms to write the data to the micro-SD which is significantly less than the 100ms data collection interval and therefore the aforementioned issues should not be experienced.

5.6 Additional Functions

An overview of a number of additional functions used by the Mansell Power Lifter data logger will be given in the following sections.

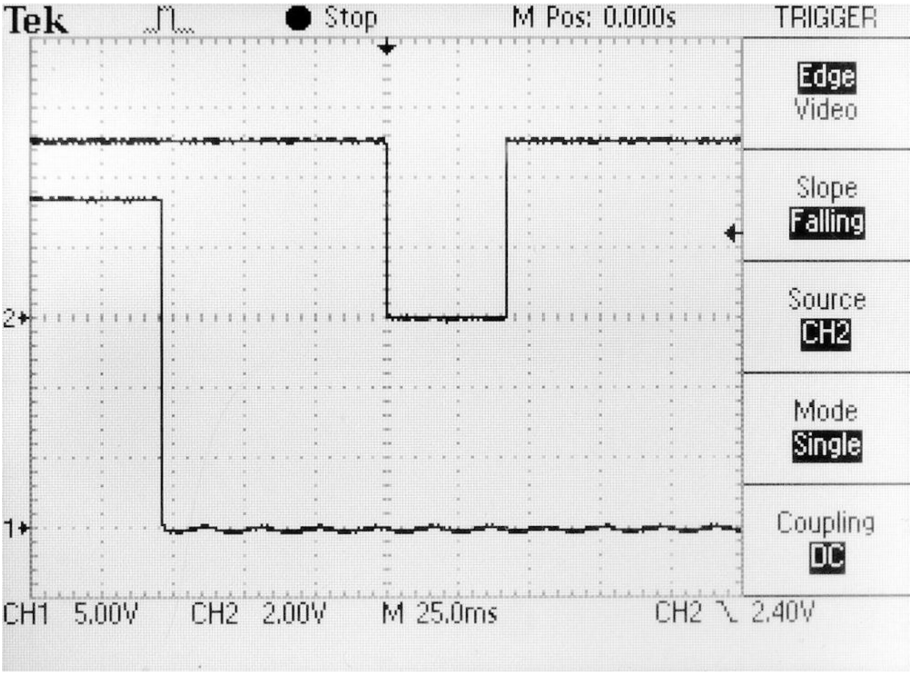


Figure 5.4: Data logger micro-SD card data write time test. (Photo: author)

5.6.1 The `readVoltage(int)` Function

The `readVoltage(int)` function is a user defined function for reading the current value from one of the actuator voltage sensing circuits. It takes a single integer parameter that defines which analogue input pin on the Arduino Nano is to be read from and returns a floating point value which is the current voltage being supplied to the specified actuator. The function first reads the current value from the analogue-to-digital converter using the Arduino `analogRead(int)` function and converts it into the actual voltage seen on the Arduino pin. Finally, the second order polynomial scaling factor outlined in Chapter 4 was applied to the voltage on the Arduino pin in order to obtain the actual voltage on the actuator power cables.

5.6.2 The `readCurrent(int)` Function

The `readCurrent(int)` function operates similarly to the `readVoltage(int)` function in that it takes a single integer parameter corresponding to one of the analogue-to-digital input pins on the Arduino, however in this case it returns the current in the specified actuator as a float. This is achieved by reading the analogue-to-digital converter value for the specified pin using the Arduino `analogRead(int)` function before converting the raw analogue-to-digital converter value to a voltage and applying the sensitivity scaling factor for the ACS712-30A Hall Effect current sensors.

5.6.3 The `readTime()` Function

The `readTime()` function was adapted from code written for the DS3231 real time clock by John Boxall at Tronixstuff. This function requests the current time from the DS3231 over the I2C bus and then converts it into a byte array containing the current values for hour, minute, etc. One caveat of the data returned by the DS3231 is that it is in binary-coded-decimal format. In order to convert each of the parameters to normal integers two functions also obtained from Tronixstuff `bcdToDec(byte)` and `decToBcd(byte)` were used to perform the required conversions (Boxall 2014).

5.6.4 The `readTemp()` Function

The `readTemp()` function was adapted from code written by Paul Priebbenow at Wenross Holdings Pty. Ltd. for the DS1631 temperature sensor. The `readTemp()` function works by requesting the current temperature from the DS1631 over the I2C bus which is received as two bytes. To convert the most significant and least significant byte received from the DS1631 into a single floating point value, the function combines the two by bit-shifting and scaling the least significant byte before adding it to the most significant byte.

5.7 Chapter Conclusion

This chapter has outlined the software design of the Mansell Power Lifter data logger. The structure of the program has been explored through the use finite state machine state charts and other key functions and features have been explained. While there are certainly possibilities for the code to be optimised and improved, particularly through the de-duplication of front and rear actuator state machine functionality, the software for the data logger was deemed to have implemented all of the required functionality and be stable enough to be used in the data loggers in the field. The next chapter will focus on the testing procedures performed by deploying the data loggers to Mansell Power Lifters in real world use and discuss the analysis of the results obtained.

Chapter 6

Data Analysis and Solution Proposal

6.1 Chapter Overview

This chapter will discuss the data collection operations performed by the Mansell Power Lifter data logger and provide an analysis and discussion of the results obtained. In light of the results, the cause of the linear actuator failures will be explored, and based on this a potential solution to the failures, which could be implemented in the future will be put forward.

6.2 Data Collection Procedure

In order to perform the collection of data from real world Mansell Power Lifters, two of the data collection devices outlined in the previous chapters were constructed. One of the data loggers was fitted to a Mansell Power Lifter at the Royal Brisbane & Women's Hospital in Brisbane and the other to a Mansell Power Lifter at the Royal Children's Hospital in Melbourne. Both of these hospitals are major centres for infant retrievals and have several Mansell Infant Retrieval Systems each. These locations were chosen for data collection purposes since the significant regular use of the systems in these hospitals will produce the largest amount of data possible and should represent a worst case scenario for excessive use of the system, if this is found to be the case. From here on the data logger which was fitted to a Mansell Power Lifter at the Royal Brisbane & Women's Hospital will be referred to as Data Logger #1 and the one fitted to a Mansell Power Lifter at the Royal Children's Hospital will be referred to as Data Logger #2.

Data logger #1 was fitted to its Mansell Power Lifter for a total of 79 days and data logger #2 was fitted to its Mansell Power Lifter for a total of 43 days. At the end of the testing period both of the data loggers were returned to Wenross Holdings Pty. Ltd. Over the course of the testing period, no issues were reported where the data loggers themselves impacted system operation and no actuator failures were reported as occurring. A picture of one of the data loggers fitted to a Mansell Power Lifter can be seen in Figure 4.14 in Chapter 4.

6.3 Data Collection Faults

While every attempt was made, and significant testing performed, in order to ensure the correct and reliable operation of the data loggers, there were a number of minor issues which were identified only after the data loggers had been returned and the data analysed. These faults are summarised as follows:

While at the time of initially designing the data logger power supply it was determined that the oversized capacitors on the voltage regulator would provide sufficient power for the Arduino to complete its power-off tasks and successfully write all remaining data to the micro-SD card. After receiving the data loggers back from field testing, it was discovered that if the data logger was only powered on for a very short amount of time in 12V operating mode, it was possible that the Arduino would not be able to write all data to the micro-SD card before losing power and therefore some data would be lost. This is likely due to the addition of extra parameters over time which also need to be recorded to micro-SD card, thus increasing the time required to write all data successfully.

Fortunately it was determined that this was not a major setback since only the summary data file which was written last before losing power was affected. Since the summary data file (`log_short`) was based on information calculated from the parameters stored in the full data file (`log_full`), it was determined that these missing data points could be reconstructed by processing the data in the full data file.

Another limitation to the design of the data logger which was identified was that if an actuator had stopped and was then started again, only the last movement of that actuator would be recorded to the summary data file. This means that the average voltage and current values written to the summary data file for that operation would only include data from the second movement of the actuator. While this issue was known about before the data loggers were deployed, it was decided that since the average voltage and current were not critical parameters to be recorded, fixing the issue was not of a high priority. Due to time constraints on finishing the design it was decided not to investigate implementing a solution.

One fault which was quickly identified while reviewing the data logger software was a code error in the rear actuator finite state machine. Upon transitioning from the "limit switch activated" state to the "actuator running" state the software should reset a counter (`limit_count_r`) which is used to ensure that the three sampling periods have passed before declaring that the actuator has hit one of its limit switches. In the rear actuator finite state machine the line of code which resets this variable was missing entirely due to an oversight when duplicating the front and rear actuator state machines. As a result, if the rear actuator was to transition from the "limit switch activated" state to the "actuator running" state the software would no longer wait three sampling periods for any further suspected limit switch activations. This error manifested itself as excessive limit switch activations for the rear actuator in the final data. Fortunately however, since the limit switch activation was determined entirely by analysing the voltage and current data collected, it was possible to reconstruct the rear limit switch activation counter by processing the voltage and current data with a MATLAB script.

6.4 Data Analysis

After receiving the data loggers back from the hospitals where the testing was performed, the data stored on the micro-SD cards was able to be transferred onto a computer for analysis. Upon initially reviewing the data gathered, the data collection faults outlined in the previous section were identified. In order to correct for these problems with the data, it was decided to write a script to import the CSV data into MATLAB and perform the necessary processing required to correct the data files. While the raw data from the data loggers could have been imported directly into MATLAB, it was identified that in order to simplify the process of working with the time and date information recorded from the real time clock, the time and date string (of the format HH:MM:SS DD/MM/YY) would need to be converted into two purely numerical variables, one for time and one for date (of the format HHMMSS and YYYYMMDD). It was decided to perform the time and date conversion in Microsoft Excel due to the ease of manipulating arbitrary strings with built in functions. After importing the raw data into Excel and performing the date and time conversion the modified data was saved to a new CSV file ready to be imported into MATLAB. This process was performed individually for each data logger.

The MATLAB script used to process the raw data performed the following functions and can be seen in full in Appendix E

Fix invalid temperature measurements. If the data logger read the temperature from the DS1631 before the first temperature conversion was complete an invalid value of 196 would be returned and recorded. In order to perform proper analysis of the temperature data these invalid values need to be removed. The MATLAB script fixes this by replacing invalid temperature measurements with the next valid ambient temperature measurement from the same operation.

Regenerate the summary data file. Since there were a number of issues with the data written to the summary data file the entire summary data file will be regenerated from the full data file (bar the average voltage and current). In addition to regenerating the summary data file this section of the script also fixes the invalid limit switch activation counts in the full data file.

Calculate the total usage time per day. This section of the MATLAB script calculates the total usage time per day for each actuator individually. This data will be plotted with the number of operations per day to obtain an idea of the usage of the system in terms of number of uses and usage duration.

Generate histogram data for operation duration. In order to understand the distribution of operation duration over the total data collection timeframe, a histogram was created to show how many operations fell into different duration categories. Since the categories chosen to display this were not uniform it was not possible to use the built in histogram functionality in MATLAB and instead the data was processed manually.

Calculate split between 12V and 24V modes and average voltage and current. To gain an understanding of how the actuators were being driven it was decided to work out the percentage split of time spent in 12V and 24V operating modes. This section of the MATLAB script also calculates the average voltage and current for both 12V and 24V modes to gain an overall understanding of electrical operating characteristics of the actuators.

The output of the MATLAB script was a number of CSV files containing the correct full and summary data files as well as the usage time per day and operation duration histogram data.

After processing the data obtained from each data logger with the MATLAB script, the resulting CSV files were once again imported into Excel for analysis and graphing. It was decided to use Excel for graphing the data since for some of the data being graphed the horizontal axis contained non-contiguous data, which can be difficult to plot as desired in MATLAB. The discussion of the results obtained will be detailed in the next section.

6.5 Results Discussion

6.5.1 Mansell Power Lifter Usage

The usage patterns of the Mansell Power Lifter, both in terms of total usage time and usage characteristics, have been a key aspect of the data collection and analysis process throughout this project. Gaining an understanding of how the Mansell Power Lifters are being used will assist in determining if the actuator failures are the result of the way in which the systems are used. In order to determine how the Mansell Power Lifters which data was collected from are being used, a timeline of number of operations per day, as well as total run time per day, was created for each system which was tested. The usage timelines for both Mansell Power Lifters can be seen in Figures 6.1 and 6.2 respectively. The number of operations per day is shown in the bar graph corresponding to the left vertical axis and the run time for the actuators are shown in the line graphs corresponding to the right vertical axis. In each of these figures the run time has been shown for each individual actuator to show whether or not there is a disparity in their usage time.

As can be seen from the usage timelines, both of the Mansell Power Lifter systems in question saw a significant number of operations on a regular basis. There were some differences in that the Mansell Power Lifter that data logger #1 was connected to, saw some periods of a couple of days or more where the system was not used at all, while the other system reported significant use on an almost daily basis. On average, the Mansell Power Lifters were being subjected to 28 operations per day with some days showing significantly higher use and an average actuator run time per day of 123 seconds.

In terms of actuator run time, the Mansell Power Lifter that data logger #1 was connected to showed very similar run times per day for both actuators. On the other hand, the results for data logger #2 showed that the rear actuator was seeing significantly more run time than the front actuator on a near-daily basis. For several days during the test period the increased run time of the rear actuator over the front actuator for data logger #2 was greater than 100 seconds. One possible explanation for this disparity in run time for data logger #2 is that the internal clutch in the actuator may be slipping when it reaches the end of its travel, resulting in the limit switch not being activated, and therefore the actuator continuing to run until the user releases the control button. Wenross Holdings Pty. Ltd. has identified that over time, and through use, it is possible for the actuator clutch to begin slipping. While a slipping clutch in a linear actuator does not normally affect the movement ability of the Mansell Power Lifter it can cause the actuator to continue to run rather than activating its limit switch at the end of its travel.

To achieve a better understanding of the distribution of the number of operations per day for the Mansell Power Lifters tested, a combined histogram for the number of operations per day for both data loggers was created and can be seen in Figure 6.3. The histogram shows that the majority of the days during the testing period saw a total number of operations in the tens of operations per day. The distribution of operations per day can be seen to have a long tail with a number of outliers in the high tens of operations per day and a couple of extreme outliers above in the low hundreds of operations per day.

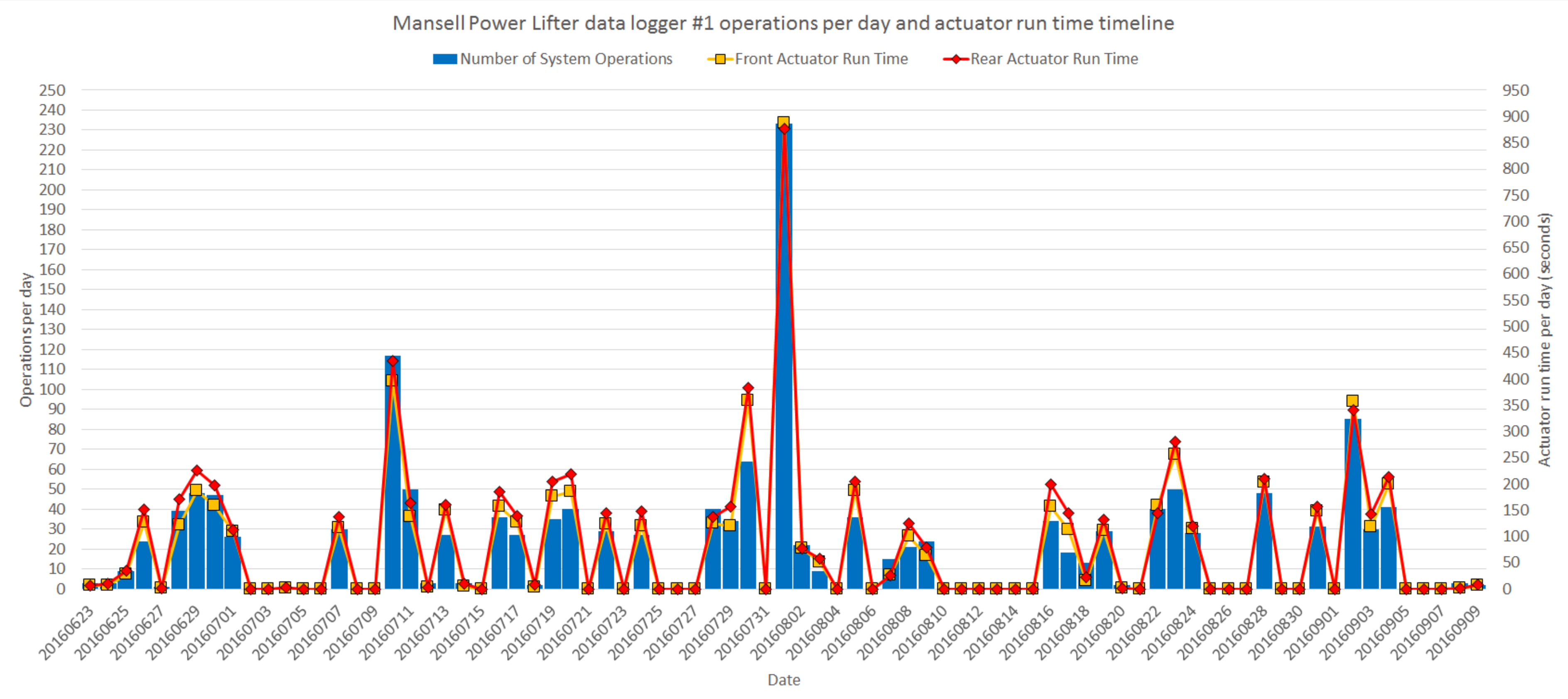


Figure 6.1: Mansell Power Lifter data logger #1 usage timeline showing operations per day and run time per day per actuator. (Source: author)

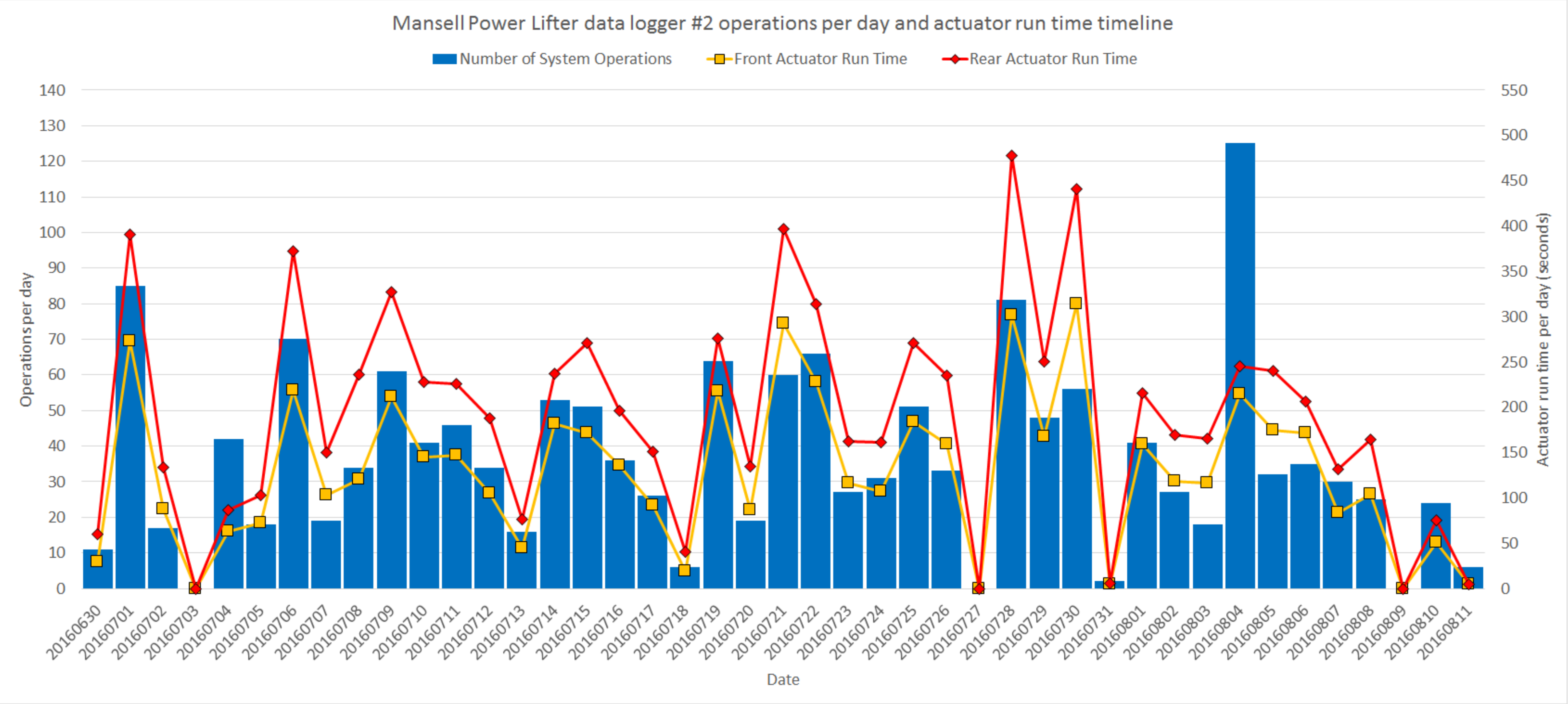


Figure 6.2: Mansell Power Lifter data logger #2 usage timeline showing operations per day and run time per day per actuator. (Source: author)

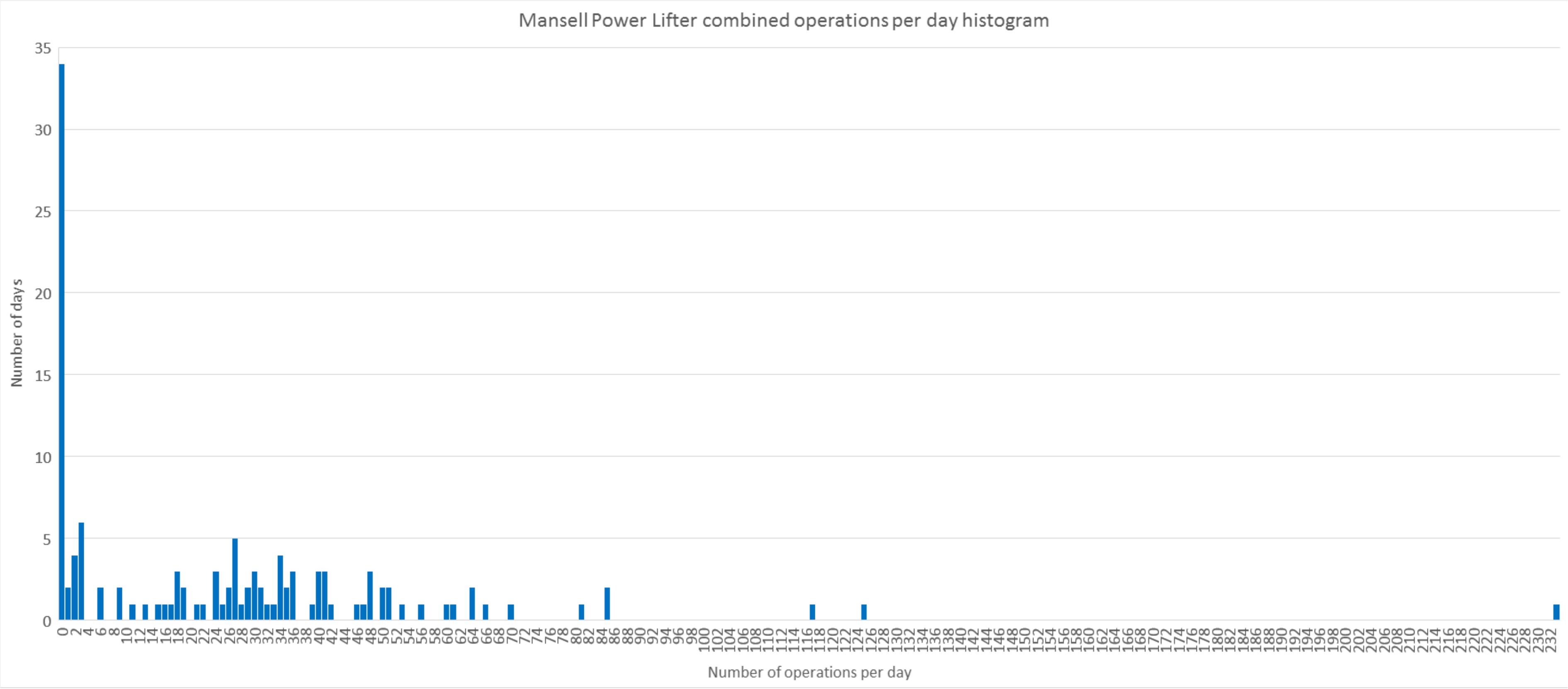


Figure 6.3: Mansell Power Lifter combined usage histogram for both data loggers showing number of operations per day. (Source: author)

While the number of operations per day for the Mansell Power Lifters is now known this does not give the full story of how they are being used, since the duration of each of these operations is not known. In order to understand the distribution of operation duration for the Mansell Power Lifters in question over the testing period, a histogram was created. The operation times from the summary data for both of the data loggers were grouped into a number of bins for both the front and rear actuator, and the histogram plotted.

The bins used for the histogram were for the most part in 5 second intervals with the exception of the first bin which contained operations shorter than 1 second and the final bin which represented operations between 40 and 100 seconds. It was chosen to have a smaller first bin since analysis of the data showed that there were a large number of short operations with the average duration for all operations recorded only being 4 seconds. Additionally, it is unlikely that there will be any operations over 40 seconds, since a full raise or lower operation typically only takes about 30 seconds, and therefore a large bin was used to catch any very long operations. Due to the disparity in actuator run time per day observed in Figure 6.2 it was decided to display the front and rear actuator operation time distribution separately to determine if there was also a disparity in actuator operation time. The histogram for the run time per operation can be seen in Figure 6.4.

The histogram confirms the initial findings that the Mansell Power Lifter linear actuators are being subjected to mostly short operations. There were a significant number of operations recorded as lasting less than 1 second, making up approximately 46% of the total number of operations. The next highest bin was the 1 second to 5 second operation duration category which accounted for a further 26% of the total operations. The remaining operation duration categories decayed away until a maximum of 25 seconds and there was one outlier in the less than 35 seconds category. A slight disparity in the operation duration between front and rear actuators could be seen as predicted by the run time per day data, showing a tendency towards higher operation times for the rear actuator. Overall it appears that the actuators in the Mansell Power Lifters are being subjected to a large number of primarily very short operations on a regular basis.

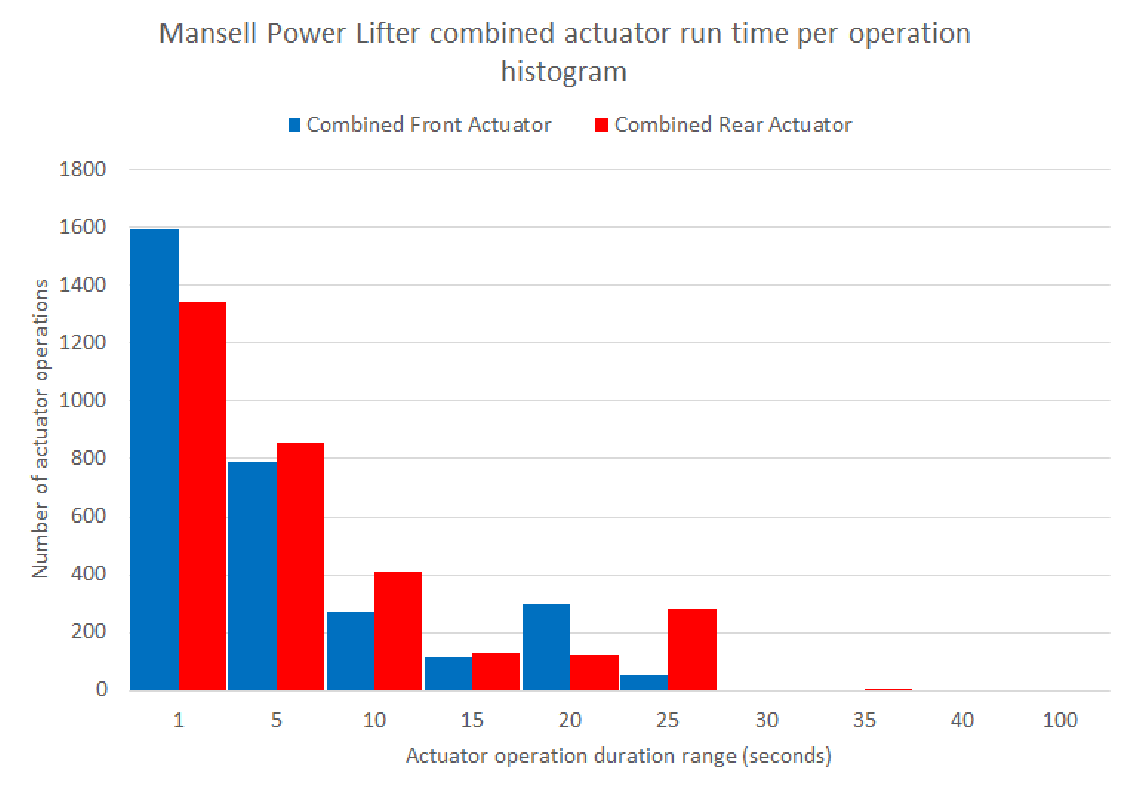


Figure 6.4: Mansell Power Lifter combined operation duration histogram for both data loggers. Unequal bins have been used to provide greater detail in specific areas of interest. (Source: author)

6.5.2 Limit Switches

The primary suspected failure mechanism for the linear actuators in the Mansell Power Lifter at the commencement of this project were the internal limit switches in the linear actuators. In order to assist in determining if they were in fact the failure mechanism, the number of limit switch activations for each linear actuator over the testing period was recorded. For data logger #1 the front actuator had 198 limit switch activations and the rear only 2, and in the case of data logger #2 the front actuator had 140 limit switch activations and the rear 76.

Since the Mansell Power Lifter has an external limit switch for one of the limits of the rear actuator, which cuts power to the actuator when the system is raised to full height, it is to be expected that the rear actuator limit switches will be activated only half as many times as the front, since only one of the rear internal limit switches will ever be activated. This matches with the data obtained from data logger #2 where the rear actuator had roughly half of the limit switch activations as the front. However, for the Mansell Power Lifter connected to data logger #1, the rear actuator was only recorded as hitting its limits on two occasions. This could either be due to another case of the clutch in the actuator slipping when it reaches the end of its travel, and thus preventing the limit switch from being activated, or it could be due to the safety sensor being out of calibration and cutting power to the actuators before they reach their limits, as a result of falsely detecting the ground as an object underneath the system. (Grant-Thomson 2014)

Despite the inconsistencies in the rear actuator limit switch activations it can be seen that overall the limit switches in the linear actuators do not appear to be experiencing excessive use. Taking the number of limit switch activations of the front actuator as an example, the combined average number of front limit switch activations per day was only 2.9. Considering that every time a Mansell Power Lifter is loaded into a vehicle for transport both legs need to fully retract for loading and then fully extend for unloading, it is not unreasonable to expect several full extend or retract operations per day resulting in several limit switch activations per day.

Table 6.1: Mansell Power Lifter data logger #1 electrical operating characteristics observed for both 12V and 24V operation modes.

Data logger #1	**12V Mode**	**24V Mode**
Percentage of run time (%)	23.43	76.57
Average actuator voltage (V)	12.020	23.212
Average actuator current (A)	3.311	3.916

Table 6.2: Mansell Power Lifter data logger #2 electrical operating characteristics observed for both 12V and 24V operation modes.

Data logger #2	**12V Mode**	**24V Mode**
Percentage of run time (%)	19.69	80.31
Average actuator voltage (V)	12.192	23.931
Average actuator current (A)	3.793	4.084

6.5.3 Electrical Characteristics

Another group of operational characteristics of the Mansell Power Lifter which are important to consider when looking for factors contributing to the failure of the linear actuators are the electrical operating characteristics of the actuators. Analysis of the voltage and current data gathered for the actuators, can be used to identify if the actuators are being excessively loaded, or if they are receiving incorrect supply voltages. The full data obtained from each data logger was processed using the MATLAB script, in order to calculate the percentage split between 12V and 24V operating modes for each Mansell Power Lifter tested. Additionally, for each operating mode the average voltage and current was calculated. The results of this data processing can be seen in Tables 6.1 and 6.2 for data loggers #1 and #2 respectively.

As can be seen from the results obtained from the data, both Mansell Power Lifters spent approximately 20 percent of their operating time in 12V operating mode and 80 percent in the 24V high speed mode. This means that while, as outlined earlier, the majority of operations are very short, those operations that do have a longer duration are spending the majority of their time in 24V mode. Since the Mansell Power Lifter will only switch to 24V mode if the load on the system is low enough, this result suggests that the system is changing into 24V mode the majority of the time and therefore it is unlikely that the actuators are experiencing excessive loads.

Analysis of the average actuator voltage for both data loggers shows that the voltage being supplied to the linear actuators in each operational mode were close to the expected values. Similarly, after examining the average current for the actuators connected to both data loggers the current observed in both 12V and 24V mode was on average quite low and well within the expected range. The relatively low average current in each mode is further evidence that generally speaking the actuators in the Mansell Power Lifters are not being subjected to excessive loads.

6.5.4 Temperature Factors

In the initial analysis of information known about the linear actuator failures in the Mansell Power Lifter it was identified that in one failure case the ambient temperature at the time of the failure was relatively high. By recording the ambient temperature in which the actuators are operating it was hoped that any regular temperature extremes experienced by the systems could be identified. Additionally, had there been an actual actuator failure during the testing period it would be possible to review the temperature data leading up to the failure.

After analysing the ambient temperature data from both data loggers it was found that the maximum ambient temperature that the systems were exposed to was 31°C (the data logger sent to Brisbane), the minimum was 14.5°C (the data logger sent to Melbourne), and the overall average for both systems was 22.5°C. A full timeline of the ambient temperature data for each data point recorded was also created in order to see the trend of what temperatures the systems were exposed to over the testing period. The full timeline is not included here but can be seen in Figures E.2 and E.3 in Appendix E. It shows that the majority of the time the ambient temperature was relatively close to the average and within the range of approximately 17°C to 25°C. Overall, the ambient temperatures subjected to the linear actuators were not excessively high or low and was always well within the rated operating range of 5°C to 40°C (Linak 2015).

6.5.5 Limitations of Data

While overall, the collection of operational data from two Mansell Power Lifters in the field was a success, there are some limitations present in the useability of the data in relation to the linear actuator failures. Firstly, during the testing period none of the linear actuators in the Mansell Power Lifters tested were reported as failing or having any faults. As a result, despite the quantity and quality of the data gathered, none of it can be directly correlated with actual monitoring data from an actuator failure. This was not entirely unexpected, since information about past actuator failures only shows approximately one reported actuator failure each year, out of the 100 estimated Mansell Power Lifters currently in use. In order to have a higher chance of capturing data relating to an actual actuator failure, a larger number of data loggers would need to be deployed for a much longer amount of time. Instead the data collected can only be analysed and interpreted, in order to identify usage trends and operational characteristics which might point towards possible causes of failure.

Another limiting factor which needs to be considered when analysing the data collected, is the timing and location of the Mansell Power Lifters selected for testing. Since the data collection period took place during the Winter months it is likely that the ambient temperatures recorded will be lower than what the systems might experience in the middle of Summer. Additionally, different geographical locations may have different ambient temperature profiles throughout the year. Also, since known high usage locations were chosen as a worst case scenario for the data collection to be performed, some of the data obtained is likely not a good representation of the usage characteristics of much lower usage locations. Although, the existing failure information does in fact show that the majority of failures were from high usage locations, and information from these types of locations is therefore more relevant for identifying failures.

6.5.6 Suspected Causes

Based on the current analysis of the data collected from the Mansell Power Lifters in the field, and largely due to the lack of any direct evidence of a failure, it is not possible to confirm the exact mechanism or cause of the linear actuator failures. However, the data gathered has provided some valuable insights into the usage and operational characteristics of the Mansell Power Lifters. The electrical characteristics of the operation of the actuators has been shown to be as expected and within specification, the ambient temperature at which the systems are operating in is not unreasonably high or low and is also within specification, and the number of limit switch activations is well within the reasonable range for normal use.

While much of the data analysis showed that the linear actuators were operating normally, one area of analysis which returned results of interest was the actuator usage count and time. Though the total usage time of each actuator was not unreasonably long, the fact that there were a very large number of short operations, 46% of which lasted less than a single second, it was theorised that these short and often jerky movements could be causing unexpected mechanical shock and vibration inside the actuator mechanisms. The initial movement of the linear actuators can often be sharp due to the initial movement acceleration. Additionally, if the Mansell Power Lifter has been running in 24V mode within the last 2 seconds or so, the actuators may start up in 24V mode, causing the initial movement to be significantly jerkier than normal. While there is not enough evidence yet to confirm this theory it is possible that these short and sharp movements could be causing the limit switches to activate or jam when the actuators are not at the end of their travel.

6.6 Potential Solutions Proposal

Since it was not possible to identify the exact mechanism or cause of the actuator failures on the Mansell Power Lifter it will not be possible to propose a complete solution to the problem. However, based on the data obtained from the real world testing and analysis of the Mansell Power Lifter design it has been possible to identify a number of options for minimising the number of actuator failures experienced. Firstly, if further testing and analysis of real world usage data were to be conducted it may be possible to determine how much usage it would take for an actuator to become prone to failure. The maintenance guidelines for the Mansell Power Lifter could then be amended to recommend the replacement of linear actuators after they have experienced a certain amount of use. A simple usage counter could be developed and fitted to the Mansell Power Lifters in the field to assist in determining the amount of use that each actuator has seen.

An additional potential solution that could be investigated, after further testing and analysis, would be to modify the control system of the Mansell Power Lifter. If the control system could identify the signs of an impending actuator failure it would be possible to warn the users and prompt an actuator replacement. Alternatively if a failure were to occur where the limit switches were jammed it may be possible to initiate a sequence of movements which could clear the jam condition. While significant investigation and testing would need to be undertaken to develop these solutions, their actual implementation would not be unreasonably difficult, since previous work by Keys in 2014 has taken significant steps towards making the Mansell Power Lifter control system fully digital with a programmable microcontroller.

6.7 Chapter Conclusion

This chapter has detailed the real world data collection and testing which was performed on the Mansell Power Lifters, presented and analysed the data collected, and provided a discussion on the implications of these results. A number of preliminary potential solutions were also outlined. Despite the large quantity of testing data collected, there were unfortunately no actuator failures observed and therefore none of the data could be directly correlated with an actual actuator failure. Despite this the analysis of the data collected has provided valuable insights into the real world usage and operational characteristics of the Mansell Power Lifters and their associated linear actuators.

The main operational characteristic of note that was identified from the data was that the linear actuators are being subjected to a very large number of short movements. Currently the primary theory for the suspected cause of the actuator failures is that the large number of short sharp movements could be causing the internal limit switches in the linear actuators to jam. The failure mode of which has been suspected in a number of cases in the past. However, further testing and analysis would be required to confirm this theory and also to validate and develop any of the proposed potential solutions.

Chapter 7

Conclusions and Further Work

7.1 Conclusions

This dissertation has sought to investigate the usage of the Mansell Power Lifter as a part of the Mansell Infant Retrieval system, in order to identify the mechanism and cause of failures which have been reported as occurring with the Linak LA34 linear actuators used in the system. For the purposes of understanding the background information relating to the problems, Chapter 2 provided an overview of the design and operation of the Mansell Power Lifter and outlined the existing information available relating to the actuator failures. Chapter 2 concluded by providing a discussion of the overall requirements of the testing procedure and system which would need to be developed in order to assist in analysing the failures.

In order to initiate the design and development of the data collection system, Chapter 3 provided a discussion of possible data collection methodologies and weighed the merits of stress testing and collection of real world data. Of the two main methodologies discussed it was decided that the collection of real world usage data through the use of a data logger device would be the most appropriate methodology to use, given the currently known information. The hardware and software design considerations relating to the autonomous data collection device were outlined. It was decided that the data collection device would be powered and obtain all required information directly from the linear actuator power cables. It would contain a variety of sensors, be controlled by an Arduino microcontroller development board, and store data locally onto a micro-SD card.

It was also identified that the most appropriate software program structure for the device would be based on Finite State Machines.

The hardware and software design of the data collection device were discussed in Chapters 4 and 5. In terms of the hardware design, a dual input, polarity independent power supply was designed to power the device entirely off of the actuator power cables, and an Arduino Nano featuring the ATMega328P microcontroller was chosen to control the device. The selection of the various sensors outlined in the design requirements was described, including the design of two voltage sensing circuits, and the addition of ACS712-30A Hall-Effect current sensors, a DS1631 temperature sensor, and DS3231 real time clock module. Overall the hardware design of the data collection device was based around the design goal of preventing any impact on the normal operation of the Mansell Power Lifter. This design philosophy was also applied to the physical construction of the device which was designed to fit entirely within a small enclosure that can be neatly attached, out of the way, to the underside of the Mansell Power Lifter.

In terms of the software design of the data collection device, the main software program was designed as a series of nested Finite State Machines, which would keep track of the operational state of each of the linear actuators as well as the entire system. Several of the key software functions used in the data collection device's software were outlined, and the use of the microcontroller's built in timers and interrupts to ensure consistent timing was discussed.

Once the data collection devices had been fully designed and developed they were deployed to two Mansell Power Lifters in hospitals in Brisbane and Melbourne where the Mansell Infant Retrieval Systems see regular use. Between the two systems tested a total of 122 days of normal operational data was successfully recorded by the data collection devices and returned for analysis. During the testing period there were no reported occurrences of actuator failures on either of the Mansell Power Lifters tested. The initial processing and analysis of the data obtained was performed in MATLAB where a small number of problems which were identified with the raw data were resolved.

Analysis of the data obtained showed that the Mansell Power Lifters were being subjected to a relatively large number of operations each day. While an average of 28 operations per day were recorded, there were a number of days with over 100 operations. In terms of the run time of the linear actuators, there were some discrepancies identified between the time that the front and rear actuators spent running, however the overall operational time was not excessively long, with an average of 123 seconds per actuator per day. The internal limit switches in the linear actuators were one potential failure mechanism which was identified early on in the project and therefore the total number of limit switch activations was recorded during the testing. However, upon analysis of the data a relatively low average of 2.9 limit switch activations per day was identified for the front actuator, which sees the most limit switch use. The electrical operating characteristics of the actuators and the ambient temperatures in which they were operating were also analysed, however neither of these operational areas contained data which was out of the norm.

One aspect of the data which was of particular interest however was the duration of the system operations. Upon analysis of the duration of system operations it was identified that the majority of the operations of the Mansell Power Lifter and its linear actuators were very short, with 46% of operations lasting less than 1 second. Combined with the finding that the Mansell Power Lifters were regularly seeing a large number of operations it was concluded that the linear actuators were being subjected to a large number of very short and potentially jerky movements on a regular basis. While further testing and analysis would be required to confirm any theories for the cause of the failures, due to the fact that no actual failure data was obtained, the current theory is that the large number of short sharp movements could be causing the limit switches in the linear actuators to jam and stopping the actuators from moving.

Given that the exact cause of the linear actuator failures in the Mansell Power Lifter cannot be confirmed at this point in time given the data collected, it has not been possible to propose a definitive solution to the problem. However, a number of potential solution concepts were identified that could be investigated further through the collection of additional data relating to the failures. These include, determining appropriate lifetimes for the linear actuators and revising maintenance guidelines to recommend actuator replacement as the likelihood of failures increases, and implementing new functionality to the Mansell Power Lifter control system to actively monitor for failures and provide a warning in advance, or to actively prevent failures by controlling the way in which the actuators are driven.

7.2 Further Work

While this project has taken significant steps towards understanding the failures of the linear actuators in the Mansell Power Lifter, further work will need to be conducted in order to fully understand the cause and characteristics of these failures and eventually propose and implement a viable solution. Firstly, since there were no linear actuator failures reported during the testing period and the failure rate identified from the past records is quite low, there is scope for the data loggers to be deployed to Mansell Power Lifters for a much longer period of time, to increase the chance of actually recording data pertaining to a failure. The collection of data over a longer period would also provide a better picture of the usage of the systems over time, particularly in terms of the ambient operating temperature which is likely to fluctuate quite significantly between seasons.

Another potential area for further testing to be conducted in the future is for in depth stress testing to be performed. While stress testing of a test Mansell Power Lifter and associated actuators was initially considered for this project, it was concluded that at least initially the collection of real world usage data would prove to be more valuable. However, given the low chance of actually observing an actuator failure in the real world, future work on this problem could consider the design and execution of a stress test in a controlled environment in order to increase the likelihood of capturing data pertaining to the characteristics of a failure.

If the further testing of the Mansell Power Lifter was to result in sufficient data relating to the failures being obtained, such that the mechanism and cause of the actuator failures could be positively identified, the development of the potential solutions outlined in Chapter 6 could be pursued. The Mansell Power Lifter maintenance guidelines could be revised by identifying reasonable operating lifetimes for the linear actuators and recommending actuator replacement as it becomes more likely that they will fail. Alternatively, the redesign of the Mansell Power Lifter control system to contain programmable digital control as proposed by Keys (2014), could be extended to include the ability for the control system to detect potential upcoming actuator failures and warn about them, to modify the drive characteristics of the actuators to prevent failures, or in the case that the limit switches are jamming to move the actuator in such a way as to clear the jam condition automatically.

References

Allegro Microsystems LLC (2012), Acs712 - fully integrated, hall effect-based linear current sensor ic with 2.1 kvrms isolation and a low-resistance current conductor, datasheet, Allegro Microsystems LLC.

Arduino (2016*a*), 'Arduino nano', `https://www.arduino.cc/en/Main/ArduinoBoardNano`. viewed 12 September 2016.

Arduino (2016*b*), 'Using an Arduino as an AVR ISP (In-System Programmer)', `https://www.arduino.cc/en/Tutorial/ArduinoISP`.

Atmel (2008), AVR910: In-System Programming, application note, Atmel.

Atmel (2009), 8-bit microcontroller with 4/8/16/32k bytes in-system programmable flash atmega48pa/atmega88pa/atmega168pa/atmega328p, datasheet, Atmel.

Backes, J. (2007), 'A Practical Guide to IEC 60601-1', Rigel Medical, `http://www.rigelmedical.com/downloads/Rigel-Medical-A-Practical-guide-to-IEC-60601-1.pdf`. viewed 16 March 2016.

Balaban, E., Bansal, P., Stoelting, P., Saxena, A., Goebel, K. F. & Curran, S. (2009), A diagnostic approach for electro-mechanical actuators in aerospace systems, *in* '2009 IEEE Aerospace conference', pp. 1–13.

Birolini, A. (2010), *Reliability Engineering: Theory and Practice*, Springer-Verlag Berlin Heidelberg.

Bodden, D. S., Clements, N. S., Schley, B. & Jenney, G. (2007), Seeded failure testing and analysis of an electro-mechanical actuator, *in* '2007 IEEE Aerospace Conference', pp. 1–8.

Bowman, M. (2013), Recommended usage of microchip serial rtcc devices, application note, Microchip.

Boxall, J. (2014), 'Tutorial using ds1307 and ds3231 real-time clock modules with arduino', `http://tronixstuff.com/2014/12/01/tutorial-using-ds1307-and-ds3231-real-time-clock-modules-with-arduino/`.

Chan, H. A. & Englert, P. J. (2001), *Principles of Stress Testing*, Wiley-IEEE Press.

Davide Giacomini, L. C. (2008), A nove high efficient approach to input bridges, *in* 'PCIM Europe'.

Evans, B. W. (2007), Arduino programming notebook, manual, Arduino.

Fairchild Semiconductor (2014*a*), 1N4001 - 1N4007 General-Purpose Rectifiers, datasheet, Fairchild Semiconductor.

Fairchild Semiconductor (2014*b*), LM78XX / LM78XXA 3-terminal 1 A positive voltage regulator, datasheet, Fairchild Semiconductor.

Fernando, A. (2011), 'Software verification and validation the role of iec 60601-1', *Biomedical Instrumentation and Technology* **45**(5), 370–373.

Grant-Thomson, J. (2014), *Mansell Power Lifter Operator's Manual*, Wenross Holdings Pty Ltd.

Holoubek, L. (2013), Watt-hour meter based on the stm32f101 microcontroller, application note AN3322, STMicroelectronics.

Ibrahim, D. (2010), *SD Card Projects Using the Pic Microcontroller*, Elsevier.

International Electrotechnical Commission (2012), IEC 60601-1, Ed. 3.1, International Electrotechnical Commission.

Ismail, M. A. A., Balaban, E. & Spangenberg, H. (2016), Fault detection and classification for flight control electromechanical actuators, *in* '2016 IEEE Aerospace Conference', pp. 1–10.

Johnson, B. W. & Aylor, J. H. (1988), Reliability and safety analysis in medical applications of computer technology, *in* 'Engineering of Computer-Based Medical Systems, 1988., Proceedings of the Symposium on the', pp. 96–100.

Kapur, K. C. & Pecht, M. (2014), *Reliability Engineering*, Wiley, Hoboken, New Jersey.

Keys, C. (2014), Mansell power lifter control system redesign. BEng dissertation, University of Southern Queensland.

Laplante, P. A. (2004), *Real-Time Systems Design and Analysis*, 3rd edn edn, IEEE Press.

Linak (2015), Actuator LA34, datasheet, Linak.

Marcus, M. L. & Biersach, B. R. (2003), 'Regulatory requirements for medical equipment', *IEEE Instrumentation Measurement Magazine* **6**(4), 23–29.

Maxim Integrated (2004), Logic-level translation, application note 3007, Maxim Integrated.

Maxim Integrated (2015*a*), DS1631/DS1631A/DS1731 High-Precision Digital Thermometer and Thermostat, datasheet, Maxim Integrated.

Maxim Integrated (2015*b*), DS3231 Extremely Accurate I2C-Integrated RTC/TCXO/Crystal, datasheet, Maxim Integrated.

McLinn, J. A. (2016), Understanding accelerated degradation, *in* '2016 Annual Reliability and Maintainability Symposium (RAMS)', pp. 1–8.

National Instruments (2013), Power electronics fundamentals - ac to dc power - rectifiers, white paper, National Instruments.

Oxford University Press (2016), 'Definition of reliable in english', `https://en.oxforddictionaries.com/definition/reliable`.

Smith, K. & Miller, J. (2013), Obdii data logger design for large-scale deployments, *in* '16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)', pp. 670–674.

Standards Australia (1999), Risk Management, AS/NZ 4360:1999, Standards Australia.

Suzdalenko, A. (2011), Guidelines for autonomous data logger design, *in* '2011 IEEE International Symposium on Industrial Electronics', pp. 1426–1429.

Suzdalenko, A., Lazdans, A. & Galkin, I. (2012), Development of multi-channel analogue signal data logger based on msp430, *in* 'Education and Research Conference (EDERC), 2012 5th European DSP', pp. 40–43.

Taiwan Semiconductor (n.d.), KBP201G - KBP207G Single Phase 2.0 Amps Glass Passivated Bridge Rectifiers, datasheet, Taiwan Semiconductor.

University of Southern Queensland (2016), 'Risk management plan', `http://www.usq.edu.au/hr/healthsafe/resources/usqsafe`.

Vergara, A. L. & Villaruz, H. M. (2014), Development of an arduino-based automated household utility power monitoring system, *in* 'Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2014 International Conference on', pp. 1–6.

Verma, A. K., Srividya, A. & Karanki, D. R. (2010), *Reliability and Safety Engineering*, Springer-Verlag London.

Yarborough, B. (2015), Components and methods for current measurement, technical note 30304, Vishay Dale.

Zhang, H. J. (2013), Basic concepts of linear regulator and switching mode power supplies, application note, Linear Technology Corporation.

Zhanyong, R., Yueqin, W. & Xuegang, L. (2014), Reliability design and analysis method under digital development environment, *in* 'Prognostics and System Health Management Conference (PHM-2014 Hunan), 2014', pp. 441–445.

Appendix A

Project Specification

ENG 4111/2 Research Project

Project Specification

For:	**Donald Bailey**
Topic:	Mansell Power Lifter Actuator Failure Analysis
Supervisors:	Dr John Grant-Thomson Dr John Leis
Sponsorship:	Wenross Holdings Pty Ltd
Project Aim:	To identify the cause of recurring failures of DC-motor-driven linear actuators used to raise and lower the Mansell Power Lifter and propose possible solutions to this problem.

Program: **Issue A, March 16, 2016**

1. Research reliability engineering standards and techniques, and the characteristics and failure modes of DC motor systems and their controls.
2. Perform an analysis of the design of the system and the known symptoms and characteristics of actuator failures, in order to identify parameters to monitor during failure data acquisition.
3. Review and determine appropriate technologies for testing actuator failure modes and methods for gathering further data for subsequent analysis.
4. Design and implement a testing system which can be deployed to a working system in the field and used in laboratory testing, to gather data pertaining to the state and operation of the actuator when a failure occurs.
5. Analyse data gathered to identify the cause of the failure and propose potential solutions for preventing failures or rectifying failures when they occur.

As time and resources permit:

1. Develop the proposed solution for actuator failures into a hardware prototype.
2. Develop required hardware to integrate the proposed solution into the exiting Mansell Power Lifter hardware.

Agreed:

Student Name: Donald Bailey
Date: 08/03/2016

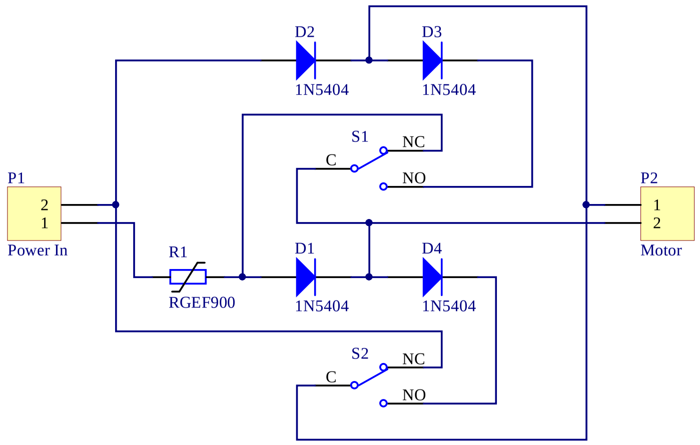
Supervisor Name: Dr John Leis
Date: 08/03/2016

Supervisor Name: Dr John Grant-Thomson
Date 08/03/2016

Appendix B

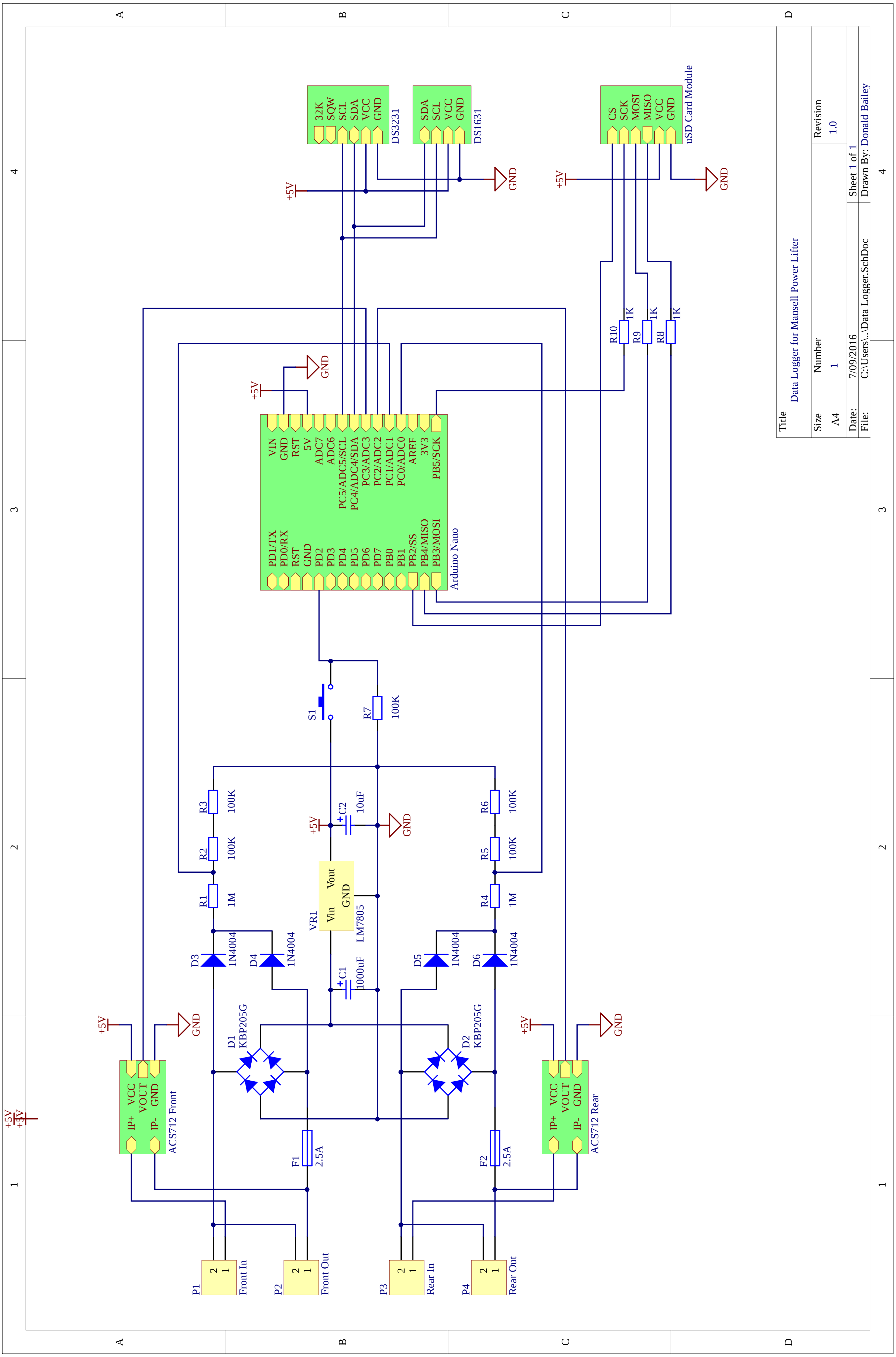
Circuit Schematics

B.1 Linak LA34 Internal Limit Switch Circuit Schematic



Title		
Linak LA34 - Limit Switch PCB		
Size A4	Number 1	Revision 1.0
Date: 7/08/2016		Sheet 1 of 1
File: C:\Users\..\Linak_LA34_PCB.SchDoc		Drawn By: Donald Bailey

B.2 Data Logger Circuit Schematic


Title
Data Logger for Mansell Power Lifter
Size
A4
Number
1
Revision
1.0
Date: 7/09/2016
Sheet 1 of 1
File: C:\Users\..\Data Logger.SchDoc
Drawn By: Donald Bailey
A
B
C
D
1
2
3
4
DS3231
32K
SQW
SCL
SDA
VCC
GND
DS1631
SDA
SCL
VCC
GND
uSD Card Module
CS
SCK
MOSI
MISO
VCC
GND
Arduino Nano
VIN
GND
RST
5V
ADC7
ADC6
PC5/ADC5/SCL
PC4/ADC4/SDA
PC3/ADC3
PC2/ADC2
PC1/ADC1
PC0/ADC0
AREF
3V3
PB5/SCK
PD1/TX
PD0/RX
RST
GND
PD2
PD3
PD4
PD5
PD6
PD7
PB0
PB1
PB2/SS
PB4/MISO
PB3/MOSI
R10 1K
R9 1K
R8 1K
S1
R7 100K
R3 100K
R2 100K
R1 1M
R6 100K
R5 100K
R4 1M
C2 10uF
C1 1000uF
VR1 LM7805
Vin
Vout
GND
D3 1N4004
D4 1N4004
D5 1N4004
D6 1N4004
D1 KBP205G
D2 KBP205G
F1 2.5A
F2 2.5A
ACS712 Front
ACS712 Rear
IP+
IP-
VCC
VOUT
GND
P1 Front In
P2 Front Out
P3 Rear In
P4 Rear Out
1
2
+5V
GND

Appendix C

Data Logger Software

C.1 The `Field_Device_v1.5.1.ino` Source Code

The Arduino source code used in the Mansell Power Lifter data logger device.

Listing C.1: The Arduino source code for the data logger.

```
/*-----------------------------------------------------------------
  Field_Device_v1.5.1
  Created by: Donald Bailey
  Last modified: 30/06/2016
  Description: This program retrieves data from a number of
     sensors connected to a Mansell Power Lifter (MPL)
               in order to monitor it's operation. Sensors are
                  polled every 100mS and data written to a
               uSD card in CSV format. Two files are written one
                   containing every data point and one
               containing a summary of each recorded operation
                  of the MPL.

  Notes:       The word ”operation” is used throughout this
      program in relation to the MPL to signify a particular
               instance of the use of the MPL. The start of an
                  operation is when the user first presses one
                  of the
               actuator buttons and the end of an operation is
                  when all actuator buttons have been released
                  for at
               least one sampling period (100mS).
-----------------------------------------------------------------*/


/*-----------------------------------------------------------------
  Major software change features
  - v1.2
    Remove external low level interrupt for power loss since the
        voltage is too low in 12V mode to trigger the interrupt
```

```
  - v1.3
    Add second voltage and current measurement channel

  - v1.4
    Implement counting of limit switch activation

  - v1.5
    Implement waiting several samples to confirm limit switch
        activation and reduce false positives
    Replace some ”magic numbers” with global constants
-----------------------------------------------------------*/

// Libraries used
#include <SPI.h>          // SPI for uSD card
#include <SD.h>           // SD card file system operations
#include <Wire.h>         // I2C communications
#include <avr/eeprom.h> // EEPROM operations

// Pin configuration constants
#define SD_CS_PIN 10        // uSD card chip select pin
#define CLEAR_PIN 2         // External button for clearing
    elapsed time
#define CURR_SENS_PIN_F A3 // Analog input used for front
    actuator current sensor
#define CURR_SENS_PIN_R A2 // Analog input used for rear
    actuator current sensor
#define POW_STATE_PIN_F A1 // Analog input used for front
    actuator voltage measurement
#define POW_STATE_PIN_R A0 // Analog input used for rear
    actuator voltage measurement

// Data address constants
#define DS1631_ADDR 0x4b // I2C address of DS1631 temperature
    sensor
#define DS3231_ADDR 0x68 // I2C address of DS3231 real time
    clock
#define ELAPSED_F_ADDR 0 // EEPROM address for front actuator
    elapsed time
#define ELAPSED_R_ADDR 4 // EEPROM address for rear actuator
    elapsed time
#define LIMIT_F_ADDR 8   // EEPROM address for front actuator
    limit switch activation counter
#define LIMIT_R_ADDR 12  // EEPROM address for rear actuator
    limit switch activation counter

// Enumerated aliases for date/time structures
#define SECOND 0
#define MINUTE 1
#define HOUR 2
#define DAY 3
#define MONTH 4
#define YEAR 5

// Data processing constants
#define INVERT_DIR_F -1        // Correct the current direction
    for the front actuator if necessary
#define INVERT_DIR_R -1        // Correct the current direction
    for the rear actuator if necessary
#define LIMIT_THRESHOLD 3      // Number of samples to wait
    before deciding if a limit has been reached
#define VOLTAGE_THRESHOLD 9    // Voltage at which the actuator
    is considered to be ”on”
#define CURRENT_THRESHOLD 0.5 // Current below which the
    actuator is considered to be ”stopped”
```

```
// EEPROM address as a pointers to allow for writing to memory
float* elapsed_f_addr = (float*)ELAPSED_F_ADDR;
float* elapsed_r_addr = (float*)ELAPSED_R_ADDR;
unsigned long* limit_f_addr = (unsigned long*)LIMIT_F_ADDR;
unsigned long* limit_r_addr = (unsigned long*)LIMIT_R_ADDR;

// Working variables
float current_f, current_r;   // Current current for each
   actuator
float voltage_f, voltage_r;   // Current voltage for each
   actuator
float run_time_f, run_time_r; // Current run time for each
   actuator
float elapsed_f, elapsed_r;   // Total elapsed time for each
   actuator
float interval_f, interval_r; // Number of seconds since last
   measurement for each actuator
float avg_curr_f, avg_curr_r; // Average current for each
   actuator for current operation
float avg_volt_f, avg_volt_r; // Average voltage for each
   actuator for current operation
float temp, tmp;              // Temporary working variables
unsigned long millis_start_f, millis_start_r; // System
   millisecond counter value for each actuator when they are
   powered on
unsigned long millis_last_f, millis_last_r;   // System
   millisecond counter value for each actuator for the last
   sample
unsigned long millis_curr;                    // System
   millisecond counter value for the current time
unsigned long count_f, count_r;               // Counter for the
    number of samples recorded for each actuator
unsigned long limit_f, limit_r;               // Counter for the
    number of limit switch activations for each actuator
int limit_count_f, limit_count_r;             // Counter for the
    current number of samples which are believed to contain
   limit switch activations for each actuator

// Byte array for storing the date and time
byte datetime[6];

// Actuator state variables
bool running_f = false; // Flag for if the front actuator is
   running
bool running_r = false; // Flag for if the rear actuator is
   running
bool limited_f = false; // Flag for if the front actuator has
   hit a limit
bool limited_r = false; // Flag for if the rear actuator has hit
    a limit

// Used by the interrupt service routine to initiate a new
   measurement
volatile bool waiting = false;

/*
 * Program setup operations to be run once upon power up
 */
void setup() {
  pinMode(CLEAR_PIN, INPUT); // Configure pin for elapsed time
      reset button

  // Set up timer 2 to interrupt the processor every 100ms
  cli();                 // Disable interrupts while configuring
```

```
  TCCR1A = 0;           // Clear TCCR1A settings
  TCCR1B = 0b00001101;  // 1024 prescaler, Clear on timer compare
      match (CTC) mode
  TCCR1C = 0;           // Clear TCCR1C settings
  OCR1AH = 1562 >> 8;   // Compare register high byte (value 1562
      ~= 0.1s)
  OCR1AL = 1562;        // Compare register low byte (value 1562
     ~= 0.1s)
  TIMSK1 = 0b00000010;  // Enable interrupt on timer compare
     register (OCA) match
  sei();                // Re-enable interrupts

  Wire.begin();         // Enable I2C communication
  SD.begin(SD_CS_PIN);  // Enable SPI (uSD card) communication

  // Start ambient temperature conversion
  Wire.beginTransmission(DS1631_ADDR); // Start communication
     with the DS1631 temperature sensor
  Wire.write(0xAC);                    // Access the
     configuration register
  Wire.write(0x80);                    // Change configuration
     to 9bit resolution, continuous converstion mode
  Wire.endTransmission();              // Close communication
     session
  Wire.beginTransmission(DS1631_ADDR); // Start new
     communication with the DS1631 temperature sensor
  Wire.write(0x51);                    // Send start conversion
     command
  Wire.endTransmission();              // Close communication
     session

  // If the clear button has been pressed reset the eeprom
     counter values
  if (digitalRead(CLEAR_PIN))
  {
    // Reset the totals for elapsed time and limit switches in
       EEPROM
    eeprom_write_float(elapsed_f_addr, (float)0);
    eeprom_write_float(elapsed_r_addr, (float)0);
    eeprom_write_dword(limit_f_addr, 0);
    eeprom_write_dword(limit_r_addr, 0);

    // Reset working variables for elapsed time and limit
       switches
    elapsed_f = (float)0;
    elapsed_r = (float)0;
    limit_f = 0;
    limit_r = 0;
  }

  // Perform measurement initialisation
  initialise();
}

/*
 * Code to be executed endlessly after setup is complete
 */
void loop() {
  // Measure voltage for each actuator
  voltage_f = readVoltage(POW_STATE_PIN_F);
  voltage_r = readVoltage(POW_STATE_PIN_R);
```

```
// Measure current for each actuator inverting direction if
   required
current_f = INVERT_DIR_F * readCurrent(CURR_SENS_PIN_F);
current_r = INVERT_DIR_R * readCurrent(CURR_SENS_PIN_R);

// If the front actuator is running process its data
if (voltage_f > VOLTAGE_THRESHOLD)
{
  // If the actuator was stopped but is now running reset it's
     variables
  if (running_f == false)
  {
    // Reset averaging variables and run time
    avg_curr_f = 0;
    avg_volt_f = 0;
    count_f = 0;
    run_time_f = 0;
    // Get a new start and last millisecond value
    millis_start_f = millis();
    millis_last_f = millis_start_f;
  }

  // The actuator is running so update the flag
  running_f = true;

  // Calculate run time
  millis_curr = millis();
     // Current system time in milliseconds
  run_time_f = (float)(millis_curr - millis_start_f) / 1000;
     // Calculate how many seconds the actuator has been
     running so far
  interval_f = (float)(millis_curr - millis_last_f) / 1000;
     // Calculate how many seconds since the last measurement
  millis_last_f = millis_curr;
     // The current milliseconds time is now the last
     measurement time
  elapsed_f += interval_f;
     // Update the elapsed time since the last measurement
  count_f ++;
     // Update the current number of measurements performed
     for this actuator
  avg_volt_f += voltage_f;
     // Add the current voltage to the average variable
  avg_curr_f += current_f;
     // Add the current current to the average variable

  // Detect if the limit switch has been activated (voltage
     present but no current)
  // The actuator must not already be limited and the current
     must be below the threshold
  // A limit condition must be maintained for a number of
     samples to prevent false positives
  if (limited_f == false && abs(current_f) < CURRENT_THRESHOLD
     )
  {
    limit_count_f ++; // Increment the limit switch time
       counter

    // If the limit counter threshold has been reached a limit
       switch has been activated
    if (limit_count_f >= LIMIT_THRESHOLD)
    {
```

```
        limit_f ++;                                    // Increment
           the number of limit switch activations
        limited_f = true;                              // Set the
           limit switch flag for this actuator
        eeprom_write_dword(limit_f_addr, limit_f); // Update the
            limit switch counter in EEPROM
      }
    }
    // Reset limit flag if actuator moves
    if (abs(current_f) > CURRENT_THRESHOLD)
    {
      limited_f = false; // Clear the limit switch flag for this
          actuator
      limit_count_f = 0; // Clear the limit switch time counter
    }
  }
  else
  {
    // If the actuator was running but is now stopped update it'
       s elapsed time
    if (running_f == true)
    {
      // Calculate run time
      millis_curr = millis();
                                                  // Current system
         time in milliseconds
      run_time_f = (float)(millis_curr - millis_start_f) / 1000;
          // Calculate how many seconds the actuator has been
         running so far
      interval_f = (float)(millis_curr - millis_last_f) / 1000;
          // Calculate how many seconds since the last
         measurement
      elapsed_f += interval_f;
                                                 // Update the elapsed
          time since the last measurement
      eeprom_write_float(elapsed_f_addr, elapsed_f);
                      // Update the elapsed time stored in EEPROM
    }
    running_f = false; // The actuator has stopped and all stop
       tasks have been performed
  }

  // If the rear actuator is running process its data
  if (voltage_r > VOLTAGE_THRESHOLD)
  {
    // If the actuator was stopped but is now running reset it's
        variables
    if (running_r == false)
    {
      // Reset averaging variables and run time
      avg_curr_r = 0;
      avg_volt_r = 0;
      count_r = 0;
      run_time_r = 0;
      // Get a new start and last millisecond value
      millis_start_r = millis();
      millis_last_r = millis_start_r;
    }

    // Get a new start and last millisecond value
    running_r = true;
```

```
    // Calculate run time
    millis_curr = millis();
       // Current system time in milliseconds
    run_time_r = (float)(millis_curr - millis_start_r) / 1000;
       // Calculate how many seconds the actuator has been
       running so far
    interval_r = (float)(millis_curr - millis_last_r) / 1000;
       // Calculate how many seconds since the last measurement
    millis_last_r = millis_curr;
       // The current milliseconds time is now the last
       measurement time
    elapsed_r += interval_r;
       // Update the elapsed time since the last measurement
    count_r ++;
       // Update the current number of measurements performed
       for this actuator
    avg_volt_r += voltage_r;
       // Add the current voltage to the average variable
    avg_curr_r += current_r;
       // Add the current current to the average variable

    // Detect if the limit switch has been activated (voltage
       present but no current)
    // The actuator must not already be limited and the current
       must be below the threshold
    // A limit condition must be maintained for a number of
       samples to prevent false positives
    if (limited_r == false && abs(current_r) < CURRENT_THRESHOLD
       )
    {
      limit_count_r ++; // Increment the limit switch time
         counter

      // If the limit counter threshold has been reached a limit
         switch has been activated
      if (limit_count_r >= LIMIT_THRESHOLD)
      {
        limit_r ++;                                 // Increment
           the number of limit switch activations
        limited_r = true;                           // Set the
           limit switch flag for this actuator
        eeprom_write_dword(limit_r_addr, limit_r); // Update the
           limit switch counter in EEPROM
      }
    }
    // Reset limit flag if actuator moves
    if (abs(current_r) > CURRENT_THRESHOLD)
    {
      limited_r = false; // Clear the limit switch flag for this
         actuator
      limit_count_r = 0; // Clear the limit switch time counter
    }
  }
  else
  {
    // If the actuator was running but is now stopped update it'
       s elapsed time
    if (running_r == true)
    {
      // Calculate run time
      millis_curr = millis();
                                                    // Current system
```

```
          time in milliseconds
      run_time_r = (float)(millis_curr - millis_start_r) / 1000;
          // Calculate how many seconds the actuator has been
          running so far
      interval_r = (float)(millis_curr - millis_last_r) / 1000;
          // Calculate how many seconds since the last
          measurement
      elapsed_r += interval_r;
                                                   // Update the elapsed
          time since the last measurement
      eeprom_write_float(elapsed_r_addr, elapsed_r);
                     // Update the elapsed time stored in EEPROM
    }
    running_r = false; // The actuator has stopped and all stop
        tasks have been performed
  }

  // Get the ambient temperature
  temp = readTemp();

  // If neither actuator is running then the current operation
      is finished
  // Work out data summaries for this operation
  if (running_f == false && running_r == false)
  {
    writeFile(”log_full.txt”); // Write the data for the last
        measurement to the uSD card
    writeFile(”log_summ.txt”); // Write the summary data for
        this operation to the uSD card

    // Wait for the power to come back on or for power loss to
        occur
    while (readVoltage(POW_STATE_PIN_F) < VOLTAGE_THRESHOLD &&
        readVoltage(POW_STATE_PIN_R) < VOLTAGE_THRESHOLD);

    // If the power is restored before the processor is shut
        down, then reinitialise and continue
    initialise();
  }
  else
  {
    writeFile(”log_full.txt”); // Write the data for the last
        measurement to the uSD card
  }

  // Wait for the timer to interrupt and break the loop
  waiting = true;
  while (waiting == true);
}

/*
 * Measurement initialisation routine
 * This is run at system power up and also if one operation ends
 * and a new operation starts before the system loses power.
 */
void initialise() {
  // Reset operation specific variables for run time,
      measurement intervals, and limit switch counter
  run_time_f = (float)0;
  interval_f = (float)0;
  run_time_r = (float)0;
  interval_r = (float)0;
  limit_count_f = 0;
```

```
  limit_count_r = 0;

  // Get start time from RTC
  readTime();

  // Read total values for elapsed time and limit switch
     activation from EEPROM
  elapsed_f = eeprom_read_float(elapsed_f_addr);
  elapsed_r = eeprom_read_float(elapsed_r_addr);
  limit_f = eeprom_read_dword(limit_f_addr);
  limit_r = eeprom_read_dword(limit_r_addr);
}

/*
 *  For writing formatted data to the sd card
 *  This function is used for both of the log files so the file
    to write is passed as a string pointer
 */
int writeFile(char* file) {
  File logFile = SD.open(file, FILE_WRITE); // Open the target
     file

  // If the file was opened successfully write the data
  if (logFile)
  {
    // Write the operation summary data fields
    if (file == "log_summ.txt")
    {
      logFile.print(avg_curr_f/count_f); // Write the average
         front actuator current
      logFile.print(",␣");
      logFile.print(avg_volt_f/count_f); // Write the average
         front actuator voltage
      logFile.print(",␣");
      logFile.print(run_time_f);         // Write the current
         operation front actuator run time
      logFile.print(",␣");
      logFile.print(elapsed_f);          // Write the total
         elapsed front actuator run time
      logFile.print(",␣");
      logFile.print(limit_f);            // Write the total
         front actuator limit switch activations
      logFile.print(",␣");
      logFile.print(avg_curr_r/count_r); // Write the average
         rear actuator current
      logFile.print(",␣");
      logFile.print(avg_volt_r/count_r); // Write the average
         rear actuator voltage
      logFile.print(",␣");
      logFile.print(run_time_r);         // Write the current
         operation rear actuator run time
      logFile.print(",␣");
      logFile.print(elapsed_r);          // Write the total
         elapsed rear actuator run time
      logFile.print(",␣");
      logFile.print(limit_r);            // Write the total rear
          actuator limit switch activations
      logFile.print(",␣");
    }
    // Write the full log data fields
    else
    {
```

```
    logFile.print(current_f);  // Write the current front
       actuator current
    logFile.print(",␣");
    logFile.print(voltage_f);  // Write the current front
       actuator voltage
    logFile.print(",␣");
    logFile.print(run_time_f); // Write the current operation
       front actuator run time
    logFile.print(",␣");
    logFile.print(elapsed_f);  // Write the total elapsed
       front actuator run time
    logFile.print(",␣");
    logFile.print(limit_f);    // Write the total front
       actuator limit switch activations
    logFile.print(",␣");
    logFile.print(current_r);  // Write the current rear
       actuator current
    logFile.print(",␣");
    logFile.print(voltage_r);  // Write the current rear
       actuator voltage
    logFile.print(",␣");
    logFile.print(run_time_r); // Write the current operation
       rear actuator run time
    logFile.print(",␣");
    logFile.print(elapsed_r);  // Write the total elapsed rear
        actuator run time
    logFile.print(",␣");
    logFile.print(limit_r);    // Write the total rear
       actuator limit switch activations
    logFile.print(",␣");
  }
  logFile.print(temp);                  // Write the current
     ambient temperature
  logFile.print(",␣");
  logFile.print(datetime[HOUR], DEC); // Write the current
     hour
  logFile.print(":");
  if (datetime[MINUTE]<10)
  {
    logFile.print("0");                 // If the current
       minute is less than 10 write a leading zero
  }
  logFile.print(datetime[MINUTE], DEC); // Write the current
     minute
  logFile.print(":");
  if (datetime[SECOND]<10)
  {
    logFile.print("0");                 // If the current
       second is less than 10 write a leading zero
  }
  logFile.print(datetime[SECOND], DEC); // Write the current
     second
  logFile.print("␣");
  logFile.print(datetime[DAY], DEC);    // Write the current
     day
  logFile.print("/");
  logFile.print(datetime[MONTH], DEC);  // Write the current
     month
  logFile.print("/");
  logFile.print(datetime[YEAR], DEC);   // Write the current
     year
```

```
    logFile.println();
    logFile.close();                              // Close the file
    return 0;
  }
  else return 1; // If an error has occurred
}

/*
 * Read voltage from one of the voltage inputs
 * The pin to read from is passed as a variable
 * The translation between measured voltage and actual voltage
     is modeled as a 2nd order quadratic
 * After calibration the equation is:
 * voltage = 0.1155*tmp^2 - 0.8175*tmp + 7.6267
 */
float readVoltage(int pin) {
  tmp = ((float)analogRead(pin) * 5) / 1024;
                                    // Read the voltage
  return tmp * ((tmp * tmp * 0.1155) + (tmp * -0.8175) + 7.6267)
      ; // Calculate actual voltage
}

/*
 * Read current from one of the ACS712 current sensors
 * The pin to read from is passed as a variable
 * The ACS712 output goes above or below 1/2 Vcc depending on
     current direction
 * It has a sensitivity of 66mV/A
 */
float readCurrent(int pin) {
  return (float)((analogRead(pin) - 514) * 4.97) / (0.066 *
      1024); // Read voltage and calculate current
}

/*
 * Request current time from the DS3231 RTC
 * This function is a modified version of the sample code
     provided
 * by John Boxall at tronixstuff for the DS3231 RTC
 * http://tronixstuff.com/2014/12/01/tutorial-using-ds1307-and-
     ds3231-real-time-clock-modules-with-arduino/
 */
void readTime() {
  Wire.beginTransmission(DS3231_ADDR);              // Begin
      communication with the DS3231
  Wire.write(0);                                    // Set DS3231
       register pointer to 00h
  Wire.endTransmission();                           // Close
      communication session
  Wire.requestFrom(DS3231_ADDR, 7);                 // Request
      seven bytes of data from the DS3231 starting from register
      00h
  datetime[SECOND] = bcdToDec(Wire.read() & 0x7f); // Get the
      current second
  datetime[MINUTE] = bcdToDec(Wire.read());         // Get the
      current minute
  datetime[HOUR] = bcdToDec(Wire.read() & 0x3f);    // Get the
      current hour
  Wire.read();                                      // Skip the
      day of week data field
  datetime[DAY] = bcdToDec(Wire.read());            // Get the
      current day
```

```
  datetime[MONTH] = bcdToDec(Wire.read());          // Get the
     current month
  datetime[YEAR] = bcdToDec(Wire.read());           // Get the
     current year
}

/*
 * Read temperature from DS1631 over I2C
 * This function is a modified version of code written by
 * Paul Priebbenow at Wenross Holdings Pty. Ltd.
 */
float readTemp() {
  // Working variables
  float temp;
  int msByte, lsByte;

  Wire.beginTransmission(DS1631_ADDR); // Begin communication
     with the DS1631
  Wire.write(0xAA);                    // Send read temperature
     command
  Wire.endTransmission();              // Close communication
     session

  Wire.requestFrom(DS1631_ADDR, 2); // Request two bytes of data
      from the DS1631
  while(Wire.available())
  {
    msByte = Wire.read(); // Get the high temperature byte
    lsByte = Wire.read(); // Get the low temperature byte
  }
  Wire.endTransmission(); // Close communication session

  lsByte = lsByte >> 4;                             // Move the
     data in the low byte to the correct position
  temp = (float)msByte + (float)lsByte * 0.0625; // Convert to
     floating point and combine the high and low byte
  return temp;
}

/*
 * Convert normal decimal numbers to binary coded decimal
 * This function is a modified version of the sample code
     provided
 * by John Boxall at tronixstuff for the DS3231 RTC
 * http://tronixstuff.com/2014/12/01/tutorial-using-ds1307-and-
     ds3231-real-time-clock-modules-with-arduino/
 */
byte decToBcd(byte val)
{
  return((val/10*16) + (val % 10));
}

/*
 * Convert binary coded decimal to normal decimal numbers
 * This function is a modified version of the sample code
     provided
 * by John Boxall at tronixstuff for the DS3231 RTC
 * http://tronixstuff.com/2014/12/01/tutorial-using-ds1307-and-
     ds3231-real-time-clock-modules-with-arduino/
 */
byte bcdToDec(byte val)
{
  return((val/16*10) + (val % 16));
```

```
}

/*
 * Timer 1 Compare Match A ISR
 * When timer 1 reaches 100mS the compare match will trigger an
    interrupt
 * This tells the microcontroller to stop waiting and perform
    the next data measurement
 */
ISR(TIMER1_COMPA_vect)
{
  // Break out of waiting loop
  waiting = false;
}
```

Appendix D

Data Analysis Software

D.1 The `Analysis.m` MATLAB Code

The MATLAB code used to process the data collected from the Mansell Power Lifters in the field.

Listing D.1: The MATLAB code used for processing the collected data.

```
clear all
close all

% Read in full data file
data = csvread('Data_Full.csv');

%% Fix the invalid temperature measurements
% If the temperature was read before the first conversion had taken place
% a value of 196 would be returned which is invalid. This will replace all
% 196 values with the next valid temperature reading. Since the next valid
% reading should be within a second it should be fairly accurate
pos = zeros(1,0);
for i=1:length(data(:,1))
    % If the current temperature value is invalid add it to the list
    if data(i,11) == 196
        pos = [pos; i];
    end
    % If the last row contained an invalid temperature and this row
    % contains a valid temperature go back and fix the invalid rows
    if ~isempty(pos) && pos(end) == i-1 && data(i,11) ~= 196
        % For each invalid temperature replace it with the correct one
        for j=1:length(pos)
            data(pos(j),11) = data(i,11);
```

```
        end
        % Clear the pos vector ready for the next run
        pos = zeros(1,0);
    end
end

%% Regenerate the summary data file
% There were a number of issues experienced with the summary
    data file as
% outlined in the dissertation. However since the summary file
    was based on
% data calculated from the full log file the missing data can be
     rebuilt by
% re-processing the full log file data.

% Array to hold the new summary data
summary = zeros(1, length(data(1,:)));

% State keeping variables
first = true;
limited_f = false;
limited_r = false;

% Loop through every line of data and regenerate the summary
    file
for i=2:length(data(:,1))
    % Analyse the current and voltage data to identify limit
        swich
    % activations. There must be at least 3 previous data points
         with zero
    % current to determine limit switch activations. Less than
        |0.5|A is
    % considered to be no active current.
    if i >= 3
        % Front actuator
        % If there has been no current flow for 3 samples and
            there is
        % still voltage a limit switch has been activated.
        if limited_f == true && abs(data(i, 1)) > 0.5
            limited_f = false;
        end
        if limited_f == false && data(i,2) > 9 && abs(data(i,1))
            < 0.5 && abs(data(i-1,1)) < 0.5 && abs(data(i-2,1))
           < 0.5
            data(i,5) = data(i-1,5) + 1;
            limited_f = true;
        else
            % Otherwise the counter should be the same
            data(i,5) = data(i-1,5);
        end

        % Rear actuator
        if limited_r == true && abs(data(i, 6)) > 0.5
            limited_r = false;
        end
        if limited_r == false && data(i,7) > 9 && abs(data(i,6))
            < 0.5 && abs(data(i-1,6)) < 0.5 && abs(data(i-2,6))
           < 0.5
            data(i,10) = data(i-1,10) + 1;
            limited_r = true;
        else
            data(i,10) = data(i-1,10);
        end
    end
```

```
    % If the current line does not have the same timestamp as
        the
    % previous line it is the start of a new operation
    if data(i, 12) ~= data(i-1, 12) || data(i, 13) ~= data(i-1,
        13)
        % Add the new operation to the data summary array
        if first == true
            summary = data(i-1, :);
            first = false;
        else
            summary = [summary; data(i-1, :)];
        end
    end
end

% Add the last line since it will always be the end of an
    operation
summary = [summary; data(length(data(:,1)), :)];

%% Work out total usage time per day
% Total usage time per day will be plotted alongside the number
    of
% operations per day to give an overview of the frequency and
    duration of
% the use of the Mansell Power Lifter.

% Read in all possible days from file
days = csvread('Days.csv');
% Run time for current day per actuator
day_time = zeros(1,2);
% For the run time per day summary
time_per_day = zeros(0, 3);

% Loop through each possible day
for day=1:length(days)
    % Look at each line in the summary data
    for i=1:length(summary(:,1))
        % If the current operation is in the current day add the
            time
        if summary(i,13) == days(day)
            day_time(1,1) = day_time(1,1) + summary(i,3);
            day_time(1,2) = day_time(1,2) + summary(i,8);
        end
    end
    % Copy the current day's data to the time per day matrix and
        reset
    time_per_day = [time_per_day; days(day), day_time(1,1),
        day_time(1,2)];
    day_time = zeros(1,2);
end

%% Generate histogram data for duration of operation
% This section generates histogram data for a number of non-
    evenly spaced
% bins placed at operation durations of interest to show what
    length of
% operations the Mansell Power Lifter is being subjected to.

% The bins
bins = [1; 5; 10; 15; 20; 25; 30; 35; 40; 100];
% Counts will be performed individually for each actuator
counts = zeros(length(bins),2);

% For each operation
```

```
for i=1:length(summary(:,1))
    % For each actuator
    for j=1:2
        % For the first bin
        if summary(i,(j*3)+((j-1)*2)) < bins(1)
            % Increment the appropriate bin count
            counts(1,j) = counts(1,j) + 1;
        else
            % For each bin
            for bin=2:length(bins)
                % If in the bin range
                if summary(i,(j*3)+((j-1)*2)) < bins(bin) &&
                   summary(i,(j*3)+((j-1)*2)) > bins(bin-1)
                    counts(bin,j) = counts(bin,j) + 1;
                end
            end
        end
    end
end

% Make the final output matrix
op_duration_hist = [bins, counts];

%% Work out the split between 12V and 24V modes and average
   voltage and current for each
% Since there will be some data points logged in which neither
   actuator is
% running the total number of "running" samples will be needed
run_count = 0;
mode_count = zeros(1,2);        % Number of records in each mode
curr_volt_count = zeros(1,2);   % Number or records used for
   current and voltage averages
current_sum = zeros(1,2);       % Sum of current in each mode
voltage_sum = zeros(1,2);       % Sum of voltage in each mode

for i=1:length(data(:,1))
    % If the voltage of either actuator is between 9 and 17
       volts the system
    % can be considered to be in 12V mode
    if (data(i,2) > 9 && data(i,2) <= 17) || (data(i,7) > 9 &&
       data(i,7) <= 17)
        % If the front actuator is running
        if data(i,2) > 9
            % Add the current and voltage information to the
               averaging variables
            curr_volt_count(1) = curr_volt_count(1) + 1;
            current_sum(1) = current_sum(1) + abs(data(i,1));
            voltage_sum(1) = voltage_sum(1) + data(i,2);
        end
        % If the rear actuator is running
        if data(i,7) > 9
            curr_volt_count(1) = curr_volt_count(1) + 1;
            current_sum(1) = current_sum(1) + abs(data(i,6));
            voltage_sum(1) = voltage_sum(1) + data(i,7);
        end
        % Update the mode counters
        mode_count(1) = mode_count(1) + 1;
        run_count = run_count + 1;
    % If the voltage of either actuator is above 17 volts the
       system can be
    % considered to be in 24V mode
    elseif data(i,2) > 17 || data(i,7) > 17
        % If the front actuator is running
```

```
        if data(i,2) > 9
            % Add the current and voltage information to the
                averaging variables
            curr_volt_count(2) = curr_volt_count(2) + 1;
            current_sum(2) = current_sum(2) + abs(data(i,1));
            voltage_sum(2) = voltage_sum(2) + data(i,2);
        end
        % If the rear actuator is running
        if data(i,7) > 9
            curr_volt_count(2) = curr_volt_count(2) + 1;
            current_sum(2) = current_sum(2) + abs(data(i,6));
            voltage_sum(2) = voltage_sum(2) + data(i,7);
        end
        % Update the mode counters
        mode_count(2) = mode_count(2) + 1;
        run_count = run_count + 1;
    end
end

% Calculate the mode percentages
mode_percentages = zeros(1,2);
mode_percentages = mode_count/run_count;
fprintf(1, '%s%f%s\n', 'The␣percentage␣of␣run␣time␣spent␣in␣12V␣
   mode␣was:␣', mode_percentages(1)*100, '␣%');
fprintf(1, '%s%f%s\n', 'The␣percentage␣of␣run␣time␣spent␣in␣24V␣
   mode␣was:␣', mode_percentages(2)*100, '␣%');

% Calculate the mode current averages
current_averages = zeros(1,2);
current_averages = current_sum./curr_volt_count;
fprintf(1, '%s%f%s\n', 'The␣average␣actuator␣current␣in␣12V␣mode
   ␣was:␣', current_averages(1), '␣A');
fprintf(1, '%s%f%s\n', 'The␣average␣actuator␣current␣in␣24V␣mode
   ␣was:␣', current_averages(2), '␣A');

% Calculate the mode voltage averages
voltage_averages = zeros(1,2);
voltage_averages = voltage_sum./curr_volt_count;
fprintf(1, '%s%f%s\n', 'The␣average␣actuator␣voltage␣in␣12V␣mode
   ␣was:␣', voltage_averages(1), '␣V');
fprintf(1, '%s%f%s\n', 'The␣average␣actuator␣voltage␣in␣24V␣mode
   ␣was:␣', voltage_averages(2), '␣V');

%% Output the corrected CSV files
% Increased precision is required to prevent integers that have
   been stored
% as floating point numbers from being rounded
dlmwrite('Data_Summary.csv', summary, 'delimiter', ',', '
   precision', 10);
dlmwrite('Data_Full_Corrected.csv', data, 'delimiter', ',', '
   precision', 10);
dlmwrite('Time_Per_Day.csv', time_per_day, 'delimiter', ',', '
   precision', 10);
dlmwrite('Op_Duration_Hist.csv', op_duration_hist, 'delimiter',
    ',', 'precision', 10);
```

Appendix E

Additional Data

E.1 Additional Figures

The following figures contain collected and processed data which was referenced in the main body of this report but was deemed to large to include in the main text.

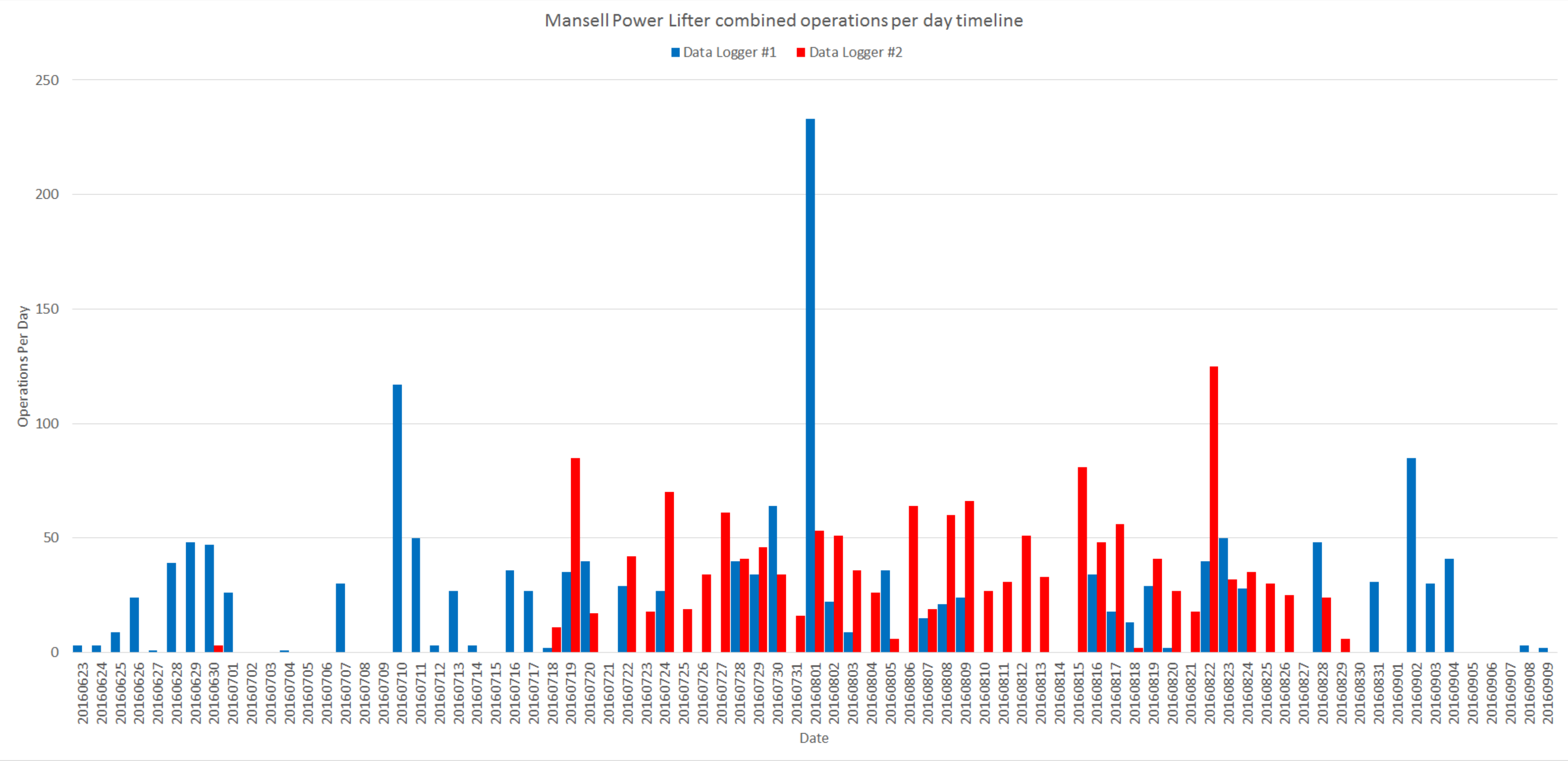


Figure E.1: Combined timeline of operations per day for both data loggers. (Source: author)

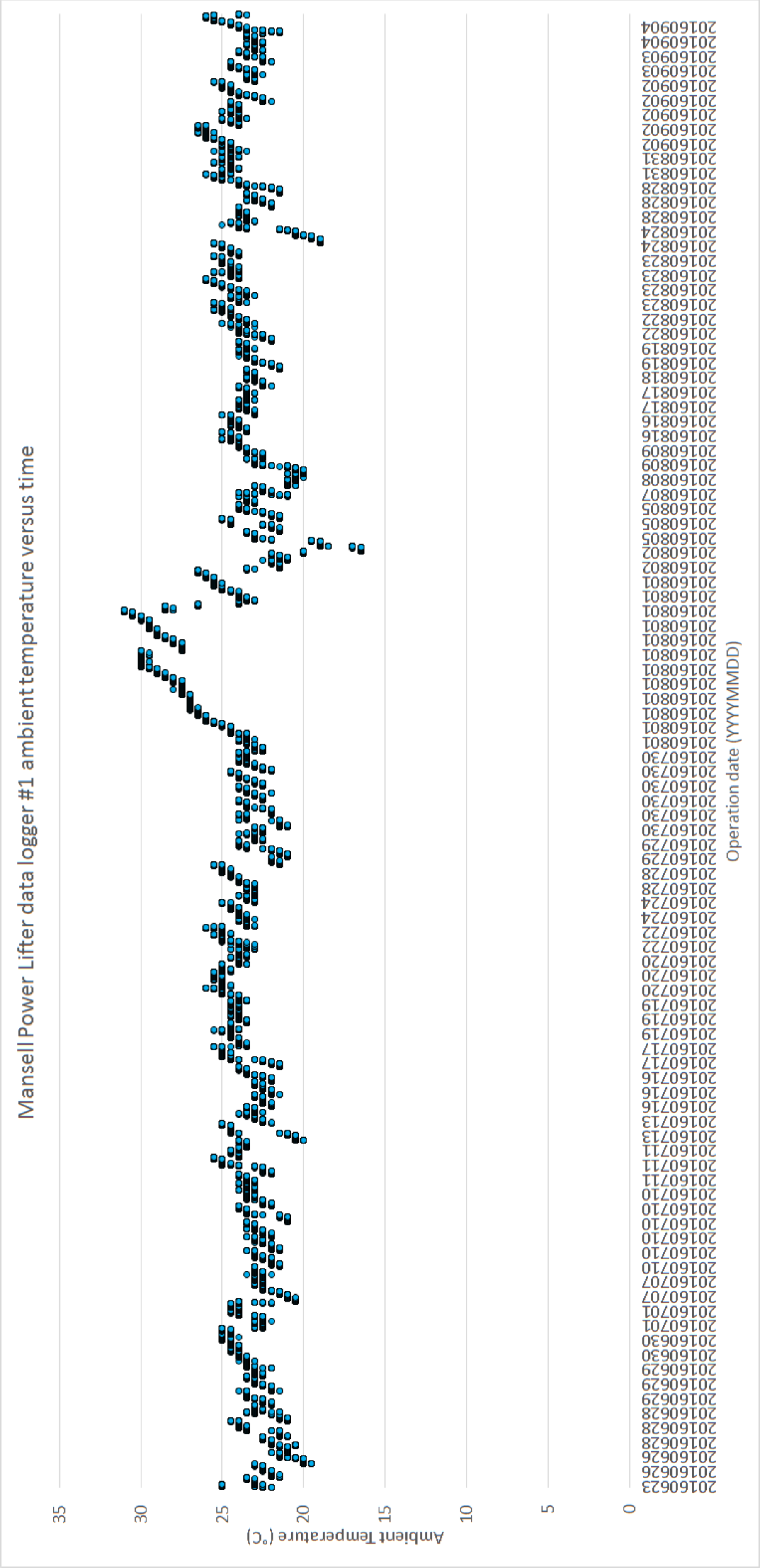


Figure E.2: Mansell Power Lifter data logger #1 ambient temperature versus time. (Source: author)

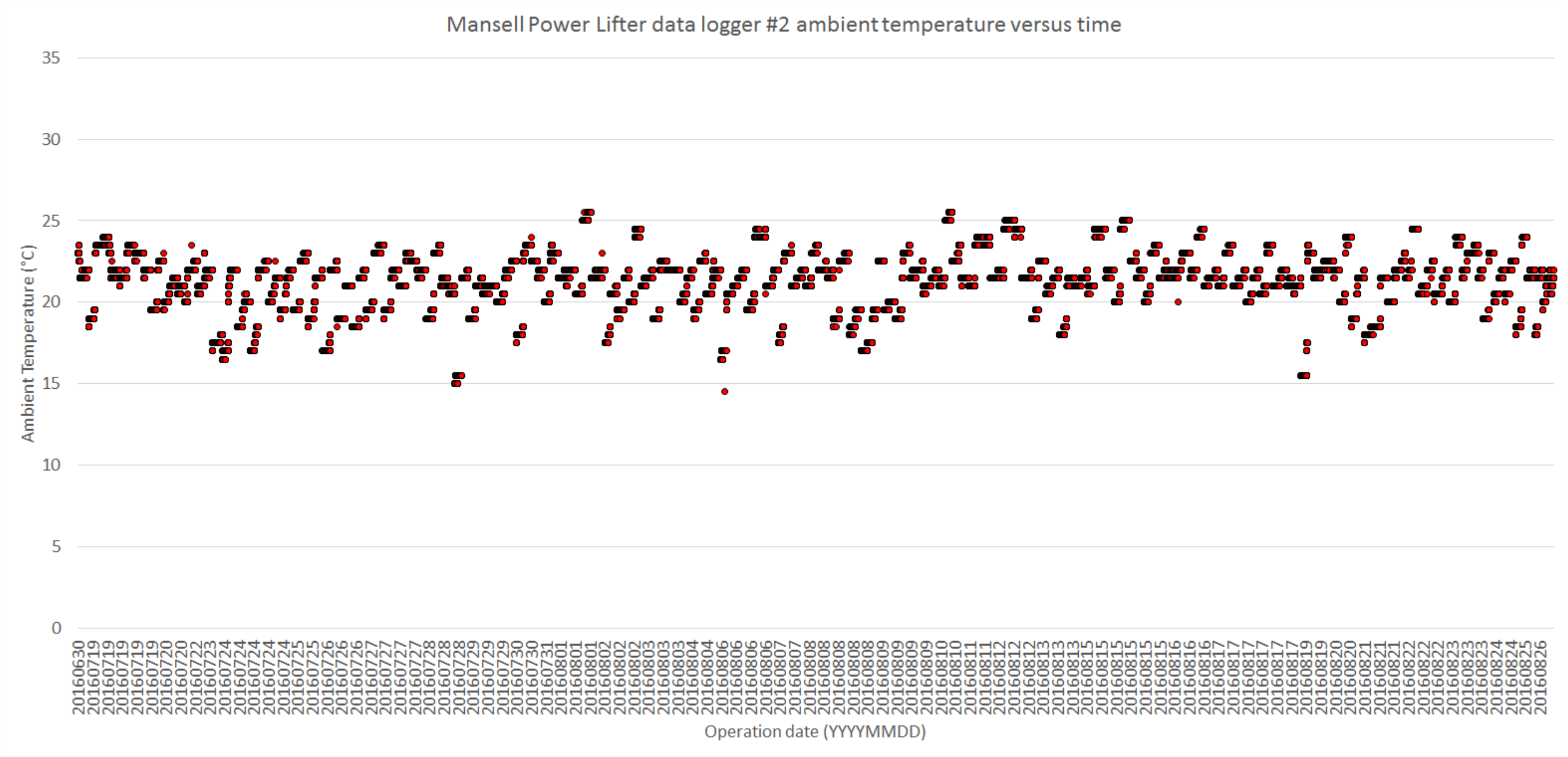


Figure E.3: Mansell Power Lifter data logger #2 ambient temperature versus time. (Source: author)

Appendix F

Risk Assessment

University of Southern Queensland

Generic Risk Management Plan

Workplace (Division/Faculty/Section):
USQ - School of Electrical and Mechanical Engineering and
Wenross Holdings Pty. Ltd.

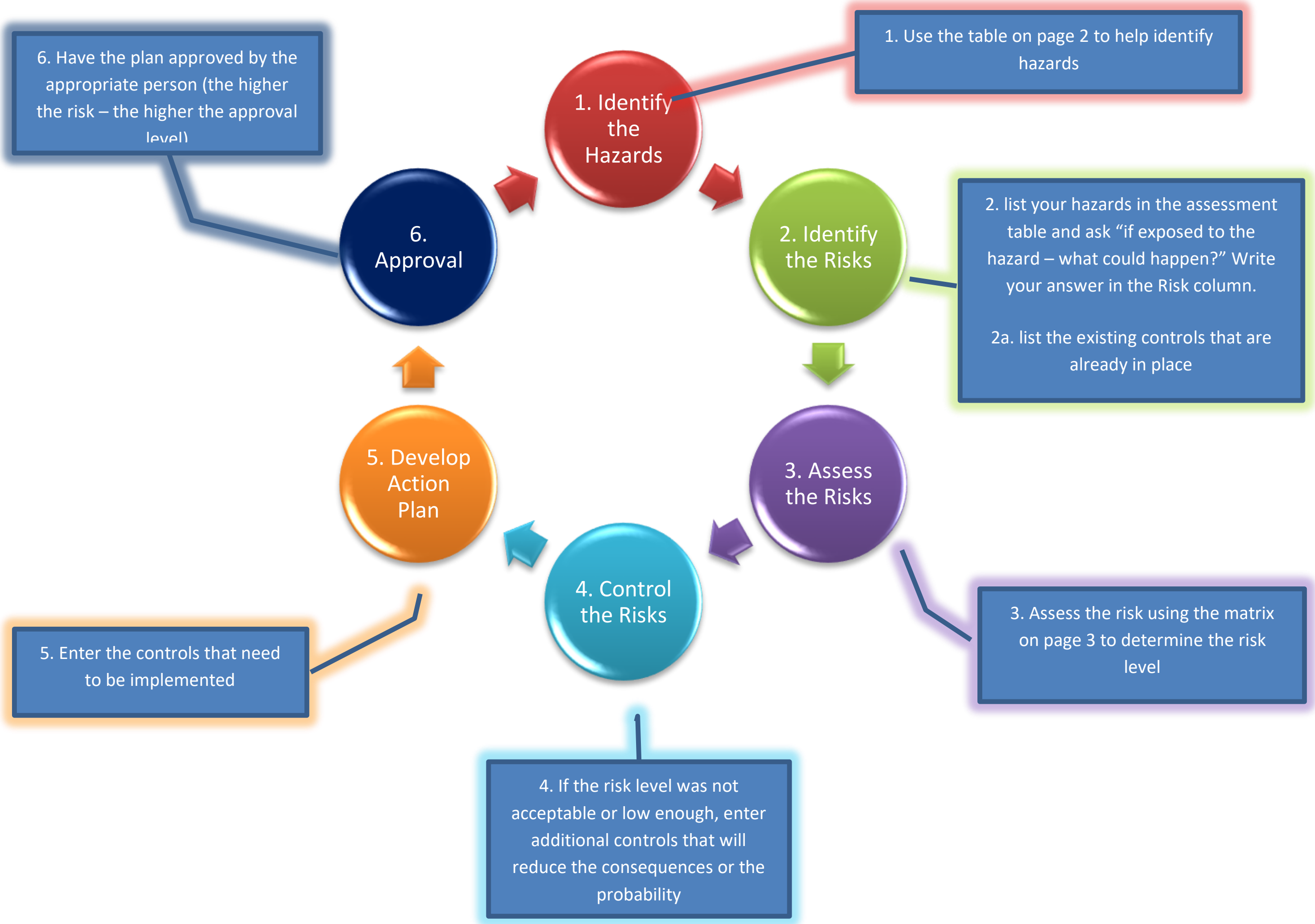
Assessment No (if applicable): N/A	Assessment Date: 24/05/2016	Review Date: (5 years maximum) 13/10/2016

Context: What is being assessed? Describe the item, job, process, work arrangement, event etc:
The risks associated with conducting the Mansell Power Lifter actuator failure analysis.

Assessment Team – who is conducting the assessment?

Assessor(s):
Dr John Leis
Others consulted: (eg elected health and safety representative, other personnel exposed to risks)
N/A

The Risk Management Process



1. Use the table on page 2 to help identify hazards

2. list your hazards in the assessment table and ask “if exposed to the hazard – what could happen?” Write your answer in the Risk column.

2a. list the existing controls that are already in place

3. Assess the risk using the matrix on page 3 to determine the risk level

4. If the risk level was not acceptable or low enough, enter additional controls that will reduce the consequences or the probability

5. Enter the controls that need to be implemented

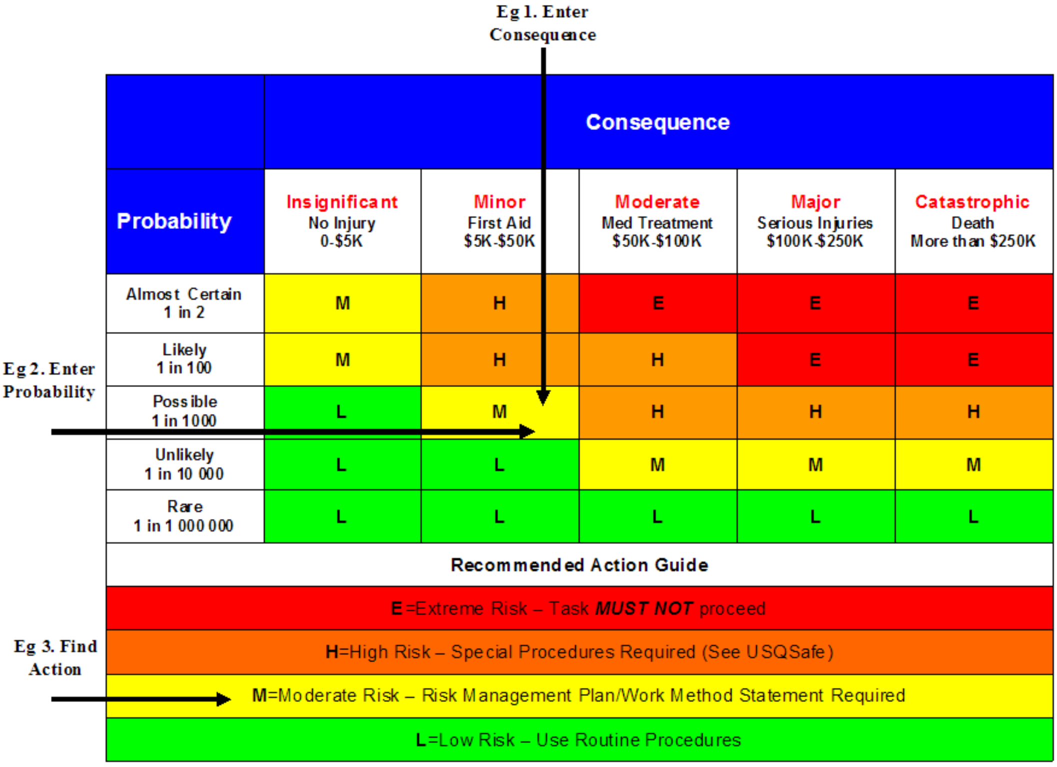
6. Have the plan approved by the appropriate person (the higher the risk – the higher the approval level)

Step 1 - Identify the hazards (use this table to help identify hazards then list all hazards in the risk table)

General Work Environment		
☐ Sun exposure	☐ Water (creek, river, beach, dam)	☒ Sound / Noise
☐ Animals / Insects	☐ Storms / Weather/Wind/Lightning	☐ Temperature (heat, cold)
☐ Air Quality	☐ Lighting	☐ Uneven Walking Surface
☒ Trip Hazards	☐ Confined Spaces	☐ Restricted access/egress
☐ Pressure (Diving/Altitude)	☐ Smoke	☐
Other/Details:		
Machinery, Plant and Equipment		
☐ Machinery (fixed plant)	☒ Machinery (portable)	☒ Hand tools
☐ Laser (Class 2 or above)	☐ Elevated work platforms	☐ Traffic Control
☐ Non-powered equipment	☐ Pressure Vessel	☒ Electrical
☐ Vibration	☒ Moving Parts	☐ Acoustic/Noise
☐ Vehicles	☐ Trailers	☐ Hand tools
Other/Details:		
Manual Tasks / Ergonomics		
☐ Manual tasks (repetitive, heavy)	☐ Working at heights	☐ Restricted space
☐ Vibration	☐ Lifting Carrying	☐ Pushing/pulling
☐ Reaching/Overstretching	☐ Repetitive Movement	☐ Bending
☐ Eye strain	☒ Machinery (portable)	☒ Hand tools
Other/Details:		
Biological (e.g. hygiene, disease, infection)		
☐ Human tissue/fluids	☐ Virus / Disease	☐ Food handling
☐ Microbiological	☐ Animal tissue/fluids	☐ Allergenic
Other/Details:		
Chemicals Note: Refer to the label and Safety Data Sheet (SDS) for the classification and management of all chemicals.		
☐ Non-hazardous chemical(s)	☐ 'Hazardous' chemical (Refer to a completed hazardous chemical risk assessment)	
☐ Engineered nanoparticles	☐ Explosives	☐ Gas Cylinders
Name of chemical(s) / Details:		
Critical Incident – resulting in:		
☐ Lockdown	☐ Evacuation	☐ Disruption
☐ Public Image/Adverse Media Issue	☐ Violence	☐ Environmental Issue
Other/Details:		
Radiation		
☐ Ionising radiation	☐ Ultraviolet (UV) radiation	☐ Radio frequency/microwave
☐ infrared (IR) radiation	☐ Laser (class 2 or above)	☐
Other/Details:		
Energy Systems – incident / issues involving:		
☒ Electricity (incl. Mains and Solar)	☐ LPG Gas	☐ Gas / Pressurised containers
Other/Details:		
Facilities / Built Environment		
☐ Buildings and fixtures	☐ Driveway / Paths	☐ Workshops / Work rooms
☐ Playground equipment	☐ Furniture	☐ Swimming pool
Other/Details:		
People issues		
☐ Students	☐ Staff	☐ Visitors / Others
☐ Physical	☐ Psychological / Stress	☐ Contractors
☐ Fatigue	☐ Workload	☐ Organisational Change
☐ Workplace Violence/Bullying	☐ Inexperienced/new personnel	☐
Other/Details:		

Step 1 (cont) Other Hazards / Details (enter other hazards not identified on the table)
Data logging equipment designed and constructed may interfere with the normal operation of the Mansell Power Lifter.
Analysis of data and conclusions made may be inaccurate.
The designed solution for actuator failures may be unreliable or result in operational issues for the Mansell Power Lifter.

Risk Matrix



Eg 1. Enter Consequence

Eg 2. Enter Probability

Eg 3. Find Action

	Consequence				
Probability	Insignificant No Injury 0-$5K	Minor First Aid $5K-$50K	Moderate Med Treatment $50K-$100K	Major Serious Injuries $100K-$250K	Catastrophic Death More than $250K
Almost Certain 1 in 2	M	H	E	E	E
Likely 1 in 100	M	H	H	E	E
Possible 1 in 1000	L	M	H	H	H
Unlikely 1 in 10 000	L	L	M	M	M
Rare 1 in 1 000 000	L	L	L	L	L

Recommended Action Guide

E=Extreme Risk – Task ***MUST NOT*** proceed

H=High Risk – Special Procedures Required (See USQSafe)

M=Moderate Risk – Risk Management Plan/Work Method Statement Required

L=Low Risk – Use Routine Procedures

Risk register and Analysis

Step 1 (cont)	Step 2	Step 2a	Step 3			Step 4				
Hazards: From step 1 or more if identified	**The Risk:** What can happen if exposed to the hazard with existing controls in place?	**Existing Controls:** What are the existing controls that are already in place?	**Risk Assessment:** (use the Risk Matrix on p3) Consequence x Probability = Risk Level			**Additional controls:** Enter additional controls if required to reduce the risk level	**Risk assessment with additional controls:** (use the Risk Matrix on p3 – has the consequence or probability changed?)			**Controls Implemented? Yes/No**
			Consequence	**Probability**	**Risk Level**		**Consequence**	**Probability**	**Risk Level**	
Example										
Working in temperatures over 35^0 C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	Regular breaks, chilled water available, loose clothing, fatigue management policy.	catastrophic	possible	high	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
Trip hazards	Personal injury due to falling.	Ensure a tidy workplace, avoid placing cables or objects across walkways.	Minor	Possible	Moderate	Existing controls are appropriate.	Minor	Possible	Moderate	Yes
Sound/noise	Hearing loss or damage due to excess noise levels.	Wear hearing protection in noisy environments, Minimise exposure to areas containing loud noise.	Moderate	Unlikely	Moderate	Existing controls are appropriate.	Moderate	Unlikely	Moderate	Yes
Machinery (portable)	Personal injury or damage to equipment.	Ensure that safety mechanisms such as guards are in place and follow safe use guidelines	Moderate	Possible	High	Only operate equipment when others are nearby and can help if something goes wrong. Ensure that the equipment is in good working order prior to use.	Moderate	Unlikely	Moderate	Yes
Moving parts	Personal injury or damage to equipment due to becoming stuck in moving parts.	Ensure that limbs, loose clothing, and other items are well away from moving parts while operating.	Moderate	Possible	High	Only operate equipment when others are nearby and can help if something goes wrong. Ensure that lockout systems are in place while working on moving parts.	Moderate	Unlikely	Moderate	Yes
Hand tools	Personal injury or damage to equipment.	Use items only for their intended purpose and according to use instructions. Wear appropriate personal protective equipment.	Minor	Possible	Moderate	Only operate equipment when others are nearby and can help if something goes wrong. Ensure that the equipment is in good working order prior to use.	Minor	Unlikely	Low	Yes
Electrical	Risk of electric shock.	Ensure that circuts under test are fitted with RCD protection.	Moderate	Unlikely	Moderate	Ensure that devices are unplugged when working on them. Discharge capacitors. Ensure that others are nearby to help if something goes wrong.	Moderate	Rare	Low	Yes
Data logger design faults	The data logger designed may interfere with the normal operation of the Mansell Power Lifter.	Perform thorough testing of the data logger before deploying it to the field.	Major	Possible	High	Follow best practices during the design and development stage, meet all required standards, ensure that electrical engineers from Wenross Holdings Pty. Ltd. oversee the design and approve its use.	Moderate	Unlikely	Moderate	Yes

Step 1 (cont)	Step 2	Step 2a	Step 3			Step 4				
Hazards: From step 1 or more if identified	**The Risk:** What can happen if exposed to the hazard with existing controls in place?	**Existing Controls:** What are the existing controls that are already in place?	**Risk Assessment:** (use the Risk Matrix on p3) Consequence x Probability = Risk Level			**Additional controls:** Enter additional controls if required to reduce the risk level	**Risk assessment with additional controls:** (use the Risk Matrix on p3 – has the consequence or probability changed?)			**Controls Implemented? Yes/No**
			Consequence	**Probability**	**Risk Level**		**Consequence**	**Probability**	**Risk Level**	
Example										
Working in temperatures over 35^0 C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	Regular breaks, chilled water available, loose clothing, fatigue management policy.	catastrophic	possible	high	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
Data analysis inaccuracies	Inaccuracies in data analysis lead to incorrect conclusion about the cause of actuator failures.	Ensure that analysis performed and conclusions made are based on reliable data and sound reasoning.	Minor	Possible	Moderate	Keep supervisors informed of the progress and have them to sanity check the results.	Minor	Unlikely	Low	Yes
Inadequate or unreliable solution design	The proposed solution for resolving actuator failures does not work or interferes with the normal operation of the Mansell Power Lifter.	Perform thorough testing and verification of any design before deploying it to the field.	Major	Possible	High	Follow best practices during the design and development stage, meet all required standards, ensure that electrical engineers from Wenross Holdings Pty. Ltd. oversee the design and approve its use.	Moderate	Unlikely	Moderate	Yes
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No

Step 1 (cont)	Step 2	Step 2a	Step 3			Step 4				
Hazards: From step 1 or more if identified	**The Risk:** What can happen if exposed to the hazard with existing controls in place?	**Existing Controls:** What are the existing controls that are already in place?	**Risk Assessment:** (use the Risk Matrix on p3) Consequence x Probability = Risk Level			**Additional controls:** Enter additional controls if required to reduce the risk level	**Risk assessment with additional controls:** (use the Risk Matrix on p3 – has the consequence or probability changed?)			**Controls Implemented? Yes/No**
			Consequence	**Probability**	**Risk Level**		**Consequence**	**Probability**	**Risk Level**	
Example										
Working in temperatures over 35^0 C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	Regular breaks, chilled water available, loose clothing, fatigue management policy.	catastrophic	possible	high	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No
			Select a consequence	Select a probability	Select a Risk Level		Select a consequence	Select a probability	Select a Risk Level	Yes or No

Step 5 – Action Plan (for controls not already in place)

Control Option	Resources	Person(s) responsible	Proposed implementation date

Step 6 – Approval

Drafter's Comments:

Drafter Details:

Name: Signature: Date: / /

Assessment Approval: (Extreme or High = VC, Moderate = Cat 4 delegate or above, Low = Manager/Supervisor)

I am satisfied that the risks are as low as reasonably practicable and that the resources required will be provided.

Name: Signature: Date: / /

Position Title: