



Development of Electronic Motor Control Prototype for Vehicle Shock Absorber

ENGR9700 Masters Thesis – 18 Units

Lucas Paix – 2121689

May 2019

Supervisors: Dr Nasser Asgari (Academic)
and Andrew Webb (Industry)

Submitted to the College of Science and Engineering in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Robotics)(Honours) / Master of Engineering (Electronics) at Flinders University – Adelaide, Australia

Declaration of Originality

I certify that this work does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Lucas Paix

May 2019

Acknowledgements

My greatest appreciation extends to the supervisors of this project: Dr Nasser Asgari and Andrew Webb, who consistently helped keep the research on track towards a meaningful direction. Their ideas and suggestions towards research helped me the most to emerge from the times of uncertainty.

Additionally, the electronic and mechanical teams at Supashock provided an abundance of information and solutions towards even the most mysterious problems. Without them, much of the project design and programming would have been utter chaos. Special thanks to Sean Reynolds, Tarkis Cooper and Ben Grant.

The design and validation of the PCB was pushed forward strongly by to the assistance of the engineering services team at Flinders University: Craig Dawson, Fiona Cramer, Damian Kleiss, Richard Stanley and Craig Peacock. When a flawed design, malfunctioning component or unresponsive program raised its head, members of the team could always immediately home in to the source of the problem and provide guidance to appropriate solutions.

Finally, I would like to thank caffeine, which seemed to supply limitless motivation and energy to help me push through all brain fog barriers and focus on the important tasks up to the final minutes.

Thank you all.

Abstract

In all road vehicles, suspension systems act to reduce the effects of road disturbance to improve driving performance and driver comfort. These can be powerful but expensive active systems, cheaper but limited passive systems or the middle-ground semi-active. Different driving conditions require different degrees of control and cost, which must be all considered when deciding what type of suspension to use. Settling the trade-off between cost and performance is typically achieved using passive systems with mechanically adjustable damper strength, as is the case for many dampers made by Supashock. These provide cheap and effective solutions for many cases, but compromise on the flexibility to adjust the damping strength while driving. Addressing this problem can be performed by building an electronic system to control the mechanical damper adjuster on-the-drive through software.

An investigation was performed into the development of an electronic shock absorber controller and demonstration test rig for Supashock. The primary objective was to demonstrate the capability for an electromechanical system to provide accurate control of a variable damper via control of hydraulic oil pressure. This was approached by designing a custom needle-and-seat continuous hydraulic valve which formed the basis of the test rig. The valve was controlled by a motor mechanism and was operated using a custom PCB with appropriate driver circuitry.

The test rig was designed to use the machined aluminium custom valve, 3D printed mounting structures, a stepper motor for actuation and a hall effect sensor for feedback. Overall, a complete prototype was fabricated using a 3D printed test valve, which was demonstrated to provide successful operation of the system mechanical motion. It was measured that the motor could fully sweep the 3D printed valve from open to closed within 0.75 seconds at a maximum speed of 1600 pulses per second. The valve was tested in dry conditions to validate the designs and is awaiting delivery of the machined aluminium valve for further testing against hydraulic pressure.

A custom PCB was designed and fabricated to provide electronic control and measurement of the test rig. This provided capability to drive a stepper motor up to 2.5A and measure the position feedback of the custom valve via a hall effect sensor analog voltage. After overcoming PCB rework requirements, the circuit could successfully demonstrate control of the test rig. The microcontroller provided complete control the stepper motor at desired speed, step count, direction and microstep size, and read the position of the valve via the hall effect sensor voltage.

Additional features of the PCB were tested such as input voltage measurement and motor driver current measurement. This found that the microcontroller successfully calculated the input voltage at 11.6V but could not reliably calculate motor current. When observing the current amplifier output, the voltage contained large noisy spikes which completely obscured any useful information. Further testing has been recommended for this including a low-pass filter to remove the high frequency spikes and analyse for useful information.

The primary outcome of this project successfully provided the development framework for an electronically controlled shock absorber damper. The design focus towards testing flexibility means that these achievements can be easily adapted by Supashock towards any specific shock absorber in a future as desired.

Table of Contents

1. INTRODUCTION	1
1.1 BACKGROUND.....	1
1.2 PROJECT SCOPE AND OBJECTIVES	3
2. LITERATURE REVIEW	5
2.1 VARIABLE CONTROL MECHANISMS	5
2.1.1 Flow Control	5
2.1.2 Rheological	7
2.2 MOTORS.....	7
2.2.1 Stepper	8
2.2.2 Brushless DC.....	11
2.2.3 Brushed DC.....	12
2.2.4 Summary	14
2.3 DISTANCE MEASUREMENT.....	15
2.3.1 Encoders	15
2.3.2 Potentiometer	17
2.3.3 Linear Variable Differential Transformer (LVDT)	19
2.3.4 Hall Effect Sensor	19
2.3.5 Time of Flight Sensor	20
2.4 EXISTING RETROFIT PRODUCTS	21
2.5 SUMMARY.....	22
3. ASSEMBLY DESIGN	24
3.1 OBJECTIVES	24
3.2 MECHANICAL.....	24

3.2.1	Hydraulic Valve Block	24
3.2.2	Hydraulic Valve Needle	26
3.2.3	Hydraulic Valve Assembly	27
3.2.4	Motor Shaft	29
3.2.5	Mounting Shell	31
3.2.6	Sensor Mount	33
3.2.7	Total Assembly	34
3.3	HALL EFFECT SENSOR	37
3.4	STEPPER MOTOR	37
3.5	PCB	38
3.5.1	Power	39
3.5.2	Motor Driver	42
3.5.3	Sensor Filtering	46
3.5.4	Microcontroller	47
3.5.5	Communication	50
3.5.6	LEDs	52
3.5.7	Current Measurement	53
3.5.8	Protection	54
3.5.9	Placement and Routing	55
3.5.10	Cost	58
4.	RESULTS	60
4.1	PRELIMINARY TESTING	60
4.1.1	Mechanical	60
4.1.2	Electronic	63

4.2	ASSEMBLY PROTOTYPING	68
4.2.1	Mechanical	68
4.2.2	Electronic.....	74
4.3	FINAL ASSEMBLY.....	77
4.3.1	Mechanical	77
4.3.2	PCB.....	80
4.4	SOFTWARE	86
4.4.1	Initialisation.....	86
4.4.2	Motor Driving	88
4.4.3	Sensor	90
4.4.4	Testing and Debugging Features.....	94
5.	CONCLUSIONS	98
5.1	RESOURCE ESTIMATE.....	99
5.2	FUTURE WORK AND RECOMMENDATIONS	100
5.2.1	Testing.....	100
5.2.2	Design Recommendations.....	100
6.	BIBLIOGRAPHY	103
7.	APPENDICES	106
7.1	VALVE MECHANICAL DRAWINGS	0
7.2	VALVE SEAL	0
7.3	VALVE BILL OF MATERIALS	0
7.4	TEST RIG ASSEMBLY BILL OF MATERIALS.....	0
7.5	PCB SCHEMATICS.....	0
7.6	PCB BILL OF MATERIALS	0
7.7	ARDUINO TEST CODES	0

7.8 PIC DEVICE CODE0

Table of Figures

Figure 1.1: Diagram of Conventional Quarter-Car Passive Suspension System	2
Figure 1.2: Primary Objective Test Assembly Layout of Demonstration Rig and Custom PCB	4
Figure 2.1: Stepper Motor (Pololu, 2019)	8
Figure 2.2: Unipolar (left) and Bipolar (right) wiring configurations for permanent magnet rotor motor (top)(Condit & Jones, 2004).....	9
Figure 2.3: Driving Waveform of full-step drive for stepper motor (Hopkins, 2012)	9
Figure 2.4: Brushless DC Motor (SeedStudio, 2019).....	11
Figure 2.5: MOSFET PWM Driver Circuit for 3-phase BLDC motor (Digi-Key, 2016)	12
Figure 2.6: Three-phase BLDC Motor Hall Effect Coil Timings, Sensing and Hall Effect Sensing and Back EMF Timing Signals (Zhao & Yu, 2011)	12
Figure 2.7: Brushed DC Motor (RobotShop, 2019).....	13
Figure 2.8: Absolute Optical Encoder Slotted Wheel (Z. Khan, 2015)	15
Figure 2.9: Incremental Optical Encoder Slotted Wheel (Z. Khan, 2015).....	17
Figure 2.10: Single Turn (Left) and Multi-Turn (Right) Potentiometers (RSComponents, 2019a), (RSComponents, 2019b)	18
Figure 2.11: Basic Operating Diagram of LVDT (National_Instruments, 2018)	19
Figure 2.12: Example Linear Analog Hall Effect Sensors (Honeywell, 2015).....	20
Figure 2.13: VL53L0X Time of flight sensor with breakout board (LittleBird, 2018)	21
Figure 2.14: Tein EDFC Active Conversion Kit (Tein, 2013).....	22
Figure 3.1: Autodesk Inventor Model of Custom Valve Block	25
Figure 3.2: Autodesk Inventor Model of Valve Block Half-Section View	26
Figure 3.3: Autodesk Inventor Model of Valve Needle.....	27
Figure 3.4: Autodesk Inventor Models of Valve Assembly Fully Opened (Left) and Fully Closed (Right)	28
Figure 3.5: Complete Shaft Mechanism of Mechanical Assembly.....	29
Figure 3.6: Exploded Assembly View of the Motor Shaft Mechanism	30
Figure 3.7: Custom Hex Socket Model.....	30
Figure 3.8: Custom Mount Shell Overall View	32
Figure 3.9: Bottom Face of Mounting Shell Highlighting Different NEMA Size Capabilities.....	33
Figure 3.10: Sensor Mounting Bracket Orthographic View (Left) and Sensor Side View (Right)	34
Figure 3.11: Total Test Rig Assembly External View	35

Figure 3.12: Total Test Rig Assembly Half Section View	35
Figure 3.13: SS49E Linear Ratiometric Hall Effect Sensor (Honeywell, 2015)	37
Figure 3.14: 42BYGHM809 Stepper motor.....	37
Figure 3.15: Altium 3D Model of Custom Driver PCB	38
Figure 3.16: Top Layer Routing Model of Custom Driver PCB	39
Figure 3.17: Power Supply Connection with Reverse Polarity Protection MOSFET	39
Figure 3.18: Power Supply Circuit.....	40
Figure 3.19: Motor Driver Circuit.....	43
Figure 3.20: Thermal Vias Below Motor Driver Power Pad to Dissipate Heat to Ground Plane.....	44
Figure 3.21: Circuit Layout of Microstepping Selection System.....	45
Figure 3.22: Hall Effect Sensor Filtering	47
Figure 3.23: PIC32MX530F128H Microcontroller Wiring Diagram.....	48
Figure 3.24: Programming Header Wiring Diagram	49
Figure 3.25: USB Micro Socket Wiring Diagram.....	51
Figure 3.26: CAN Transceiver with End Termination and Protection Diodes	52
Figure 3.27: DB9 CAN Connector Wiring Diagram with Reverse Polarity MOSFET and 0Ω Resistor	52
Figure 3.28: Power and Notification LEDs Wiring Diagram	53
Figure 3.29: Current Sense Amplifier Circuit with Gain 200	54
Figure 3.30: Example Arrangement of PCB Decoupling Capacitors at Minimised Distance to Microcontroller and Oscillator Layout.....	56
Figure 3.31: CAN Component Placement Adjacent to DB9 Connector for Minimised Noise Risk	56
Figure 3.32: PCB Variable Sized Trace Layout Between Stepper Driver and Motor Socket.	57
Figure 3.33: PCB Layout of Microstepping Mode Selection Headers with Extra Space Between Rows	58
Figure 4.1: Design and 3D Print of First Mechanical Mounting Shell	61
Figure 4.2: Remaining Screw Length Too short	62
Figure 4.3: Second Generation Mounting Shell Model and Print with Improvements.....	63
Figure 4.4: Arduino C Code Used to Read and Display Analog Voltage on Arduino Pin A0 Using Arduino IDE Serial Monitor	64
Figure 4.5: Serial Plotter Graph of Hall Effect Sensor Response at Minimum, Maximum and Neutral Magnetic Field Strength	65
Figure 4.6: Basic Motor Test Code with Variable Speed and Rotation Count Setting.....	66
Figure 4.7: Screwdriver Hand Piece with 4mm Hex Shaft and Socket End	69

Figure 4.8: Complete Flexible Shaft Coupler and Hex Socket Assembly	69
Figure 4.9: Stepper Motor and Shaft Assembly	70
Figure 4.10: Motor and Shaft Mechanism Successfully Screwed to Mounting Shell	70
Figure 4.11: 3D Printed Version of Custom Valve from Front and Top Views with Needle Separate and Inserted	71
Figure 4.12: 3D Printed Valve Needle with Ring Magnets Attached	71
Figure 4.13: 3D Printed Valve and Motor Mounted to Assembly	72
Figure 4.14: Hall Effect Sensor with Soldered Extension Leads	73
Figure 4.15: Hall Effect Sensor Installed into Mount Bracket with Header	73
Figure 4.16: 3D Printed Valve with Additional Custom Long Needle	74
Figure 4.17: Test Setup of Complete Test Rig Assembly	75
Figure 4.18: Sensor Output (raw, V) VS Distance (mm) from Fully Opened Valve.....	76
Figure 4.19: Sensor Output (raw, V) VS Distance (mm) from Fully Opened Valve with 1.5RPS Motor Rotation	76
Figure 4.20: Sensor Output (raw, V) VS Distance (mm) from Fully Opened Valve with 3RPS Motor Rotation	77
Figure 4.21: Failed First Attempt at 3D Printing Custom Valve with 1mm Threads	78
Figure 4.22: Second Valve 3D Print Attempt with 2mm Threads	79
Figure 4.23: Blank Custom PCB.....	80
Figure 4.24: Stacking Two 8.2K Resistors to Compensate for Missing 4.12K Resistor R9	81
Figure 4.25: PCB1 and PCB2 Demonstrating Dip Switch S2 and Shorted Switch Terminals.....	82
Figure 4.26: Pin 16 Wired to Grounded Power Pad Instead of 3.3V Pin 15.....	83
Figure 4.27: Aftermath of Removing Unwanted Ground Trace and Shorting Reset Pin to Adjacent Charge Pump Pin	84
Figure 4.28: Removed Ground Trace from Remaining Blank Boards to Prevent Issue from Reoccurring ..	84
Figure 4.29: Sensor Value and Voltage Over Valve Motion Through Custom PCB Filtering Hardware	85
Figure 4.30: Main Setup Code for Microcontroller Hardware and Macro Definitions	87
Figure 4.31: Software Initialisation Subroutine	88
Figure 4.32: Wait 1 ms and Wait 5 μ s Delay Subroutines	89
Figure 4.33: Step_motor Function to Move Motor at Specified Distance, Direction and Speed	90
Figure 4.34: Set Microstepping Size with Corrections for Forbidden Values	90
Figure 4.35: Save Contents of ADC Buffers to Data Registers	91

Figure 4.36: Sensor Response (mV) from Valve Full Sweep Motion91

Figure 4.37: Non-Linear Sensor Response More Significant for Slow Motion.....92

Figure 4.38: Valve Cycle Sensor Response for 1, 1/2, 1/4 and 1/16 Microstepping93

Figure 4.39: Current Sensor Output Signal Excessively Noisy95

Figure 4.40: Main Loop Structure and Error Check Subroutine to Analyse for Faults and Flash Red LED If True96

Figure 4.41: Yellow LED Illuminates During Motor Operation and Red During Fault Condition.....97

Table of Tables

Table 2-1: Summary table of motors and important characteristics	14
Table 3-1: Microstepping Mode Per Mode Pin Configuration	45
Table 3-2: Table of Microcontroller Pins Used with Their Purpose, Port Location and Description.....	49
Table 4-1: Voltage Response of Hall Effect Sensor with Different Magnetic Field Conditions	64
Table 4-2: Table of Rated and Observed Characteristics of Project Stepper Motor	67

1. INTRODUCTION

Suspension system dampers have become ubiquitous in modern road vehicles to improve vehicle performance and driver comfort. These suppress oscillations induced in the suspension springs and attempt to prevent road disturbances from affecting the driver. When tuning a damper, there always exists a trade-off between vehicle performance and driver comfort, which must be analysed contextually with the expected operating conditions of the vehicle. Furthermore, there exists a trade-off between damper control and cost, where more powerful active systems incur significantly greater expense than the simpler fixed passive systems. The research conducted here aims to assess these trade-offs by providing semi-active control to variable passive dampers by the addition of a custom electromechanical device. This will be performed in partnership with Supashock Advanced Suspension, who are an Adelaide-based company in the automotive industry, conducting research into advanced suspension systems.

1.1 Background

Shock absorbers in suspension systems act to absorb the shock produced by road surface irregularities. Conventional passive shock absorbers use a combination of a spring and mechanical damper applied to each vehicle wheel, which respectively absorb road disturbance energy and suppress vertical vibration oscillations. These passive systems are typically configured for general vehicle use on general road conditions, which makes them simple and cost-effective solutions for cheaper vehicles (Dixon, 2007). A consequence of their simpler design is that passive dampers have little to no damping control, which contributes to reduced performance in road situations optimally requiring harder or softer damping. Figure 1.1 displays the diagram of a basic passive suspension system, where the vehicle shock absorber consists of the main spring and damper. This can be converted to a semi-active system by providing dynamic control of the damper damping coefficient C_{damper} .

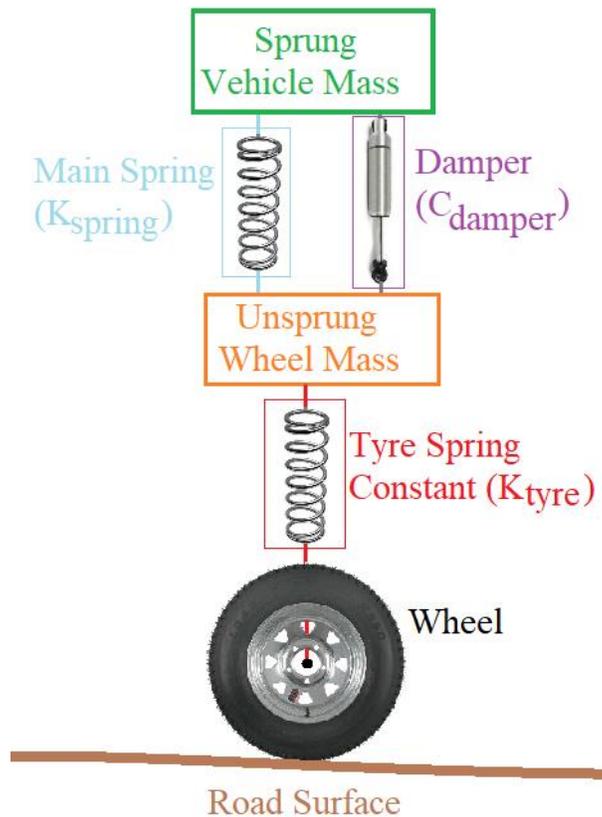


Figure 1.1: Diagram of Conventional Quarter-Car Passive Suspension System

Shock absorber dampers are defined as passive, semi-active or active depending on their power usage. Passive dampers require no external power to be added to the system and are typically comprised of only mechanical components (not electronic). These are the cheapest and simplest to engineer, but are the least flexible form of damper (Storey, 2011). Active dampers allow the highest level of control but are also the most expensive in terms of cost, power and computation. These allow for a high level of control, with application of any level of force desired. Due to their cost, such systems are typically only used on very high-end vehicles such as race cars. Semi-active suspension systems bridge the compromise between active and passive, with lower complexity and power consumption, but less control of damping characteristics. Unlike active suspensions, semi-active cannot produce actuation force, rather they can only control the magnitude of the road disturbance energy absorbed by the damper. The attraction of Semi-active dampers active dampers is due to their much lower power consumption. Ivers and Miller (1991) describe that they should “require no more power than a headlight”.

Semi-active damping approaches the generic tuning issue by providing a mechanism to adjust certain characteristics of the damper such as damping fluid (magnetorheological) or mechanical components (variable orifice) to dynamically vary the damping ability (Potter, Neild, & Wagg, 2010). This technique has become very popular in the automotive scene as it provides a more effective trade-off to control and power consumption compared to passive and fully active systems respectively. There exist a variety of control mechanisms to dynamically adjust suspension damping, however these are typically designed to be implemented as is, rather than retrofitted to an existing passive damper. This means that converting a vehicle suspension system from passive usually requires replacement of the suspension strut. This project involves the development of a proof-of-concept electromechanical system, which will form the technology basis of a semi-active retrofit upgrade for Supashock passive dampers.

A simple internet search reveals an abundance of purchasable conversion kits to change semi-active suspensions to passive, but there appear very few products and very little information to perform the reverse operation, as desired for this project. One notable example is the Tein Active EDFC (Tein, 2018), which contains up to eight stepper motors that simply bolt onto the adjustment point of the shock absorber strut. These are then controlled remotely via a custom control unit, which mounts inside the driver cabin and allows for remote adjustment of the damping parameters. Such a design provides an effective approach to this problem, however this unit is only compatible with the shock absorbers provided by Tein themselves. It is therefore necessary that a similar mechanism be designed and assembled specifically for the adjustable dampers used at Supashock.

1.2 Project Scope and Objectives

The primary objective of this research is to design and construct an electromechanical device to demonstrate electronic control of shock absorber damping parameters. This will be performed by designing a custom PCB to provide electronic control and a custom mechanical and hydraulic test rig to demonstrate the successful functioning of the PCB, as displayed in Figure 1.2.

The designs will be performed with a focus towards high flexibility of testing and high ease of access for test engineers. Rather than implementing this design on one specific shock absorber damper, the custom demonstration rig will be built to allow for easy future adaptation towards any vehicle shock absorber unit. The final outcomes will serve as a proof of concept of electromechanical control of damping parameters, which can be used by Supashock to further their research into advanced semi-active suspension systems. The

outcomes of this project will be focused specifically towards control of a single damping parameter and will not provide research into any high-level vehicle or suspension control systems.

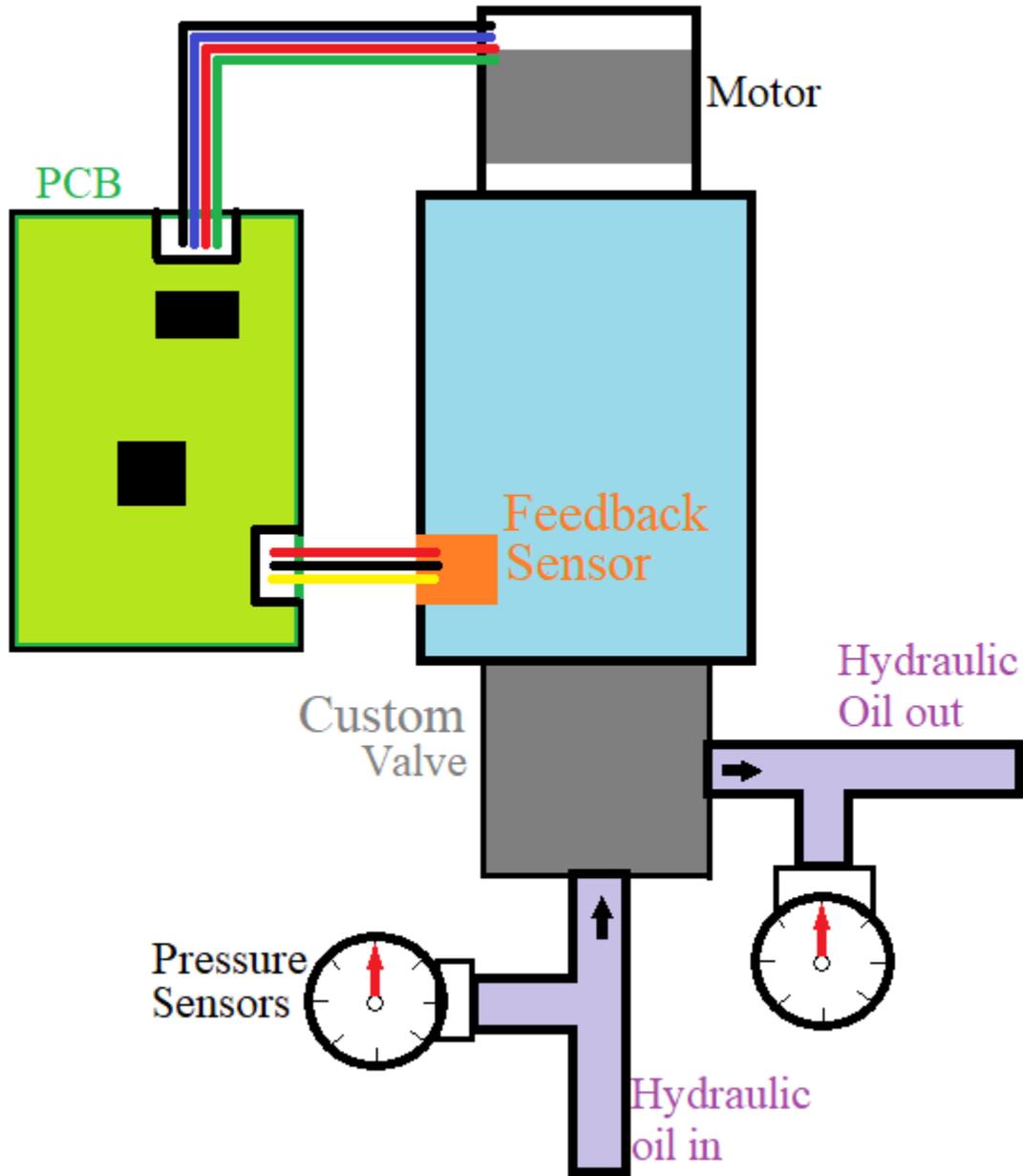


Figure 1.2: Primary Objective Test Assembly Layout of Demonstration Rig and Custom PCB

2. LITERATURE REVIEW

This report provides an analysis of existing technology to form a basis for the design of a motor-based proof-of-concept device to provide electronically adjustable control of shock absorber damping. This will include an analysis and comparison of existing semi-active control mechanisms to provide an argument for motor control specifically. Additionally, this report will investigate and compare the common motor types for such an application and their corresponding driver circuitry. Finally, an analysis of motion sensors will be performed to determine the most appropriate option for motor distance feedback.

2.1 Variable Control Mechanisms

This project involves the design of a motor-based electromechanical device to provide dynamic control of shock absorber damper characteristics. As such, it is useful to explore the alternative solutions to provide an argument for a motor specifically. This section will briefly investigate existing alternative semi-active damper mechanisms and describe their relative advantages and disadvantages.

2.1.1 Flow Control

There has been a wide variety of research and development performed towards fluid controlled semi-active systems (Els, Uys, Snyman, & Thoresson, 2005), (Wagner, Liu, & Liu, 2000). These typically use a solenoid to control the flow of air or hydraulic oil in the damper via a variable orifice or valve. Such mechanisms are advantageous in that they can perform very fast switching to different damping modes, allowing for very fast response to the user setting. However, these have the downside that they typically only allow on-off style operation, rather than continuous damping selection. An example of this operation is the experiments performed by Els et al. (2005), which used a solenoid valve on a Land Rover Defender 110 suspension system. This demonstrated fast switching between comfort and performance modes with a maximum delay of 100ms. However, this was limited to the fixed configurations within these modes, as the mechanism was effectively designed as two different passive spring and damper packs in one suspension system.

Wagner et al. (2000) experimented with semi active and active controlled damper designs for isolating vibration from vehicle seats. The semi-active mechanism made use of a solenoid-controlled valve to adjust the size of a fluid flow orifice between two piston chambers. This used an electronic controller to configure

the solenoid position based on the relative pressures of both chambers. Their active damper design used a DC motor, gear train and rack drive to directly control the chamber pressures, and therefore flow rates. This is noteworthy as Supashock dampers use a variable orifice oil flow mechanism like that described in their semi-active design, and this project aims to apply a motorised controller system like that described in their active design. The DC motor, gear train and rack drive design may be useful to consider for the motor driving mechanism within this project, however the research by Wagner et al. (2000) was more focused towards the development of electronic control systems, rather than the mechanical design, hence further investigation should be performed on this aspect.

Despite their widespread usage, solenoids have several disadvantages that should be considered. Due to their on-off switching nature, they can only allow a very small degree of control over the damper characteristics; typically, by switching between different preconfigured damper settings. Additionally, solenoids have a nominal position to which they return in a power-down condition. This means that during vehicle operation, if power to the suspension system is interrupted, the solenoids inside will default to fully open or fully closed and cause unexpected softening or hardening of the damper. This can produce a dangerous vehicle response depending on the driving circumstances. An example of this would be a driver setting a vehicle to “sporty” suspension mode, before driving more aggressively. If the suspension system power is interrupted and suddenly completely softens as the driver is taking a corner, the vehicle may become unstable and lose control.

These issues can be counteracted by using a motor instead of solenoid to drive a valve. Motors can be easily configured to move to a specific position or number of rotations, which allows for very precise control of damping parameters. Although, this would require that a damper mechanics allow for variable tuning, rather than switching between preconfigured parameters. In the case of this project, the damper technology used in Supashock research makes use of the variable orifice technique, which can be varied across a continuous damping range.

Additionally, if a motor driven valve encounters a power interruption, the motor can hold the damping ability at the previous known value with a degree of detent torque. The magnitude of the detent torque will vary based on the motor type, size and gearing configuration, and should be engineered specifically to suit the force encountered by the valve; namely due to oil pressure. This is preferable over to the solenoid default position response as a power interruption causes the damper to simply revert to a passive mechanism at its previous value, rather than forcing a sudden change to a potential unsuitable value.

2.1.2 Rheological

Another commonly used semi-active suspension technology is the rheological damper. This operates by using magnetically (MR) or electrically (ER) charged particles suspended in a solvent, which respectively change their viscosity upon application of magnetic or electric fields (Storey, 2011). Much research has been done towards the science and applications of magnetorheological fluids in damping and other applications. Jolly, Bender, and Carlson (1999) published a research paper detailing the chemical properties of MR fluids and appropriate applications such as vehicle suspensions, seat vibration isolation and seismic activity damping for buildings.

Unlike valve controlled semi-active mechanisms, MR control can operate over a spectrum of damping magnitudes based on the current provided to the control coils. This is advantageous as it allows for a much larger degree of damping control compared to the on-off mechanisms of solenoid valves. Additionally, MR has other advantages over valves such as “the absence of mechanical valving (for flow control) in the damper, insensitivity to impurities penetration, fluid stability, and long operational life” (Gordaninejad & Kelso, 2000).

Like solenoid valve systems, MR dampers have a nominal non-energized value that they occupy when no power is applied to the system. Due to the improvements in MR fluid technology, however, this nominal value can be designed to produce a damping force equal that of a passive fluid damper (Gordaninejad & Kelso, 2000). This means that such damper technology can maintain a fail-safe condition during power interruptions, however this still involves defaulting to an externally defined nominal value. As this project is focused towards developing a retrofitting electromechanical extension to existing hydraulic damper technology, the properties and engineering behind MR dampers are not the primary focus of this paper and are provided for background purposes only.

2.2 Motors

As the project involves the design of a motorised mechanism of damper control, it is important to investigate the available motor options. To allow for ease of commercialisation, this project will make use of existing and purchasable motors, rather than designing a new motor specifically for this application. This section will provide a comparison between the major motor types and describe their advantages and disadvantages relative to this project application.

2.2.1 Stepper

The first thought of many when choosing a motor for fine rotating position control is a stepper motor. These typically use rotors containing many small “teeth” along the rotor edge, which rotate to align with the energised coils of the stator and represent individual steps of the motor. There exist three major construction types of stepper motor known as variable reluctance (VR), permanent magnet (PM) and hybrid. The specifics of the internal mechanisms are not relevant to their application within this project as a motor will be simply purchased from a third party, however it is important to note their application properties.

*Figure 2.1 has been removed due to
Copyright restrictions*

Figure 2.1: Stepper Motor (Pololu, 2019)

Due to their ease of construction, VR stepper motors are typically lower cost than their counterparts but are very noisy at high speeds (T. Khan, Taj, & Ijaz, 2014). In a commercial product, these aspects would both need to be carefully considered, however in a proof of concept design such as this project, these are less important points as they can be improved in commercial revisions of the concept. Permanent magnet stepper motors use radially polarised permanent magnets within the rotor which align to the oppositely polarised electromagnets within the stator. This allows them to be manufactured with short step size and high torque/volume ratio, which would be advantageous in applications needing high torque and precision within a small space (T. Khan et al., 2014). Combining the principles of VR and PM motors, hybrid motors contain both a toothed rotor and permanent magnet rotor axis. This causes increased detent, holding and rotating torque compared to the previous two types, which is more desirable for this project to hold the motor position steady in a power-loss situation.

Stepper motors are also available with multiple different winding configurations. Unipolar winding configurations use two windings with centre taps. These are typically grounded, and the windings are individually energised to control a single coil at a time. Conversely, bipolar winding does not use centre taps, which causes current to drive an entire winding per cycle instead of half a winding. As a result, bipolar motors can produce approximately 30% more torque for the same volume, but require more power and more complicated driver circuits (Condit & Jones, 2004). In the early stages of a proof of concept design such as this project, it is considered more important to demonstrate the successful functioning of the basic device before optimising the power characteristics. As a result, it would likely be more effective to begin with a bipolar motor configuration as there is a more torque available for testing a wider variety of mechanical loads.

In the later stages of the project, it may be advantageous to switch to a unipolar design for improved power reduction and miniaturisation purposes.

Figure 2.2 has been removed due to Copyright restrictions

Figure 2.2: Unipolar (left) and Bipolar (right) wiring configurations for permanent magnet rotor motor (top)(Condit & Jones, 2004)

Stepper motors are inherently open loop controlled, but typically have very reliable motion so do not require an encoder or other feedback sensor. Such feedback sensors are usually only required when the accuracy of the motor position is critical, and the motor is likely to slip, such as with large torque loads or sharp acceleration spikes. As this project involves determining the relationship between the motor position and the shock absorber oil pressure, it is important to fully account for motor slip. In this case, a distance sensor will be equipped to the motor setup to directly measure the movement distance and ensure any slip is accounted for. An analysis of sensor options will be explored further in Section 2.3 of this paper, with a focus on the application to this project.

The process of driving a stepper motor relies on the accurate timing of a sequence of DC pulses to each motor coil. This causes sequential activation of the coils and causes the rotor to rotate. There are many different integrated circuit stepper motor drivers available for purchase with many different features and low cost. Such features include high continuous current capability (STMicroelectronics, 2018), high resolution microstepping up to $1/256^{\text{th}}$ of a step (STMicroelectronics, 2017) and various different decay modes (Texas_Instruments, 2014). Full step control of a stepper motor uses two square waves in quadrature arrangement to drive the motor coils, with the direction of the motor rotation determined by the lead-lag relationship of the waveforms (Hopkins, 2012).

Figure 2.3 has been removed due to Copyright restrictions

Figure 2.3: Driving Waveform of full-step drive for stepper motor (Hopkins, 2012)

Given that pre-packaged motor driver integrated circuits are relatively cheap and widely available for PCB usage, it was recommended that one be purchased for use in this project, as opposed to custom designing the driver circuit components. It is desired that the proof of concept unit designed within this project can accommodate a variety of motor types and sizes for increased flexibility, therefore it is advantageous to purchase a stepper driver IC to suit the highest power motor that may be used.

Additionally, as the project involves fine control of the motor position, it may be necessary to implement stepper motor microstepping. Typically, generic driver ICs allow for a small degree of microstepping such as half or quarter, but higher degree capabilities such as $1/16^{\text{th}}$ may need to be selected here as this requires more complicated electronic processing. Alternatively, small valve movements could be achieved by making use of down-gearing in the mechanical design rather than with the motor control.

2.2.2 Brushless DC

Brushless DC motors (BLDC) have gained popularity in recent years for applications such as hobby aircraft, white goods and electric vehicles (Keeping, 2013), as shown in Figure 2.4. These provide several advantages over standard brushed DC motors such as quieter and more efficient operation and reduced maintenance requirement, but also require more complicated dedicated driver circuitry (Zhao & Yu, 2011).

These operate by sequentially energising a set of stator coils with specific timing sequences to produce a rotating magnetic field and synchronously attract magnets within the rotor. BLDC motors are available in one, two or three phase coil arrangements, with three-phase being the most common (Digi-Key, 2016). This means that three separate signal lines must be provided by a driver circuit and individually energised to provide control.

*Figure 2.4 has been removed due to
Copyright restrictions*

Figure 2.4: Brushless DC Motor (SeedStudio, 2019)

Due to their requirement for timing logic, BLDC motors contain rotational position sensors to provide timing triggers for the coil energising process. This can be in the form of hall effect sensors within the motor chassis or back-emf detection through the motor driver circuit. A three-phase BLDC motor will typically use three hall-effect sensors, which can provide feedback for 60° of movement (Zhao & Yu, 2011).

This may be useful as a servo feedback mechanism; however, this is a much lower angular resolution than the 1.8° step size from a conventional stepper motor. This describes that very small angle control such as that required for this project would be better performed by a stepper motor. This limit may be bypassed, however, by making use of down-gearing in the mechanical design, where one rotation of the motor contributes to a fraction of a rotation of the damper adjustment system. In this case, it may be unnecessary to utilise such small angle control as offered by a stepper motor, and the coarser BLDC angle control may become more suitable. The construction of BLDC motors means they are advantageous for higher speed and smoother motion compared to stepper motors, which may be beneficial in this project if a strong down-gearing mechanism is implemented between the motor and damper control.

The multi-phase coil arrangement of the BLDC motor requires that appropriate circuitry be applied to perform the coil switching function. This is often performed using a series of PWM-driven MOSFETs such

as that seen in Figure 2.5. The PWM signals are combined with signals from the hall effect sensors to provide accurately synchronised switching. Zhao and Yu (2011) provide a useful timing diagram (Figure 2.6) demonstrating the energising sequence for coils U, V and W based on hall effect signals a, b and c.

*Figure 2.5 has been removed due to
Copyright restrictions*

Figure 2.5: MOSFET PWM Driver Circuit for 3-phase BLDC motor (Digi-Key, 2016)

Alternatively, BLDC motors can be produced without hall-effect sensors for improvement in small or harsh environments and reduced electrical complexity. These then calculate signal timings based on motor coil back-EMF within the driver circuit (back EMF waveform, Figure 2.6). However, as the magnitude of the back-EMF is proportional to the speed of rotation, this mechanism is ineffective at low motor speed such as start-up conditions (Zhao & Yu, 2011).

*Figure 2.6 has been removed due to
Copyright restrictions*

Figure 2.6: Three-phase BLDC Motor Hall Effect Coil Timings, Sensing and Hall Effect Sensing and Back EMF Timing Signals (Zhao & Yu, 2011)

Controlling a BLDC motor can be performed using a dedicated driver IC such as the Texas Instruments DRV8313. This unit offers three individually controllable MOSFET half H-bridges which can be used to drive a three-phase BLDC motor (or other loads) up to 2.5A. This does not contain sensing electronics for the hall effect feedback, however, so this should be externally calculated by a microprocessor.

2.2.3 Brushed DC

Brushed DC motors such as that shown in Figure 2.7 are one of the oldest and simplest types of electric motors. These contain wound coil rotors and typically permanent magnet stators. The wound rotors are

connected to a DC power source via internal contacting brushes, which periodically toggle the polarity of the coil windings as the rotor rotates.

*Figure 2.7 has been removed due to
Copyright restrictions*

Figure 2.7: Brushed DC Motor (RobotShop, 2019)

Unlike the BLDC and stepper motors, DC motors can be powered by simply applying a DC current to its terminals with no additional logic required. Additionally, the motor direction can be reversed by simply reversing the polarity of the power supply and the speed of rotation can be changed by varying the power supply voltage (Schweber, n.d.). This has the disadvantage though that DC motors cannot apply a powered holding torque, as being powered inherently causes rotation.

Due to their simple construction, DC motors are not inherently capable of performing complicated speed or position operations without additional components such as microcontrollers and encoders. Additionally, adding an encoder to a DC motor will allow for more accurate measurement of motor movement, but will not provide any powered holding torque at the desired position. This is different to stepper and BLDC motors which can naturally apply a holding torque by holding the motor coils charged at the desired position.

Although DC motors are cheaper and simpler than the alternative options, they would be less useful for this project as cost is not a significantly limiting factor and the importance of accurate position control is far more effectively achieved by stepper and BLDC motors. Additionally, this project requires the ability to move to and hold a specific position, which cannot be performed with DC motors.

2.2.4 Summary

This section contains a summary table of the main characteristics of the three discussed motors in relation to their application within this project. The motors are ranked for each criterion and any additional useful information is provided in the relevant cell.

Table 2-1: Summary table of motors and important characteristics

	Stepper	Brushless DC	Brushed DC
Position control	Excellent (1.8° steps typical with smaller microstepping often available).	Good	Poor, Good with added encoder
Speed Control	Excellent (controlled by varying stepping rate in microcontroller)	Excellent (controlled by varying coil voltage and current)	Good (controlled by varying applied voltage-PWM).
Max speed	Moderate	Excellent	Excellent
Power Consumption	Moderate-high	Low	Moderate
Complexity	High	High	Low
Durability	High	High	Low (brushes wear due to friction; brush carbon dust can contaminate motor internals)
Cost	Moderate	Expensive	Cheap
Additional hardware required	Driver switching	Driver switching, hall effect sensor processing	Encoder for precise operation
Key use case	Very precise position control	Precise speed control, Smooth angular movement (not stepping)	Cheap and simple prototyping, Low complexity systems
Additional notes			No powered holding torque

2.3 Distance Measurement

An important aspect of the design within this project is the ability to accurately measure the linear movement of a screw component within an oil valve. This is required to calculate the relationship between the motion of the control mechanism and the damper characteristics via oil pressure changes. As this project involves the use of a motor to induce linear motion, the displacement can be measured either directly via a linear motion sensor or indirectly via angular sensors on the motor shaft. Additionally, adding a distance measurement sensor allows the motor control loop to be closed; accounting for any motor slippage or step missing in the motor driving process. This section will explore the advantages and disadvantages of various motor movement measurement techniques in relation to the goals of this project.

2.3.1 Encoders

A widely used rotation sensor in modern electronics is the rotary encoder. These exist in multiple different forms and functions, which have their own advantages and disadvantages. This section will explore the two major types of encoder and their potential use within this project.

2.3.1.1 Absolute

Absolute encoders use unique position value for each rotational position. These can use optical sensors with a binary grey code windowed disk (as seen in Figure 2.8) or a magnetic wheel and detector. Using a unique data value per position means that the position of the shaft can be simply read by the detector sensor. This reduces the need for complicated electronic processing, however this also increases the cost of production to manufacture such small slots and accompanying electronic sensors. Absolute encoders are commonly used in environments with reliability concerns, as the position memory is stored on the disk instead of the electronic memory. This means that the machine position remains known even if the device encounters a power interruption or the shaft is moved while it is turned off (Z. Khan, 2015).

*Figure 2.8 has been removed due to
Copyright restrictions*

Figure 2.8: Absolute Optical Encoder Slotted Wheel (Z. Khan, 2015)

Absolute encoders are available in single-turn and multi-turn forms. Single turn is simpler and cheaper to produce, and functions by simply overflowing the encoder data value on a new rotation. Multi-turn encoders are more complicated but can uniquely account for many rotations. These are often made by using multiple optical or magnetic channels, with various degrees of down gearing such that they complete full rotations at different rates. There are several active patents (Bielski et al., 1999; Strasser, 2000; Wong & Lee, 2008) which display variants of this design. The most recent publication, produced by Wong and Lee (2008) presents an encoder with sets of inductive emitter and receiver coils with appropriate down-gearing to detect up to 4096 shaft rotations.

Optical encoders operate using a combination of light source (LED) and photodetector positioned either side of the disk. As light is emitted onto the disk, it is either allowed or blocked based on the data slot grating. Using light as a transmission medium means that optical encoders are inherently susceptible to contaminants such as dust, dirt and grease. If a dirt cluster is large enough to block a low-bit slot, the device could misread the location and create a position error.

Magnetic encoders, however, use windings printed onto a PCB, which “are unaffected by foreign matter and their measurement performance is generally unaffected by offsets or mounting tolerances. This means that they do not require their own precision housings or bearing assemblies but can be simply screwed to the host system e.g. a motor or actuator” (Howard, n.d.). As this project involves the development of a proof-of-concept device to be implemented in a shock absorber, it is important to consider the environment that will be experienced by such a device. In this case, it is highly likely that the device will be exposed to dust, oil and other contaminants, which means that a magnetic encoder would be the most suitable choice if an encoder is used in the build.

2.3.1.2 Relative

Relative encoders function using a slotted disk which produces pulses of light as the shaft rotates (as seen in Figure 2.9). Unlike absolute encoders, these do not inherently hold information regarding the position of the shaft, however this can be calculated by using a microcontroller to count the pulses as they occur from a known position. Typically, incremental encoders include a separate home track containing a single datum such as a magnet and hall effect sensor, which indicates a full rotation (Howard, n.d.). When an incremental encoder is first powered, it must cycle to the home position before the shaft position can be accurately known. Due to their simpler construction, incremental encoders are cheaper than absolute encoders, but are susceptible to loss of data in the event of a power interruption.

*Figure 2.9 has been removed due to
Copyright restrictions*

Figure 2.9: Incremental Optical Encoder Slotted Wheel (Z. Khan, 2015)

In this project it is necessary for the planned electromechanical device to survive power interruptions with minimal risk. As a result, it is most likely that an absolute encoder will be more effective than an incremental encoder at angle sensing due to its inherent data retention ability. Additionally, measuring shaft angle can be performed more easily with an absolute encoder, as it can be simply retrieved from the sensor. Furthermore, incremental encoders need additional counting hardware or software to measure the pulses from the home position.

2.3.2 Potentiometer

Potentiometers (pots) are very commonly used sensors that operate by producing a continuous variable resistance between two pins with predefined maximum known resistance (such as 10k Ω). These are available in both linear and rotary forms and function electrically as a variable voltage divider circuit. Their simple design means they can be purchased very cheaply and with almost any desired maximum resistance value. Common applications of potentiometers include servo feedback for DC motors and volume adjustment knobs on audio systems (Todd, 1975). A potentiometer would be a very cost effective and simple motion sensor to measure motor movement for this project, however there are several disadvantages which must be considered.

The most common form of potentiometer is a single-turn rotary “gang” pot as shown in on the left in Figure 2.10. These are typically very cheap and easy to acquire (\$6.44 from RSComponents (2019a)) but would not be immediately suitable for this project as it requires measurement of multiple rotations. If such a pot were to be used in this design, it would need to be down-gearred such that the entire multiple rotation motion of the device causes only 300° of rotation on the pot. An alternative solution to this would be to use a form of multi-turn pot such as that seen on the right in Figure 2.10, which can produce unique resistance values over multiple rotations. These are typically more expensive than single turn units but could simply be connected to the motor shaft without any down-gearing required.

*Figure 2.10 has been removed due to
Copyright restrictions*

Figure 2.10: Single Turn (Left) and Multi-Turn (Right) Potentiometers (RSComponents, 2019a), (RSComponents, 2019b)

Unlike rotary encoders, potentiometers require component contact to operate. This means they are more likely to wear out over less time, which affects their accuracy and reliability. Additionally, like optical encoders, potentiometers are susceptible to contamination by grease and dust, which may cause unexpected resistance values. As the pot wears over time, the friction caused by motion may also generate dust from the internal mechanism. The rightmost unit displayed in Figure 2.10. has an expected lifespan of one million rotations, which is significantly smaller than the minimum 300 million rotation lifespan of optical encoders (Ganssle, 2012). In this project, it is expected that a full valve motion will occupy multiple cycles, which could easily add up to failure over long term use of the device.

2.3.3 Linear Variable Differential Transformer (LVDT)

Linear Variable Differential Transformers (LVDTs) are linear motion sensors which operate based on inductive coupling between primary and secondary coils through a magnetically permeable core, as shown in Figure 2.11. The coils are typically mounted to a stationary component and the core can move linearly; varying the magnitude of magnetic flux captured by the core. These sensors are advantageous as they require no contact between the sensing elements, which greatly improves their lifespan (National_Instruments, 2018).

LVDTs typically require an external AC voltage source with a frequency between 50Hz and 25kHz to generate the magnetic flux, however DC equivalents can be purchased which have an onboard oscillator and supporting electronics. Due to the presence of these additional electronics, DC LVDTs are more limited by temperatures than AC equivalents (Sensorland, n.d.).

This should be considered further as shock absorber struts often encounter high environmental temperatures. This means that if an LVDT is chosen to perform sensor measurements, a trade-off will need to be considered between temperature constraints and circuit complexity to generate an external AC signal. The use of an LVDT may be performed in this project to detect the linear component of the oil valve motion as it is rotated by the motor.

Figure 2.11 has been removed due to Copyright restrictions

Figure 2.11: Basic Operating Diagram of LVDT (National_Instruments, 2018)

2.3.4 Hall Effect Sensor

Linear analog hall effect sensors can be used as distance measurement devices by producing a voltage proportional to the magnetic field strength incident on the sensor. These have advantages like those of the LVDT, where they can use non-contact operation, which improves their lifespan. Additionally, measurement of magnetic flux is highly resistant to contaminants such as grease, dust and moisture, which are typically common risks in automotive environments such as this.

Another advantage of this sensor for this application is that it can provide a simple measurement of absolute position without the need for memory or position cycling. This is performed by simply measuring the strength

of the magnetic flux, which would be provided by an external magnet mounted within the valve system. An added advantage from this is that the sensor mechanism is immune to power-loss events, which would otherwise cause a loss of data or need to rehome the valve position. This sensor could therefore provide a much simpler and more effective setup than some of the alternative possible sensors. For example, unlike potentiometers, a linearly moving magnet would not need to consider any additional gearing or homing mechanics and is immune to physical wear from contact.

One disadvantage of this sensor for use within this project is that a suitable magnet and corresponding mechanism would need to be separately sourced or designed. This means that the effectiveness of the sensor would depend on the strength, position and motion of the magnet introduced to the system, which requires careful consideration to ensure it responds efficiently. Additionally, careful consideration would need to be performed regarding the metals within the test rig, as some materials may disturb the magnetic flux before it reaches the sensor. This could cause nonlinear or non-functional results if the magnitude of the disturbance was too strong.

Figure 2.12 displays example linear hall effect components for PCB surface mount and thru-hole applications. These units can use an input voltage of 2.7V to 6.5V and have a typical voltage sensitivity of 1.4mV/Gauss, while typically consuming 6mA of current. These provide a linear analog voltage response for both positive and negative gauss, which means they can be used to effectively measure the distance from a magnet from either the north or south sides (Honeywell, 2015). This sensor provides multiple advantages over alternative sensors, so will likely be considered as part of the design for this project.

Figure 2.12 has been removed due to Copyright restrictions

Figure 2.12: Example Linear Analog Hall Effect Sensors (Honeywell, 2015)

2.3.5 Time of Flight Sensor

Another notable linear distance sensor that may be useful to this project is the VL53L0X Time-of-flight distance sensor as shown in Figure 2.13. This allows for measurement of linear distances up to 200cm with 1mm resolution and could be used to directly observe the linear movement of the oil valve. Such a sensor is advantageous over the alternatives as it does not require contact and will therefore not wear from friction. Conversely, because this sensor requires direct view of the sensing target, it makes it susceptible to

contaminants like dust and grease, which are likely to be present in this project's environment. These points will need to be carefully considered in the design stages of this project to determine the suitability of this sensor.

Another noteworthy advantage of this sensor is that it is significantly smaller than all the alternatives presented so far. This makes it significantly easier to adapt in a future revision to suit a more specific shock absorber design, since this can be easily applied without need for miniaturisation. As this project represents the design of a general proof of concept device, it is useful to consider the future adaptability of such a design, so it can be more easily adjusted in future to suit specific cases. The small size of this sensor and total PCB mounting allows for easy transfer of the design and circuit to suit specific shock absorber conditions including space constraints.

Figure 2.13 has been removed due to Copyright restrictions

Figure 2.13: VL53L0X Time of flight sensor with breakout board (LittleBird, 2018)

2.4 Existing retrofit products

Currently there are very few existing retrofit devices to modify existing passive shock absorbers into semi-active systems. Many options exist to perform the reverse operation; however, this typically involves replacing the electronic damper with an air spring or similar, rather than modifying the existing damper.

One of the most noteworthy existing semi-active conversion kits is the EDFC Active developed by Tein, displayed in Figure 2.14 (Tein, 2013). This contains a set of four stepper motors which bolt directly onto Tein suspension systems and directly drive existing adjuster fittings on the strut head. The front and rear motors are then linked into their own respective controller boxes, which connects wirelessly to the central control unit in the driver cabin. Using a variety of sensors, the system can be configured to respond automatically to vehicle acceleration and speed. Additionally, two of these systems can be linked to a single central controller unit to provide individual control of both bump and rebound characteristics on all four wheels (total 8 stepper motors).

Unfortunately, as the system is proprietary, there is little information regarding the construction of the stepper motor mounting mechanism and driving circuitry. This product is closely comparable to the goal of this project; however, this project is focused towards building a more generic prototype which can later be adapted

to suit any desired strut. Additionally, this project is focused towards existing Supashock strut designs, rather than Tein strut designs.

Finally, many of the total vehicle sensing and processing features provided within this product are integrated within the motor and vehicle control systems, rather than the motor mechanism. Although important, these topics are considered out of scope of this project and would be part of a future continuation project to further develop the vehicle control system and associated firmware.

Figure 2.14 has been removed due to Copyright restrictions

Figure 2.14: Tein EDFC Active Conversion Kit (Tein, 2013)

2.5 Summary

This chapter explores the existing literature of key background areas for this project. Analyses were performed in three major areas including alternative damper control mechanisms, theory of motor hardware and control, and distance sensors. It has also been observed that the literature regarding projects of similar goals is very limited. A single project competitor was identified, and their product analysed. The Tein EDFC semi-active damper kit provides very similar functionality to the goal of this project, however it is limited exclusively to Tein suspension dampers. Information regarding the exact construction of their mechanical and electronic systems is also very limited as it is a proprietary consumer product.

An analysis of potential motor options was performed which compared the properties of stepper, brushless DC and brushed DC motors. This found that the stepper and brushless DC motor options would be the most effective solutions as they provide far greater control capability and can provide a power holding torque to maintain a specified device position. Stepper motors in particular were considered as they can achieve much higher angular resolution with typical step size of 1.8° and further potential for microstepping. BLDC motors, however, were still considered a potential option as their lower angular resolution could be negated by mechanically down gearing the device, which would also make greater use of their improved speed control.

Additionally, a variety of distance measurement sensors were investigated for motor movement feedback capabilities. An analysis of rotary encoders was performed which found that absolute encoders would be more useful than incremental encoders to this project due to their inherent non-volatile position memory. Such a feature was considered important to account for power interruptions, which could otherwise cause

unpredictable behaviour in damper response such as instant damper softening or encoder rehomming. Magnetic encoders were compared to optical absolute encoders and were found to provide superior qualities. As optical encoders rely on light signals, they are susceptible to signal interference caused by grease, dust and other contaminants. Magnetic encoders, however, are immune to such issues. One of the key disadvantages identified with general consumer encoders is that they typically do not allow for counting multiple rotations. Several patents were identified for multi-rotation magnetic encoders using a variety of sensors with different gear ratios, which may be a suitable technique to explore if encoders are selected for the final design.

Potentiometers were also investigated as potential rotary position sensors. These are typically cheap and simple to implement but often have a much shorter lifespan than rotary encoders due to frictional wearing. One multi-turn potentiometer product was identified for possible usage within this application.

Two linear motion sensors were also analysed for possible use in measuring the linear motion of an oil flow valve. Both linear variable differential transformers (LVDTs) and a PCB-mounted time of flight sensor were explored, which were both found to be suitable potential options for this project. The time-of-flight sensor was advantageous in that it was significantly smaller than the LVDT and was PCB mounted, which would greatly simplify future adaptations of the device towards specific Supashock struts constraints. The LVDT, however, had the advantage that the AC powered versions contain no sensitive or optical electronics, which allows them to support much harsher environments including dust, grease and high temperatures.

Linear hall effect sensors were also investigated as distance measurement sensors, which could operate by detecting the linear motion of an external magnet. These had the advantage that magnetic flux is immune to disturbance by contaminants and frictional wearing. Additionally, as the motion information is contained within the position of the magnet, this sensor would not be affected by power loss or non-powered shaft rotation situations. The use of a magnet, however, meant that it was important to consider the materials being used in the rest of the test rig structure, as certain metals may negatively affect the magnetic flux and therefore sensor response.

Analysing the advantages and disadvantages of each sensing technique lead to the conclusion that the linear hall effect sensor and appropriate magnet would be the most beneficial system for this application. This was decided as they are immune to contact and contamination issues that affect many of the other systems, while being cheap and simple to implement.

Overall, an analysis of the supporting literature was performed, and the characteristics of potentially useful electronic components were investigated.

3. ASSEMBLY DESIGN

3.1 Objectives

This section details the design process of the overall test-rig assembly. This includes all components of the mechanical structure and the design process for the electronics and PCB. The primary goal of this process was to develop a generic testing rig to demonstrate the capability of a stepper motor to control the damping ability of a shock absorber damper. Here, the mechanical design provides the test rig and the electronic design provides the controller PCB, which together form the main components of the motor servo mechanism.

3.2 Mechanical

The mechanical structure of the device predominately formed the testing rig, which acts as a demonstration device to prove the successful functioning of the electronic components. This consisted of a custom-machined aluminium hydraulic valve, a mounting shell and the internal motor shaft mechanism. This section explores the design decisions and features of the mechanical part of the project. The final engineering drawings of the hydraulic valve components and assembly are presented in Appendix 1.

3.2.1 Hydraulic Valve Block

As the aim of the project was to provide electronic control of a shock absorber hydraulic process, it was considered important to develop a custom hydraulic valve mechanism to allow demonstration of such control. This decision was made as it removed the need to design the device to one specific shock absorber layout, which meant this system could be easily adapted in the future to suit one or multiple shock absorbers as desired by Supashock. The valve was designed to use a needle-and-seat configuration, which was considered simple and effective enough for this purpose. This section details the design of the valve block and the corresponding design features and choices.

Figure 3.1 displays the Autodesk Inventor model of the valve block design, which presents a view of the general valve shape, as well as the needle access hole and oil flow exit socket. For this design, standard 5-4F50X-S hydraulic fittings were chosen for the oil connections as they are commonly used and are readily available within the Supashock workshop. Although it is not visible within this image, there exists another hydraulic fitting on the bottom side of the block for oil flow input. The valve block is cylindrical with maximum diameter of 55mm and height of 60mm.

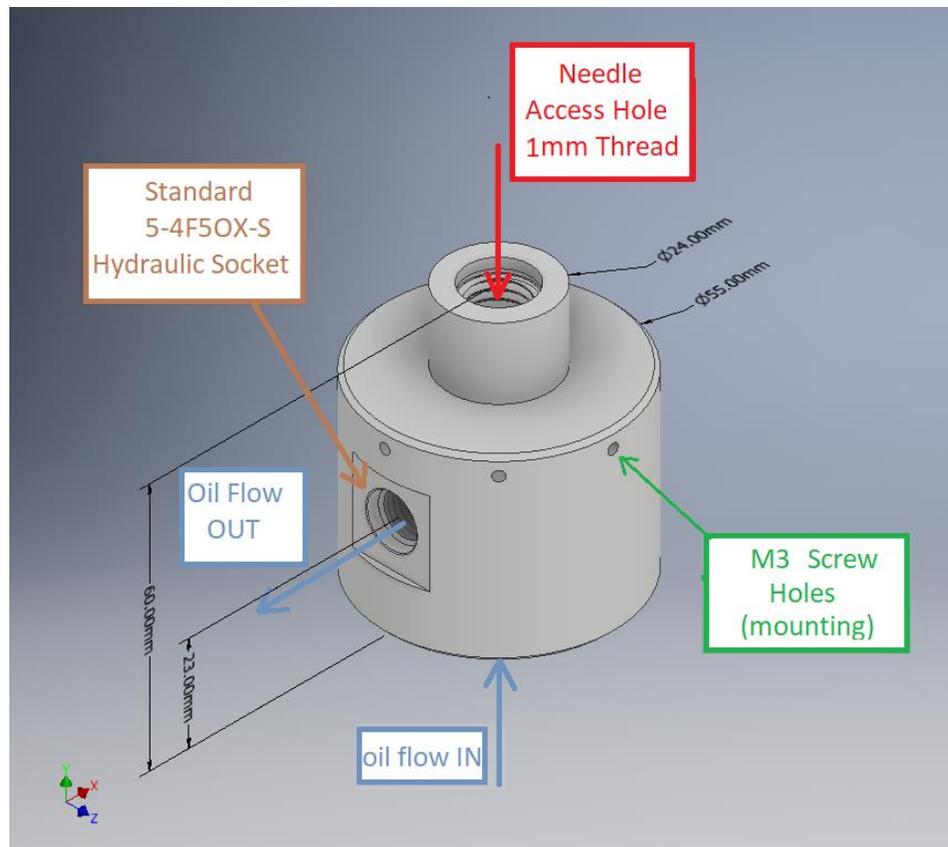


Figure 3.1: Autodesk Inventor Model of Custom Valve Block

The top side of the model shown in Figure 3.1 contains the access hole for the needle component, which provides the primary opening/closing mechanism to control the oil flow. This section of the valve block contains an access port with a 1mm thread for the needle motion and space for the motor shaft mechanism, as detailed in Section 3.2.2. The uppermost section of the block also has a reduced diameter of 24mm, which has been designed to securely mate with the mounting shell.

Figure 3.2 displays the half-section view of the valve block, which provides a view of the oil flow mechanics. The angle of the seat has been designed as 118° , which matches the angle of standard drill bits. This was chosen for ease of manufacturing and because the precision of the needle and seat mating mechanism was

not considered significantly important for this project. In this case, using a seat angle equal to that of standard drill bit angles means that the shaft can be simply drilled out once without needing multiple passes with different sizes. This reduces the cost and complexity of manufacture as it reduces machinist time and tooling requirements. Additionally, as this project exists simply to demonstrate the control of oil flow, it was not necessary to specify the exact mechanics of the such flow; instead this could be simply modelled later using oil pressure sensors and processed in software.

Also shown within this figure is a 15mm internal circlip groove, which exists to lock the needle to within the allowable motion range. This provides a simple mechanism to prevent the needle from unscrewing out of the thread, which could cause damage to the shaft mechanism and motor in the complete assembly.

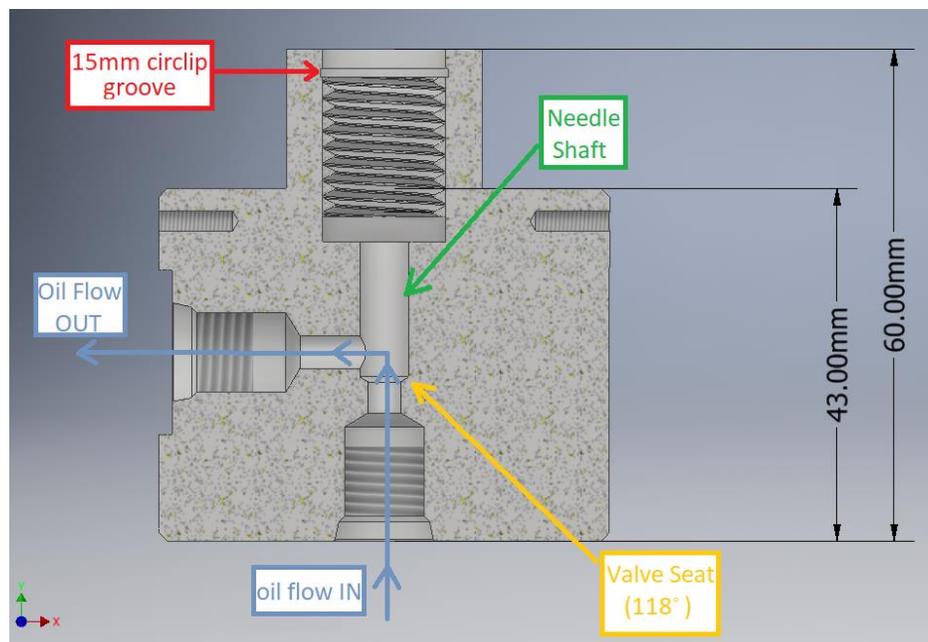


Figure 3.2: Autodesk Inventor Model of Valve Block Half-Section View

3.2.2 Hydraulic Valve Needle

Figure 3.3 displays the needle component of the hydraulic valve. This was designed to seal with the seat part of the valve block and fully allow or fully prevent the flow of the hydraulic oil. The needle tip was designed as 5mm long and with a 30° angle, which will mate adequately with the seat. The main shaft of the needle contains grooves for two O-ring seals, which seal the oil in the main flow chamber and prevent it from leaking into the drive thread and electronics area. In this case, double seals have been chosen to allow for extra protection if the first seal extrudes or fails. The seal chosen for this application is detailed in Appendix 2.

The red box in Figure 3.3 describes the 1mm pitch needle thread, which mates with the 1mm pitch thread described by the red box in Figure 3.1. This allows the motion of the needle to form the opening and closing mechanism of the valve. The total needle shaft length has been designed such that there is 4mm of allowable motion between fully open and fully closed, which equates to 4 full rotations. Additionally, a 4mm hex shaft section has been designed for the top of the needle, which interfaces directly with the motor shaft hex socket and allows for rotational coupling with linear sliding. This is described further in Section 3.2.4.

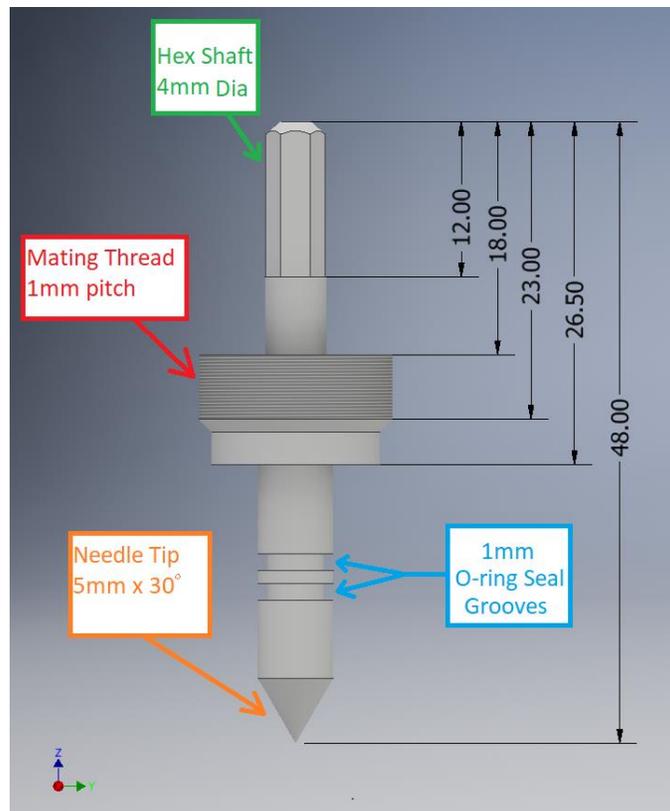


Figure 3.3: Autodesk Inventor Model of Valve Needle

3.2.3 Hydraulic Valve Assembly

Combining the two valve components into the full assembly produces the model shown in Figure 3.4. This displays the valve at fully opened (left) and fully closed (right) positions, with 4mm of displacement between them. Component colours presented in this figure are for visualisation purposes only and are not representative of the final product. It is worth noting that when the needle is in fully opened position, there still exists a section of space between the top of the needle thread and the circlip – as noted by the orange callout box. This is designed for a set of ring magnets, which form the basis of the sensor feedback mechanism and will be described further in Section 3.2.4.

The 1mm thread pitch of the rotating mechanism was chosen as it was the smallest safe size for this application as recommended by the Supashock mechanical engineering team. Such a fine-pitched thread allows for much finer control of the oil flow mechanism and reduces the torque requirement to drive the shaft against incident oil pressure. This also allows for better measurement capability of the motor to control the valve and oil flow, which directly addresses the goal of this project.

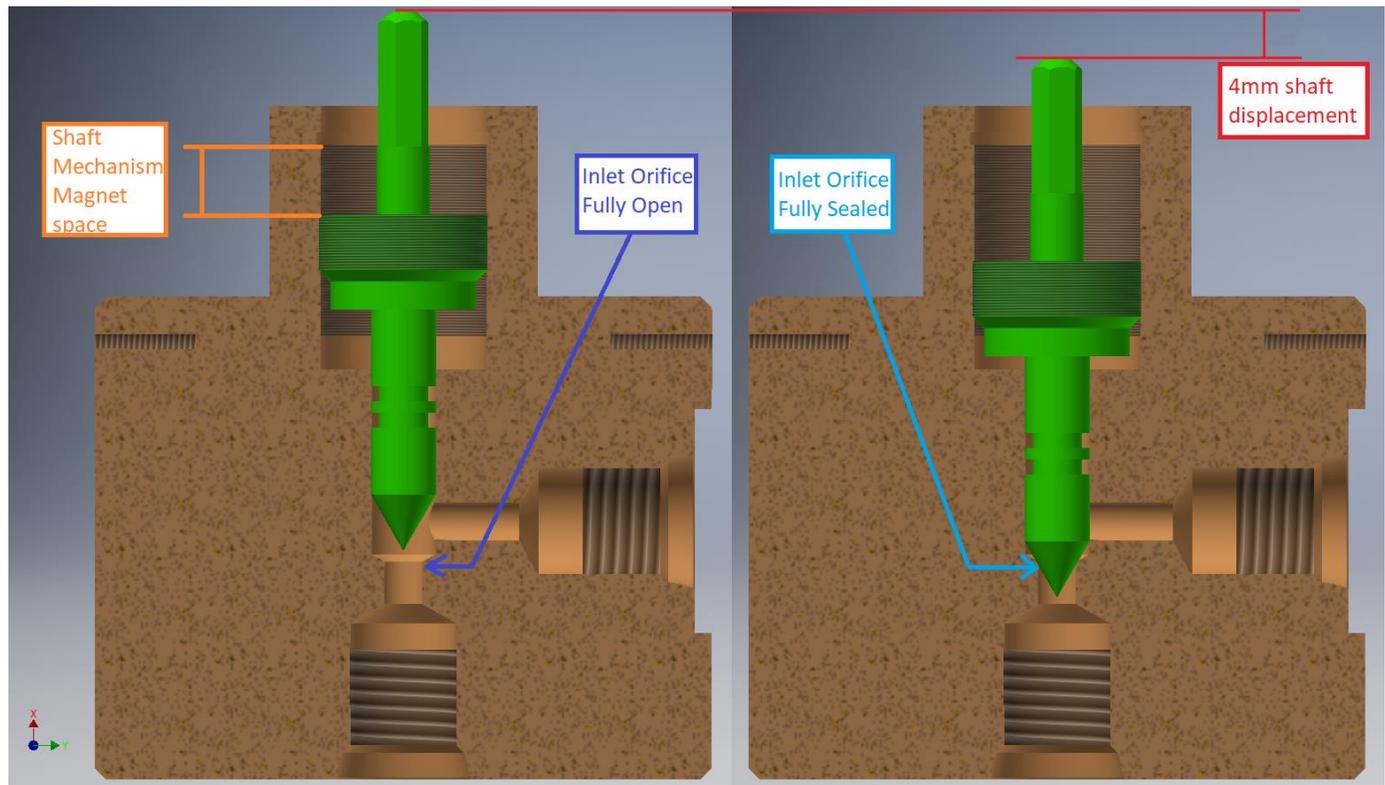


Figure 3.4: Autodesk Inventor Models of Valve Assembly Fully Opened (Left) and Fully Closed (Right)

Both major components of this valve have been designed such that they will be machined from aluminium 6061. This was chosen as it is a strong and lightweight material, which can withstand the force of the pressurised oil without being too bulky and heavy. In this case, the strength of the needle thread was the most critical component, where failure could release the oil pressure and potentially destroy the device. Another benefit of aluminium 6061 is that it has very low ferromagnetic interaction, which means the detection of magnetic fields from ring magnets on the needle shaft would not be affected by the external sheath of the block. This design demonstrates a simple hydraulic valve mechanism with 4 rotations between open and close, and designs to external control and measure the valve motion and position.

3.2.4 Motor Shaft

The process of driving the valve needle was performed with the combination of several small components. Ultimately, it was required that the rotational motion of a stepper motor with a 5mm diameter shaft be converted to a rotational motion of the needle with linear decoupling. Figure 3.5 displays the complete shaft assembly design, which details the conversion from the motor shaft to the valve needle.

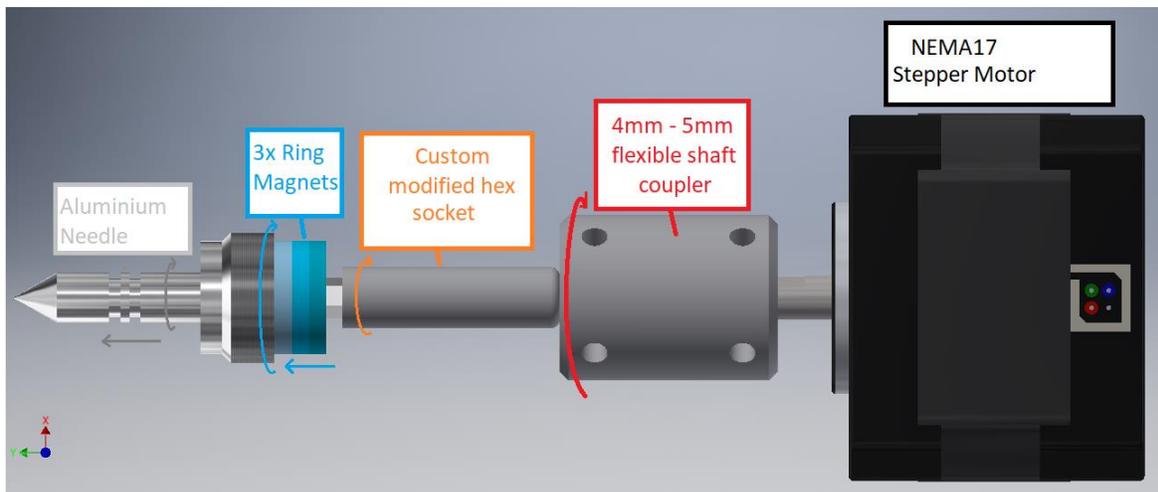


Figure 3.5: Complete Shaft Mechanism of Mechanical Assembly

For this project, the NEMA17 stepper motor size was chosen as it could provide suitable torque and power without being too bulky. Such motor sizes use 5mm shafts, which meant that a size-shifting shaft coupler was necessary to step-down the shaft size. In this case, the custom modified hex socket used a 4mm shaft, so a 5mm-4mm conversion was necessary. It was also decided that the shaft coupler should be flexible, which bends to correct any misalignment in the shaft angle.

This was important as a misalignment between the motor shaft and the needle shaft would cause inefficient driving and unnecessary extra friction, which could strain the mechanism. A suitable flexible shaft coupler was found for this project and cost \$7.99 AUD from eBay. Mounting between the coupler and the shafts is performed using two pairs of grub screws included with the coupler. Figure 3.6 displays the exploded assembly view of the motor shaft mechanism.

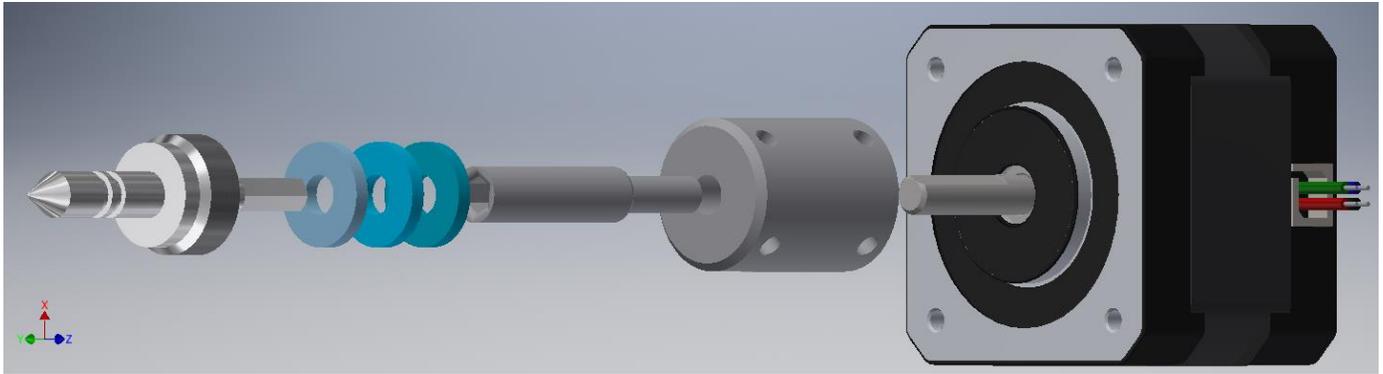


Figure 3.6: Exploded Assembly View of the Motor Shaft Mechanism

The next part of the shaft mechanism was the hex socket as shown by the orange box in Figure 3.5. This design was chosen as it creates an interface with the aluminium needle that provides rotational coupling but can slide freely in and out. The linear sliding capability was important as it allowed the needle to screw in and out of the valve block without shifting the position of the rest of the motor shaft. The other end of the hex socket was designed with a 4mm circular shaft which can mount directly to the shaft coupler via the included grub screws. This part was also designed to be machined from aluminium along with the valve components. Figure 3.7 displays the Autodesk Inventor model of the custom hex socket component.

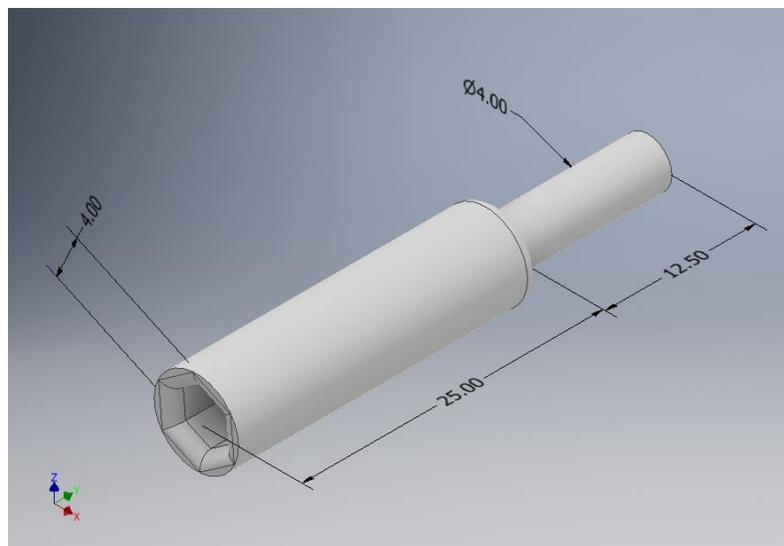


Figure 3.7: Custom Hex Socket Model

The final part of the shaft mechanism was the three ring magnets, as shown by blue in Figure 3.5. These were included to allow for sensor feedback of the needle position, where a fixed hall effect sensor could analyse the variation in magnetic flux as the needle screwed in and out. The magnets chosen for this experiment were OD13xH2xID5mm from Frenergy.com.au at \$4.25 each. The choice of three magnets was made as it allowed

for a good trade-off between good magnetic field strength without being too bulky. It was intended that the magnets would be held in place on the needle by the hex socket, although if necessary, they could be simply glued to the needle surface.

It is important to note that the magnetic flux vectors of the ring magnets are produced perpendicularly from the flat surface of the magnets, rather than along the ring radius. This was important as it allowed for constant magnetic field vectors at a given distance from the magnet centre, independent of the magnet rotation. This would allow the magnet sensor to read only the linear movement of the magnet, without any effect from the rotational movement.

The discussions presented here form the main choices behind the design of the mechanical shaft mechanism.

3.2.5 Mounting Shell

The process of mounting together the aforementioned mechanical components was performed using a custom-designed mounting shell as displayed in Figure 3.8. This component was designed to be 3D printed with PLA plastic, so could easily accommodate specific curves and shapes that traditional top-down manufacturing might otherwise struggle with. This unit provides the mechanical connections between the motor and valve with space inside for the shaft mechanism. Mounting to the valve is performed using eight 3mm screws, aligned radially. This allows for a strong connection between the bracket and the valve and opposes any torsional disturbances between the two components. The valve slot has been designed with multiple interlocking sheathes between the valve and mount shell, which attempt to prevent any shaft misalignment, and thus reduce motion inefficiencies.

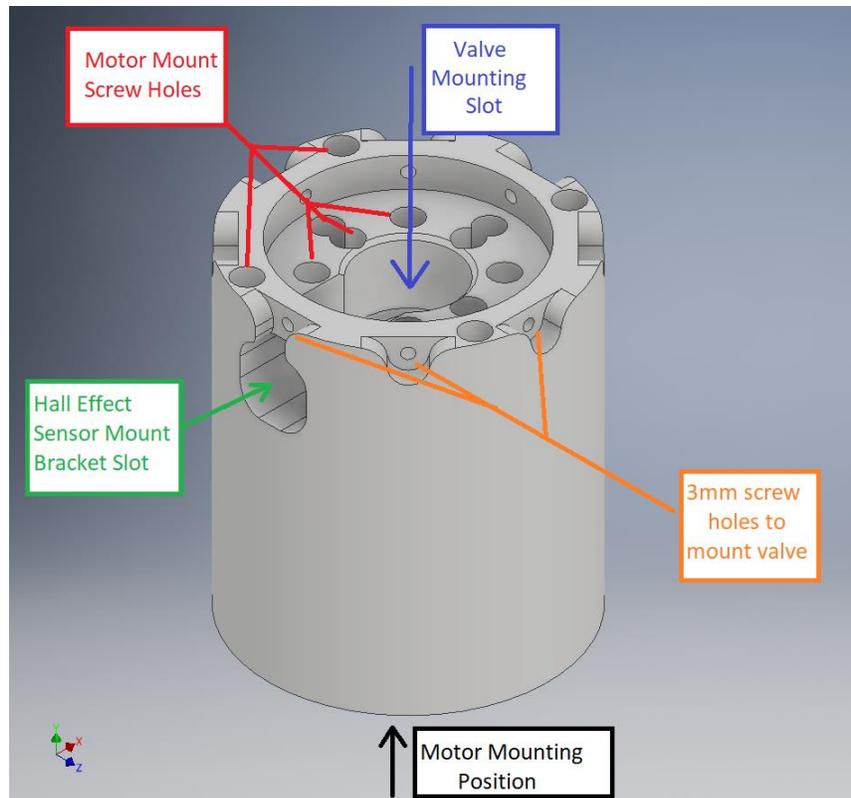


Figure 3.8: Custom Mount Shell Overall View

One of the requirements of the project was that it was highly flexible to different testing options. For this reason, the mounting shell was designed such that it could accept a variety of stepper motor sizes, as shown in Figure 3.9. Mounting hole positions for NEMA sizes 11 through to 23 were included in the design, which would allow the experimenter to easily change the motor size if desired. Although, different NEMA sizes can use different shaft diameters, which would require that different flexible shaft couplers be acquired.

The green box in Figure 3.8 describes the slot for the hall-effect sensor mounting bracket. This provides a means to mount the hall effect sensor in the mounting shell and complete the cylindrical look of the shell. The bracket is mounted to the shell using one of the 3mm screws that secures the valve. Since the bracket fits tightly into the slot, it was considered adequate that only the single screw supported the bracket in place.

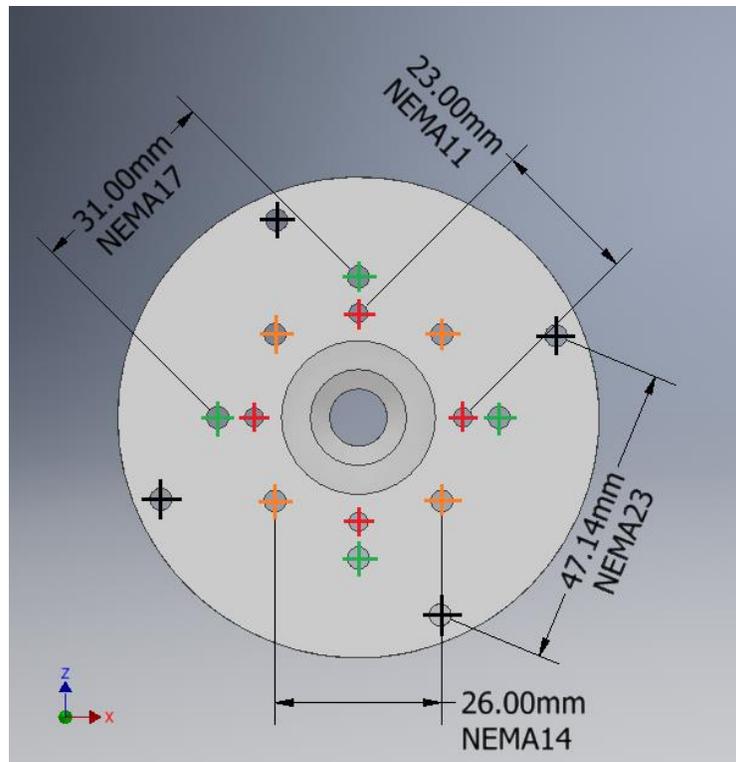


Figure 3.9: Bottom Face of Mounting Shell Highlighting Different NEMA Size Capabilities

3.2.6 Sensor Mount

Mounting the sensor to the shell was performed using a 3D printed sensor bracket as displayed in Figure 3.10. This was designed to fit flush in the mount shell and holds the sensor as close as possible to the inside of the assembly with wire access to the outside. The bracket contains a single lug for the 3mm mounting screw and a grab point for easy insertion and removal with pliers. The bracket design allows the sensor to be held in place by friction, although glue could be used if necessary.

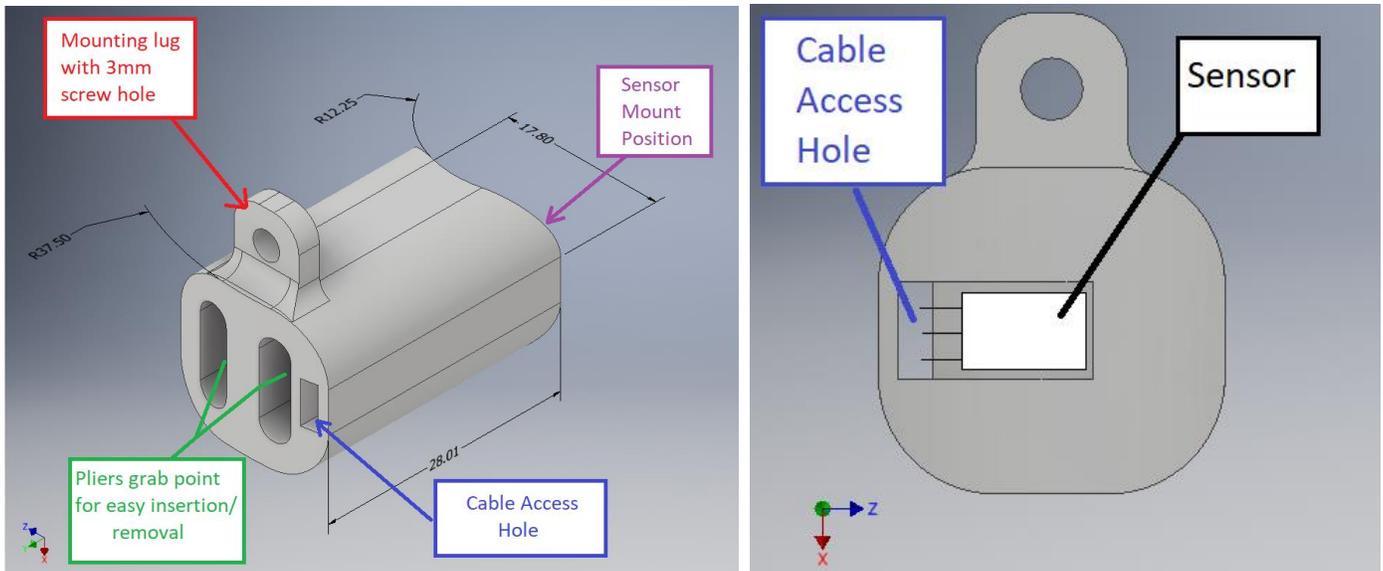


Figure 3.10: Sensor Mounting Bracket Orthographic View (Left) and Sensor Side View (Right)

3.2.7 Total Assembly

Bringing all the components together as a total assembly creates the device shown in Figure 3.11, with cross sectional view shown in Figure 3.12. This provides a complete testing rig which uses stepper motor motion to control the oil pressure in a custom hydraulic valve. The valve has 4mm of displacement between fully opened and fully closed, which is controlled by 4 full rotations of the stepper motor. The screw motion of the valve is formed by using a hex socket, which provides rotational coupling but allows linear sliding from the motor. Feedback of the valve position is provided using a hall effect sensor to measure the linear movement of three ring magnets on the valve needle.

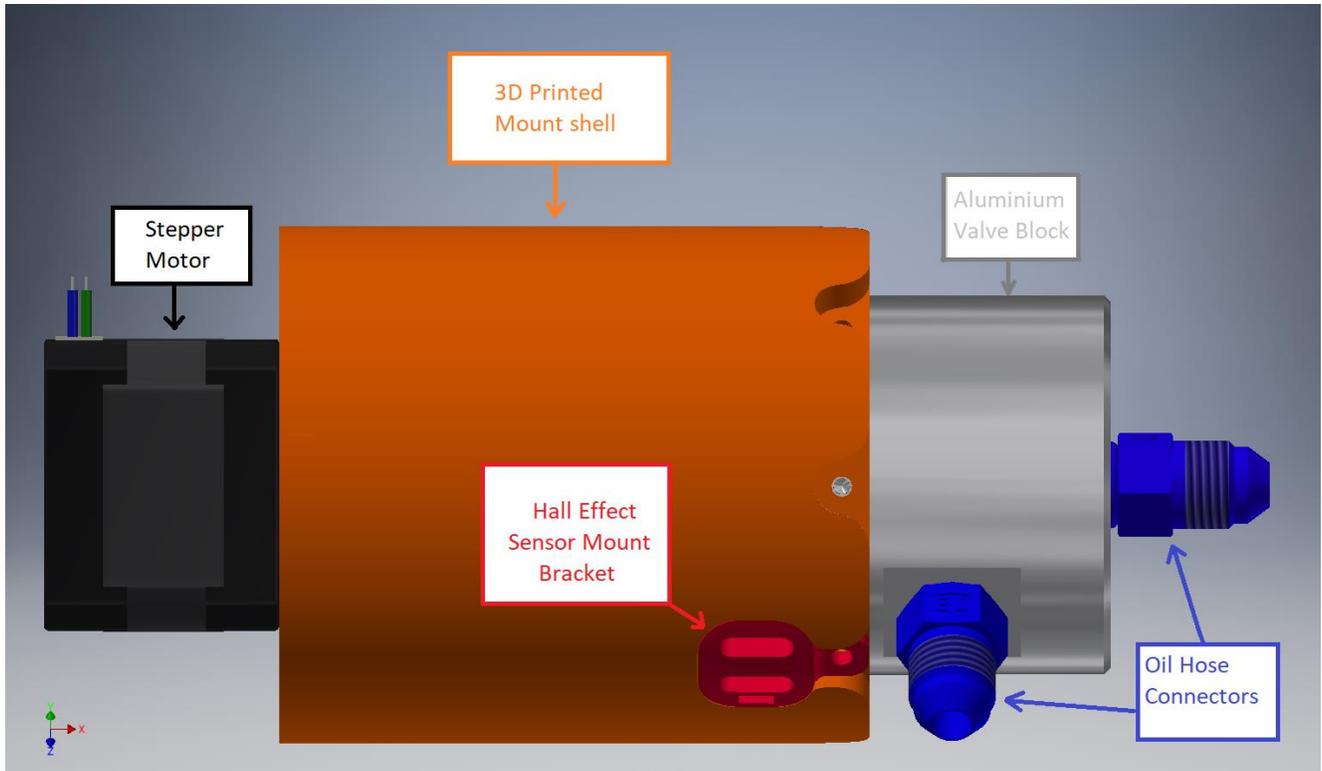


Figure 3.11: Total Test Rig Assembly External View

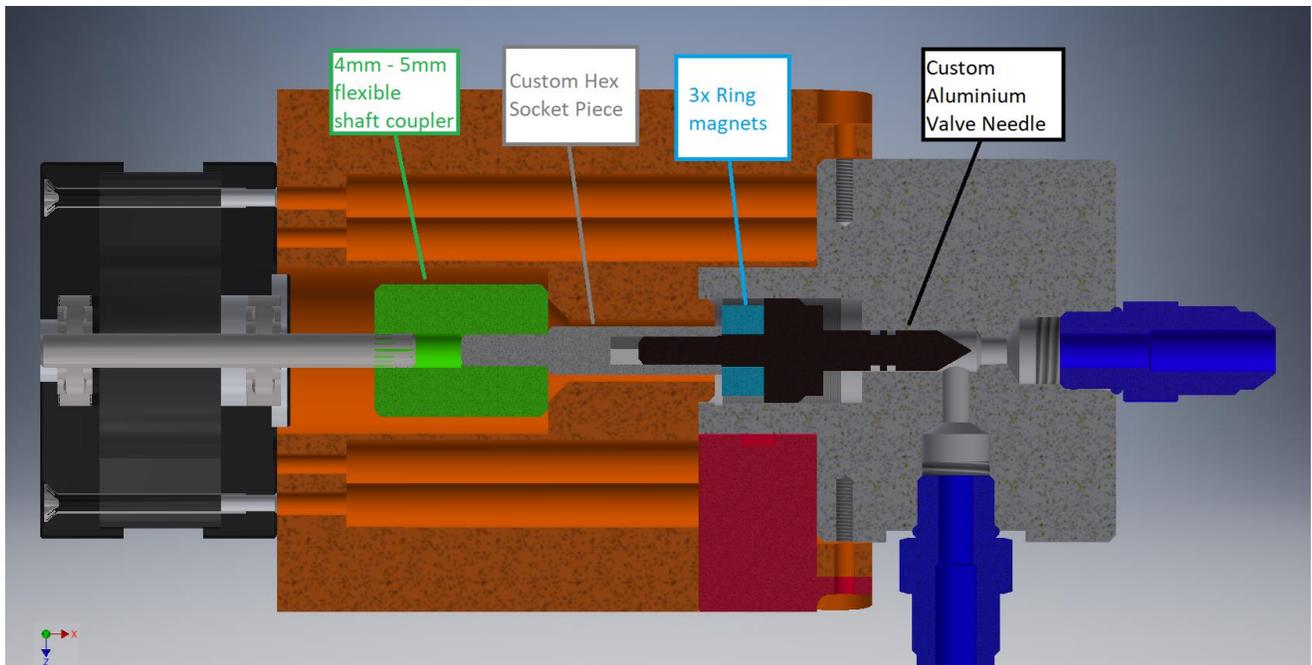


Figure 3.12: Total Test Rig Assembly Half Section View

3.2.7.1 Assembly Instructions

Assembling the rig from fully separate pieces should be performed using the following steps:

- 1) Connect 5mm – 4mm shaft coupler to stepper motor shaft. Secure using two of the provided grub screws.
- 2) Connect the custom hex socket piece to the 4mm – 5mm shaft coupler. Secure using the remaining two provided grub screws.
- 3) Insert motor shaft mechanism into appropriate side of the 3D printed mount shell.
- 4) Secure motor to 3D printed mount shell using four screws appropriate to motor size (3mm in the case of NEMA17). Use a long screwdriver to access the motor mount screws from the opposite end of the 3D printed mount shell.
- 5) Insert 3D printed sensor bracket into sensor mount slot. The sensor should come pre-mounted in the bracket with extension wires and a header socket.
- 6) Connect three ring magnets together coaxially via magnetic attraction.
- 7) Attach ring magnets along the hexagonal shaft of the custom valve needle. Place them flush against the surface of the needle.
- 8) Take needle-magnet assembly and screw it into the custom valve block until it fully closes the valve (i.e. the screw motion reaches the end).
- 9) Insert a 15mm internal circlip into the circlip groove in the valve block to prevent the needle from coming out.
- 10) Carefully insert the valve assembly into the 3D printed bracket assembly, making sure that the hex socket aligns with the hex plug of the needle.
- 11) Secure the valve assembly with eight 3mm screws, mounted radially. This will also simultaneously secure the hall effect sensor bracket.

3.3 Hall Effect Sensor

This assembly involves the use of an analog hall effect sensor to measure the displacement of the valve needle as it travels from fully opened to fully closed. For this purpose, the SS49E linear ratiometric hall effect sensor was chosen, as displayed in Figure 3.13. This provides a linear analog voltage output between $0.95V$ to $V_s - 0.95$ and can be powered by either a $5V$ or $3.3V$ supply rail. Additionally, this sensor operates reliably within a temperature range of $-40^{\circ}C$ to $100^{\circ}C$, which is within the expected operating range for this project.



Figure 3.13: SS49E Linear Ratiometric Hall Effect Sensor (Honeywell, 2015)

3.4 Stepper Motor

The choice of motor for this project was the 42BYGHM809 stepper motor shown in Figure 3.14. This was chosen as it allows up to $1.7A$ of current with 400 steps per rotation resolution. This therefore allows for strong but precise control of the valve and can theoretically resist a high valve oil pressure with strong holding torque.



Figure 3.14: 42BYGHM809 Stepper motor

3.5 PCB

As this project involves the use of a stepper motor and feedback sensor, it was considered important to design and manufacture a custom PCB to combine all the electronic functions, as shown in Figure 3.15 and Figure 3.16. This was performed using Altium Designer 2019 software and manufactured by PCBway.com. The custom PCB was designed as a two-layer board with 1.6mm FR-4 base material 0.4mil copper thickness, with a total board thickness of 1.7mm. The minimum track spacing was selected as 6/6mil and the minimum hole size was 0.3mm. The complete schematics for this PCB are displayed in Appendix 5 and the bill of materials is presented in Appendix 6.

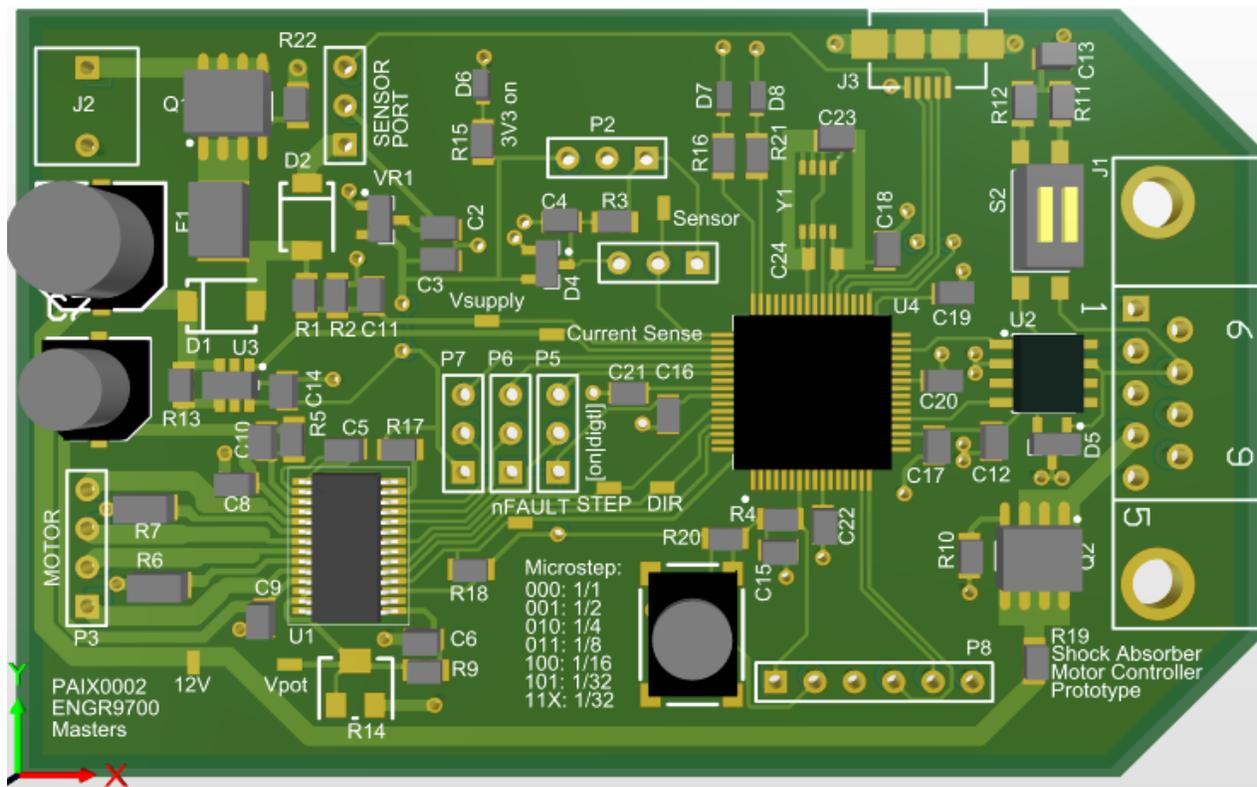


Figure 3.15: Altium 3D Model of Custom Driver PCB

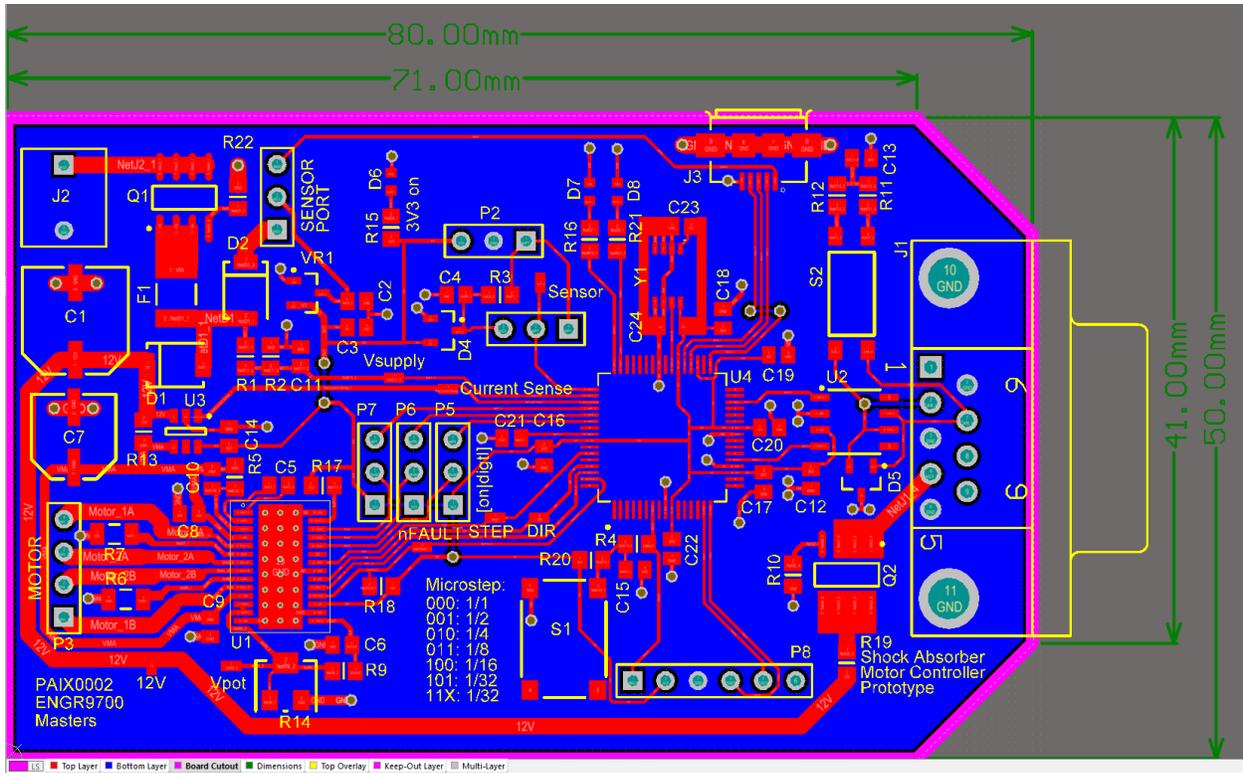


Figure 3.16: Top Layer Routing Model of Custom Driver PCB

3.5.1 Power

As this project is designed around the automotive industry, it was decided that the input power supply would be a 12V source with unspecified maximum current. This was designed to match the 12V level of car batteries, so this device or an adapted version of, could be simply wired in to an available source in a vehicle. Power is supplied to the device via a 2-pin screw terminal block, which allows for maximum flexibility for interfacing with a power source, as they connect via raw wires. Also included in circuit area is an IRF7416TRPBF power MOSFET to provide reverse polarity protection, which is explained further in Section 3.5.8.

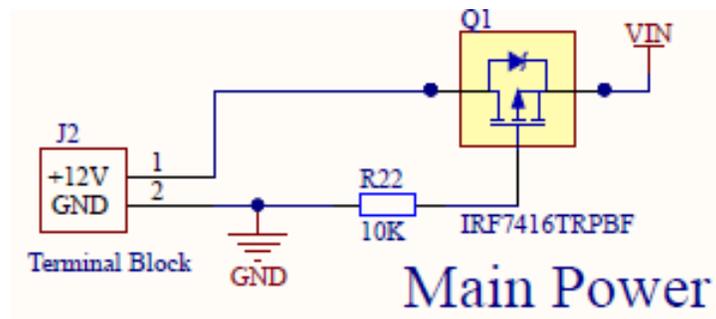


Figure 3.17: Power Supply Connection with Reverse Polarity Protection MOSFET

applies a 172.1Hz low-pass filter to reduce any higher frequency noise. These values are calculated using Equations 1 to 9.

$$V_0 = \frac{R_1}{R_1 + R_2} \times V_{in} \quad \{\text{voltage divider}\} \quad (1)$$

$$\therefore V_0 = \frac{(1.1K)}{10K + 1.1K} \times 12V \quad (2)$$

$$= 1.19V \quad (2.1)$$

Cut-off Frequency of RRC low pass filter:

R_1 has impedance $Z_1 = R_1$, R_2 has impedance $Z_2 = R_2$

$$C_{11} \text{ has impedance } Z_{11} = \frac{1}{2\pi f C} \quad (3)$$

consider Z_2 in parallel with Z_{11} :

$$Z_{eq} = \frac{1}{\frac{1}{Z_2} + 2\pi f C} \quad (4)$$

$$= \frac{Z_2}{1 + 2\pi f C Z_2} \quad (5)$$

consider the transfer function of a voltage divider:

$$H(f) = \frac{V_0}{V_{in}} = \frac{Z_{eq}}{Z_{eq} + Z_1} \quad (6)$$

$$= \frac{\frac{Z_2}{1 + 2\pi f C Z_2}}{\frac{Z_2}{1 + 2\pi f C Z_2} + Z_1} \quad (6.1)$$

$$= \frac{Z_2}{Z_2 + Z_1 + 2\pi Z_1 Z_2 f C} \quad (6.2)$$

$$= \frac{\frac{Z_2}{Z_1 + Z_2}}{\frac{2\pi Z_1 Z_2 f C}{Z_1 + Z_2} + 1} \quad (6.3)$$

cutoff frequency occurs when denominator = 2

$$\therefore \frac{2\pi Z_1 Z_2 f C}{Z_1 + Z_2} + 1 = 2 \quad (7)$$

$$\therefore \frac{2\pi Z_1 Z_2 f C}{Z_1 + Z_2} = 1 \quad (7.1)$$

$$\therefore f = \frac{Z_1 + Z_2}{2\pi Z_1 Z_2 C} \quad \{\text{cutoff frequency of RRC filter}\} \quad (8)$$

$$\therefore f = \frac{10K + 1.1K}{2\pi(10K)(1.1K)(1\mu F)} = 160.6\text{Hz} \quad (9)$$

3.5.2 Motor Driver

To drive the stepper motor, a DRV8825 motor driver was selected. This provides up to 2.5A per motor phase and is driven using simple step and direction pins. It can also perform microstepping up to 1/32 sized steps, which is well within the precision required for this project. Figure 3.19 displays the motor driving circuitry; based on the DRV8825 driver IC. This has passive components arranged according to the driver datasheet and digital lines connected directly to digital IO pin on the microcontroller. In this case, this driver was easily capable of driving the 1.7A motor initially chosen for this project, however it also allowed for other motors to be selected with higher current requirement if needed.

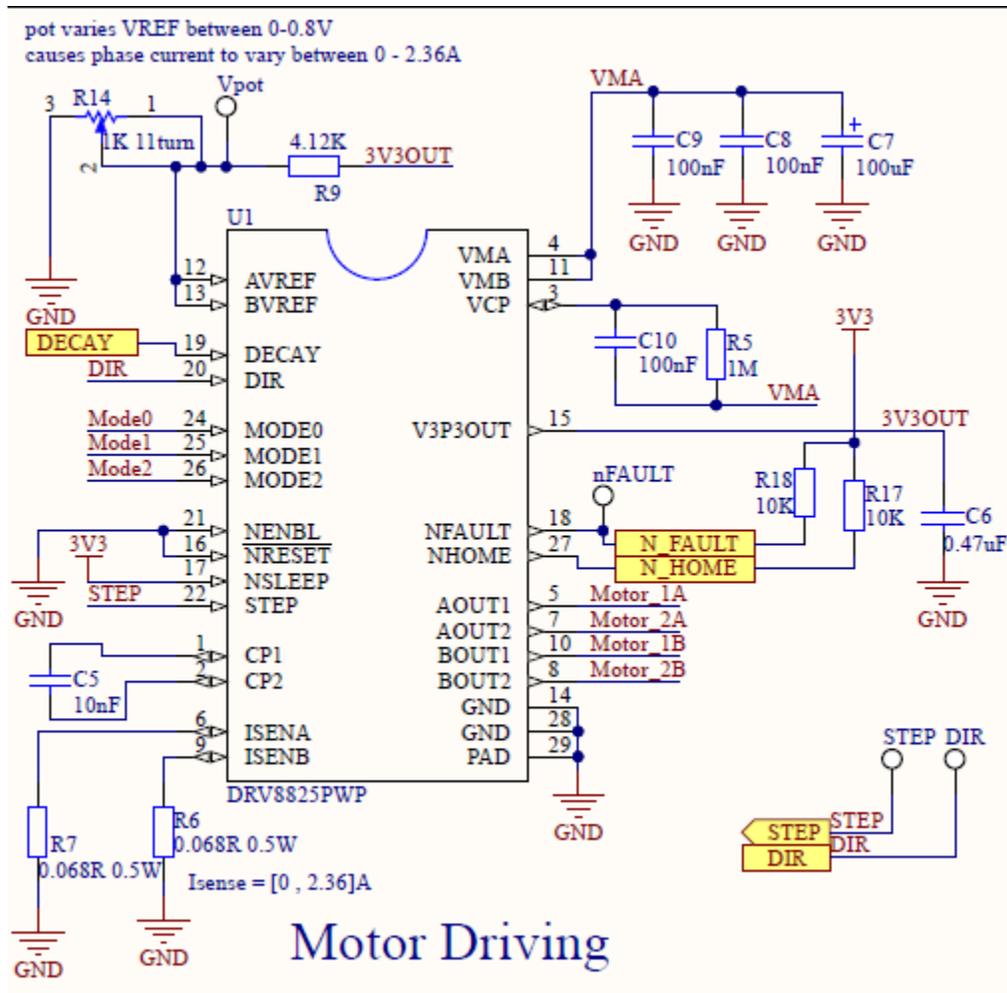


Figure 3.19: Motor Driver Circuit

As this motor driver can encounter high current flow during normal operation, it was necessary to consider the heat dissipation to prevent error and thermal shutdown. This was performed by using many small thermal vias beneath the chip to dissipate heat from the power pad to the ground plane on the other side of the board. Figure 3.20 displays the pattern of 0.2mm thermal vias under the driver chip, as recommended by page 31 of the DRV8825 datasheet (Texas_Instruments, 2014).

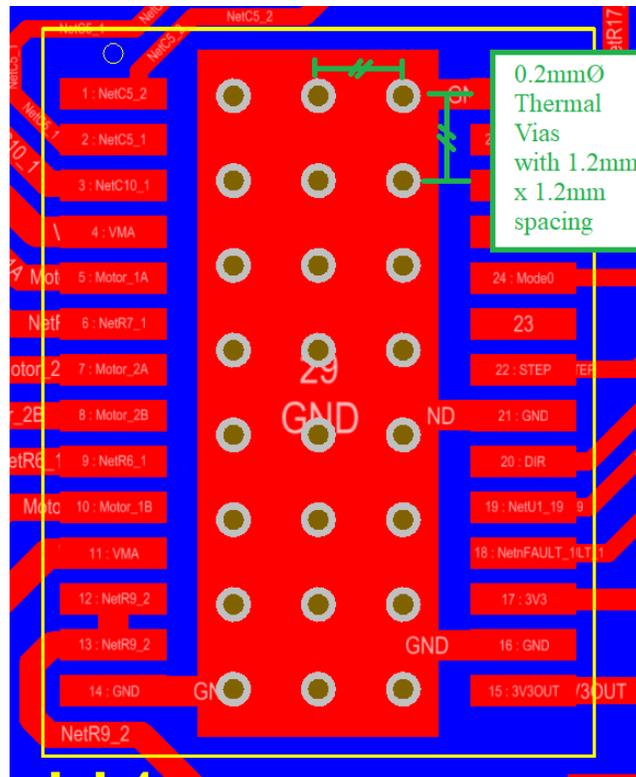


Figure 3.20: Thermal Vias Below Motor Driver Power Pad to Dissipate Heat to Ground Plane

Another advantage of this motor driver was the many safety features included, which reduces the risk of electrical faults and device failure during prototyping. Included within this package are overcurrent, undervoltage and thermal shutdown protections, which automatically drive the nFAULT pin low during a fault state. This pin has been routed directly to the microcontroller and a test point to simplify diagnostic and fault-finding procedures. The overcurrent protection engages during a short circuit to ground, supply or across the motor windings and remains until the device is reset. Undervoltage protection engages when the supply voltage falls below 7.8V and disables all internal circuitry until the voltage returns to above this value. Thermal shutdown engages when the temperature of the die exceeds 150°C, during which the circuitry becomes disabled and remains this way until the temperature falls below this value (Texas_Instruments, 2014). These features provide a high degree of protection, which is very useful during circuit prototyping.

3.5.2.1 Microstepping

The DRV8825 driver IC has microstepping ability up to 1/32 step size. This is configured by setting the logic level on the mode0, mode1 and mode2 pins according to Table 3-1. This was implemented in the circuit using male headers with one jumper per pin, which switched each pin between fully-on mode and microcontroller mode. Additionally, fully-off mode could be engaged by removing the jumper entirely as the

pins contain internal pull-down resistors. This configuration allowed all combinations of hardware logic-1, hardware logic-0 and digital control of the microstepping modes for maximum testing flexibility. Figure 3.21 displays the circuit layout for this selection system.

Table 3-1: Microstepping Mode Per Mode Pin Configuration

Microstep Size	MODE2	MODE1	MODE0
1/1	0	0	0
1/2	0	0	1
1/4	0	1	0
1/8	0	1	1
1/16	1	0	1
1/32	1	1	X

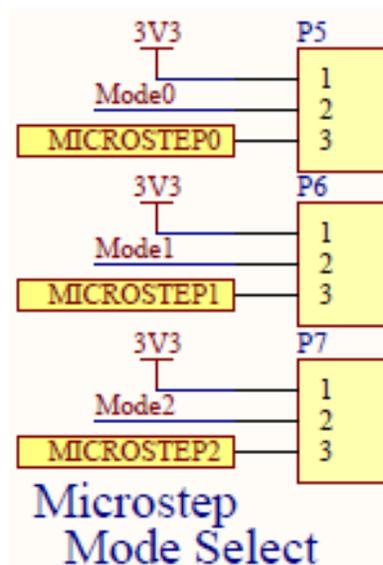


Figure 3.21: Circuit Layout of Microstepping Selection System

3.5.2.2 Current Selection

Selecting the output current for this driver is performed by controlling the voltage at AVREF and BVREF. This was performed using an eleven-turn 1K trimmer potentiometer (trimpot) and was configured as part of a voltage divider to provide a voltage between 0 and 0.8V. For simplicity, the trimpot used the 3.3V charge pump pin supplied by the chip itself on pin 15.

Page 12 of the DRV8825 datasheet describes that the output current relates to the reference voltage according to the relationship described in Equation 9. For this PCB, sense resistors of 68mΩ were chosen, which limited the maximum current to 2.35A as described in Equation 10. Furthermore, the power dissipation across the sense resistors was calculated as 0.376W in Equation 11, which was within the 0.5W maximum limit of the

selected resistors. Due to supplier stock restrictions, the selection of surface mounted resistors with 0.068Ω resistance and $0.5W$ power were limited to 1206 footprint size, rather than the typical 0805 used for most other passive components in this build. This was not considered a significant issue though; there was adequate board space to accommodate the larger size.

$$I_{\text{Chop}} = \frac{V_{\text{REF}}}{5 \times R_{\text{Sense}}} \quad (10)$$

$$I_{\text{CHOP(MAX)}} = \frac{V_{\text{REF(MAX)}}}{5 \times 0.068\Omega} \quad (11)$$

$$= \frac{0.8V}{5 \times 0.068\Omega} = 2.35A \quad (12)$$

$$P_{\text{sense}} = I^2R = 2.35^2 \times 0.068 = 0.376W \quad (13)$$

3.5.3 Sensor Filtering

Interfacing the hall effect sensor to the PCB was performed using a 3-pin keyed header plug, which provided 3.3V power, ground and signal lines externally to the sensor. The PCB was designed to include a basic RC low-pass filter and sensor voltage diodes to filter and protect the signal line. The cut-off frequency of the low-pass filter was calculated at 312Hz using Equations 12 and 13 below.

The diode D4 contains a double diode package with SOT-23 footprint and restricts the voltage of the sensor signal line to within 0-3.3V. This exists mainly to protect the microcontroller, where any voltage abnormalities outside this range could cause damage to the analog pins. Despite these protections, the circuit also included the option to bypass them entirely by switching a jumper between header pins on plug P1. This was included in case the sensor was found to produce worse results with filtering compared to without. This section also included a test point to allow the user to externally probe the sensor level without using the microcontroller.

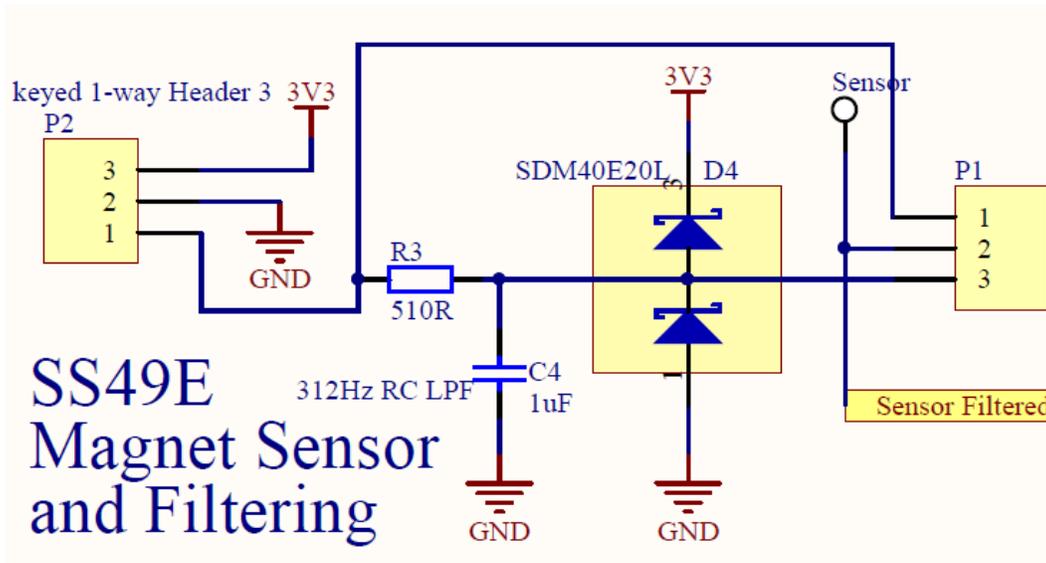


Figure 3.22: Hall Effect Sensor Filtering

$$f_c = \frac{1}{\sqrt{2\pi RC}} \quad \{\text{cutoff frequency of RC LPF}\} \quad (14)$$

$$\therefore f_c = \frac{1}{2\pi(510)(1 \times 10^{-6})} = 312.07\text{Hz} \quad (15)$$

3.5.4 Microcontroller

The process of combining all the electronic processes was performed using a PIC32MX530F128H microcontroller as shown in Figure 3.23. This is a 32-bit, 64-pin microcontroller which uses a TQFP package and contains many tools suitable for this project. This unit was chosen specifically as it was the smallest and cheapest PIC option that contained internal modules for both CAN and USB; removing the need to include them separately on the PCB. Device power is provided using the 3.3V power line, which is generated by the MCP1703 linear regulator.

Device clocking was performed using a LTC6930-8.00 silicon oscillator which provides an 8.00MHz clock signal to the microcontroller OSC1 pin. The choice of a silicon oscillator was made as they are typically more stable than crystal oscillators and involve a simpler design process to select the corresponding passive components. In this case, two 100nF ceramic decoupling capacitors were simply recommended by the oscillator datasheet with no additional calculations necessary. It was intended that the microcontroller would be operated at a 48MHz base frequency using a combination of the 8MHz clock and internal PLL multipliers, which is the highest achievable base frequency within the maximum limit of 50MHz for this device.

Also shown within Figure 3.23 is a circuit for a reset switch S1, which was provided in the form of a tactile micro switch and allows the user to easily force a device reset. This connects to the device nMCLR line with a 1.1K resistor to limit current flow into the microcontroller, a 10K pullup resistor to hold the signal high when the switch is not pressed and a 100nF capacitor to reduce high frequency switch bouncing noise.

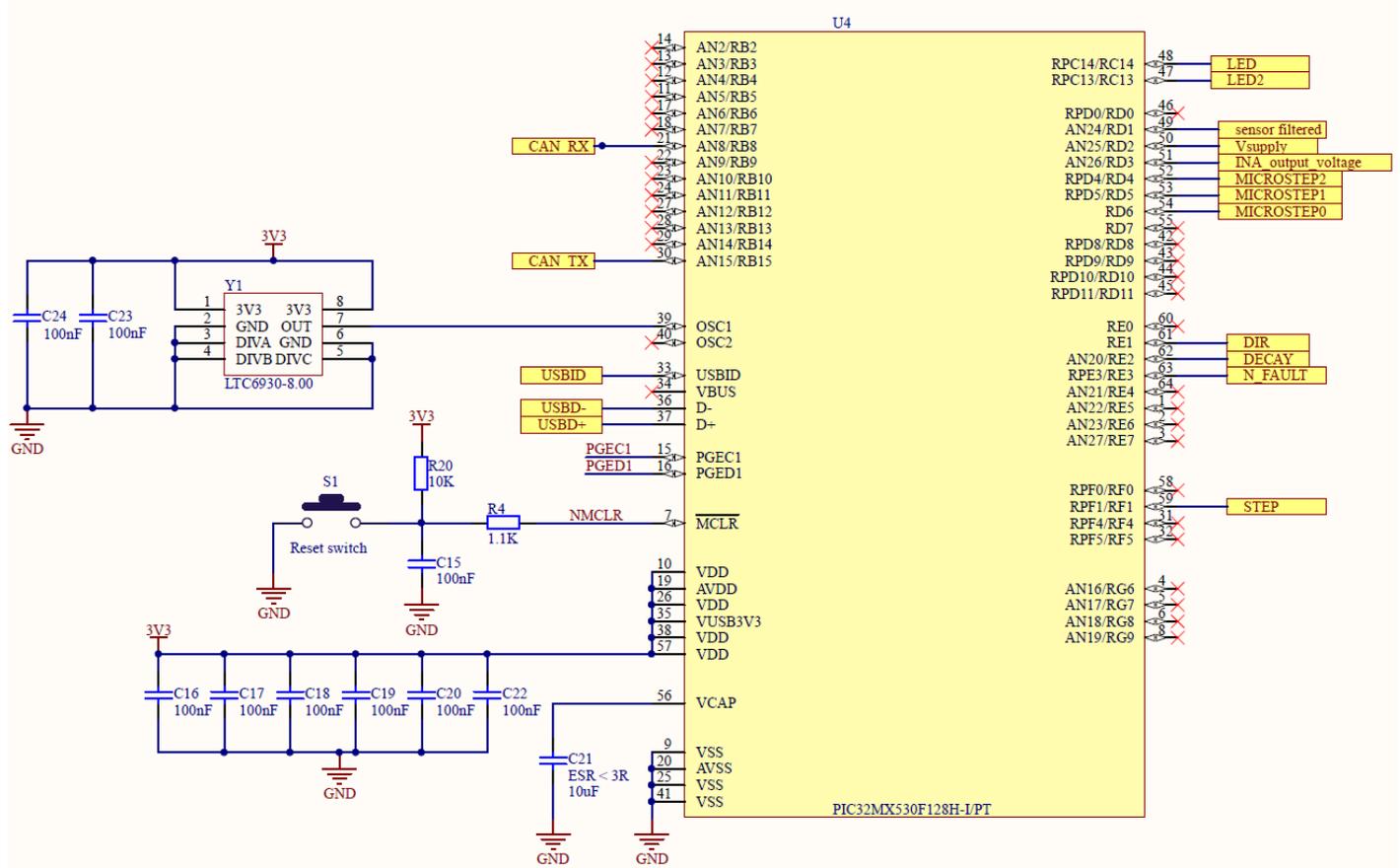


Figure 3.23: PIC32MX530F128H Microcontroller Wiring Diagram

Programming the microcontroller was performed using a generic header plug with wiring diagram shown in Figure 3.24. This was designed to match the wiring pinout provided by ICD3 and PICKIT3 programming devices, as they were both available and intended for use with this project. Additionally, the wiring configuration of the programming header allowed the programming devices to supply power to the 3.3V line on the entire board. This meant that the microcontroller and other 3.3V devices could all be powered and tested using only the programming port, which allowed improved flexibility in testing.

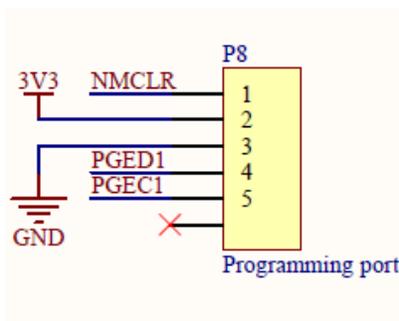


Figure 3.24: Programming Header Wiring Diagram

One of the important internal modules of this microcontroller is the 10-bit analog to digital converter. This exists as a single module inside the microcontroller but can be configured to read up to 16 of the 28 analog pins on the device; limited by the 16-register internal buffer. In this case, only three analog inputs will be utilised for the sensor signal, input voltage measurement and motor driver current measurement.

Another advantage of this microcontroller unit is the peripheral pin select ability, which allows the user to configure many of the general purpose I/O pins as input, output, digital, analog or special function. This greatly improves the design flexibility around this device, where pins can be easily reconfigured in software to suit different purposes. For this project, however, there are excess I/O pins compared to what is needed, so pin remapping has not been specifically required to save complexity; instead it acts more as a bonus feature. Table 3-2 displays a summary of the microcontroller pins used and describes their location and purpose.

Table 3-2: Table of Microcontroller Pins Used with Their Purpose, Port Location and Description

I/O PIN number	Port Location	Name	Pin configuration	Description
21	PORTB8	CAN-RX	Digital IN	CAN receive line, connects to CAN transceiver component
30	PORTB15	CAN-TX	Digital OUT	CAN transmit line, connects to CAN transceiver component
47	PORTC13	LED2	Digital OUT	General purpose notification LED (red)
48	PORTC14	LED	Digital OUT	General purpose notification LED (yellow)
49	PORTD1	Sensor filtered	Analog IN	Measure hall effect sensor value
50	PORTD2	Vsupply	Analog IN	Measure scaled PCB input voltage
51	PORTD3	INA output voltage	Analog IN	Measure variable voltage from current sense amplifier
52	PORTD4	Microstep 2	Digital OUT	Set microstepping mode bit 2
53	PORTD5	Microstep 1	Digital OUT	Set microstepping mode bit 1
54	PORTD6	Microstep 0	Digital OUT	Set microstepping mode bit 0

59	PORTF1	Step	Digital OUT/PWM	Generate square wave to drive stepper motor driver
61	PORTE1	Direction	Digital OUT	Set high or low for clockwise or anticlockwise motor movement
62	PORTE2	Decay	Digital OUT	Set decay mode on stepper motor driver
63	PORTE3	N_fault	Digital IN	Read stepper motor driver nFault pin
Other Pin Number	Name	Description		
7	nMCLR	Master clear pin provides device reset. Used by reset switch and programming port		
9	Vss	Ground reference for I/O pins		
10	Vdd	Positive supply reference for I/O pins		
15	PGEC1	Programming port clock pin		
16	PGED1	Programming port data pin		
19	AVdd	Positive supply reference for analog modules		
20	AVss	Ground reference for analog modules		
25	Vss	Ground reference for I/O pins		
26	Vdd	Positive supply reference for I/O pins		
33	USBID	USB ID pin for OTG devices		
35	VUSB3V3	USB internal transceiver 3.3V supply. Wired to general 3.3V rail here		
36	D-	USB data- line		
37	D+	USB data+ line		
38	Vdd	Positive supply reference for I/O pins		
39	OSC1	Primary external oscillator signal		
41	Vss	Ground reference for I/O pins		
56	Vcap	Internal voltage regulator capacitor		
57	Vdd	Positive supply reference for I/O pins		

3.5.5 Communication

The PCB design for this project includes multiple external communication systems, which provide capability for programming, debugging and communication to an external controller. This section details the design choices and implementation behind the CAN and USB communication capabilities.

3.5.5.1 USB

A USB micro A socket has been included on the PCB to interface with the microcontroller USB module and allow for serial communication to an external monitor. This has been designed to improve testing flexibility, where the user can easily monitor the processes and signals of the microcontroller from an external serial monitor. Additionally, once the device has been programmed via the programming port for the first time, it can be loaded with a bootloader that allows the USB module to program the device in all future programming

actions. This further improves design and testing flexibility as it reduces the need for the user to carry a programming device around.

The choice of a USB micro port specifically was made as they are very small and common. Compared to alternative USB plugs, USB micro ports take up the smallest PCB real estate, which helps for miniaturisation while still providing all the functionality. Additionally, as these sockets are very common on many electronic devices, it is highly likely that a user would have access to the corresponding USB micro cable to connect with. Figure 3.25 displays the wiring diagram for the USB micro socket on this PCB.

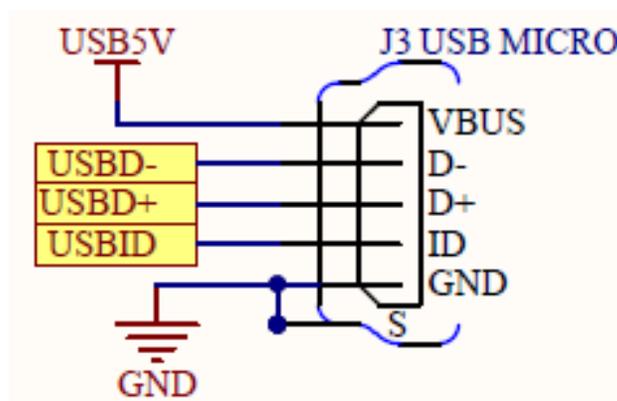


Figure 3.25: USB Micro Socket Wiring Diagram

3.5.5.2 Controller Area Network (CAN)

Another communication system included on this device is a Controller Area Network (CAN) system. As CAN buses are commonly used in the automotive industry, the inclusion of this system allows this device to easily interface with other vehicle hardware. This means that a future development of this project involving an overall vehicle controller can be simply applied to this device without needing to redesign it to include CAN.

To operate, a CAN network needs a bus module, a transceiver and line end termination. In this case, the bus module is already available inside the microcontroller, so no extra hardware is necessary. The transceiver section has been designed using a SN65HVD232-Q1 transceiver IC, which is the simplest unit of this device family. Other units in this range provide additional functionality such as different modes of operation and sleep modes, however this was not considered necessary for this project. The purpose of the transceiver is to convert the CAN transmit and receive lines on the microcontroller to appropriate high and low voltage lines for the differential signal transmission. In this case, the transmission operates over a -2V to 7V common

mode range. Also included within this circuit is the option to enable 120Ω termination resistance with a double dip switch. This allows the user to specify whether this device will be the final node on the CAN bus.

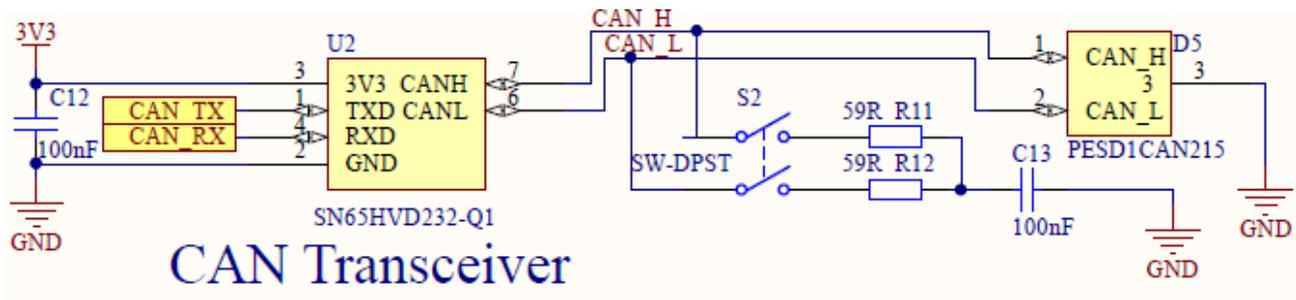


Figure 3.26: CAN Transceiver with End Termination and Protection Diodes

Transmitting the CAN signal off the device was performed using a standard DB9 socket with wiring diagram shown in Figure 3.27. This used only pins 2 and 7 to respectively provide the low and high signal lines. Additionally, a 12V supply line has been provided on the previously-unused pin 4, which allows any additional peripherals to access the 12V supply provided by this board. This supply contains a reverse polarity protection MOSFET as discussed previously and a 0Ω resistor for if the user decides to disable the feature altogether.

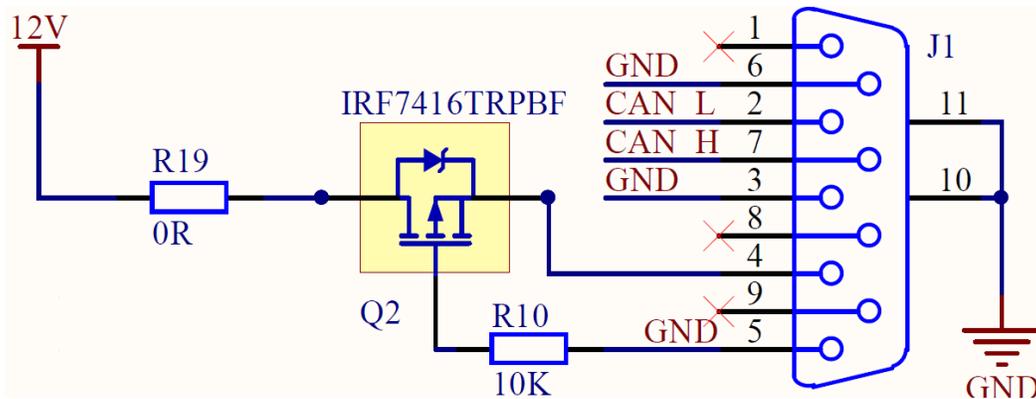


Figure 3.27: DB9 CAN Connector Wiring Diagram with Reverse Polarity MOSFET and 0Ω Resistor

3.5.6 LEDs

Three LEDs have been included on the PCB to provide visual feedback to the operator, as shown in Figure 3.28. D6 is hard wired to the 3.3V power line, which provides a permanent indicator that the microcontroller power source is active – either from the linear regulator or the programming port. This LED is green and provides a clear indicator to the user if the device is powered. LEDs D7 and D8 have been respectively wired to PORTC14 and PORTC13 of the microcontroller and can act as general-purpose notification lights from software. In this case, D7 is red and D8 is yellow so that all LEDs can be easily distinguished from each

other. All three LEDs have a 2.2V voltage drop and consume 5mA of current, which is regulated by the 220 Ω series resistors and is within the 15mA per pin maximum current sourcing limit of the microcontroller.

An example use of these would be to configure the red LED as an error light, which activates in any error condition such as motor driver over temperature, supply voltage too low or sensor out of expected range. The yellow light could be configured to activate during occasional processes such as when the motor is running or when a CAN or USB data transfer is in process.

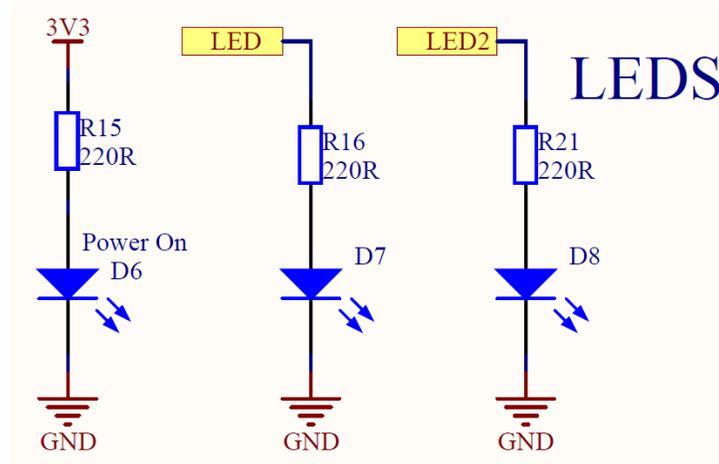


Figure 3.28: Power and Notification LEDs Wiring Diagram

3.5.7 Current Measurement

As this device involves handling comparatively high current through the motor driver, a current sensing amplifier circuit has been included to measure the current and transmit a corresponding analog voltage to the microprocessor, as shown in Figure 3.29. This allows for easy measurement through software, so the user can easily monitor the motor current. This operates using an inline 5m Ω current sense resistor directly before the motor driver and an INA180 sensing amplifier to measure the voltage drop across it. The INA180 has been selected with the largest possible gain of 200V/V to allow for greatest output voltage precision, where a current range of 0-3A corresponds to an output voltage range of 0-3V (see Equations 14-16). Also included within this circuit is a test point to allow the user to probe the current sense amplifier output without using the microcontroller if desired.

$$V = IR \text{ \{Ohm's Law\}} \quad (16)$$

$$\therefore \Delta V = 3A \times 0.005\Omega = 0.015V \quad (17)$$

$$V_{\text{out}} = G \times \Delta V = 200 \times 0.015V = 3V \quad (18)$$

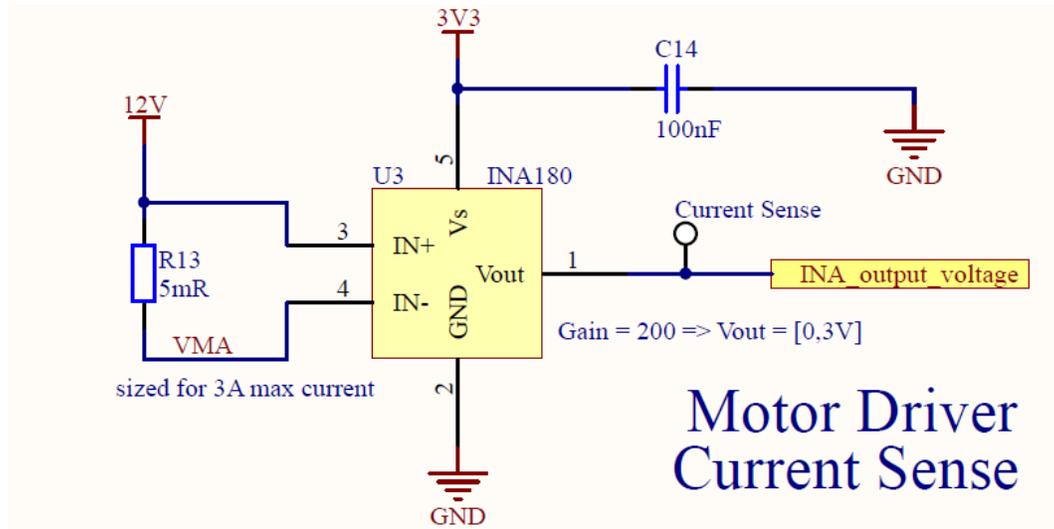


Figure 3.29: Current Sense Amplifier Circuit with Gain 200

3.5.8 Protection

Multiple safety features were included in the design of this PCB to provide maximum protection against short circuits, electrical errors and user errors. This included reverse polarity protection on high voltage power lines, a self-resetting fuse on the main supply line, sensor signal line voltage regulation, voltage level protection diodes on the CAN bus lines and 1-way keyed headers to peripherals.

Reverse polarity protection was added using IRF7416TRPBF PNP power MOSFETs. This was included on both on the main 12V supply line (Figure 3.17) and the 12V out line in the DB9 connector (Figure 3.27) to protect against 12V being wired backwards and damaging sensitive components. This operates by pulling the MOSFET gate to ground using a 10K resistor, which causes the MOSFET to allow current to flow. If the voltage is wired backwards, the ground plane will encounter a higher potential than the 12V supply line, which will cause the MOSFET to become open circuit and prevent current from entering the circuit. The IRF7416TRPBF MOSFET has a bias voltage of -1V, can source current up to -3.1A continuously and has a very low on-resistance of 0.02Ω . This makes it perfectly suitable for this purpose as it can safely block all the input current in open-circuit mode and allow all the current with minimal voltage drop during closed-circuit mode.

After the input voltage passes the reverse polarity protection MOSFET, the next safety feature is a 0ZCG0260BF2B self-resetting PTC fuse (represented by F1 in Figure 3.18). This protects the remaining circuitry from excessive current draw, where this fuse will trip between 2.6 and 5A. The range presented here was designed to accommodate the 2.5A motor driver maximum, so any current larger than this will

automatically be fused out. Additionally, this fuse is rated at 16V, which is acceptable for the expected 12V supply of this system. The self-resetting nature of the fuse also means the user does not need to manually reset or replace the fuse if it trips, which improves user operability.

The hall effect sensor circuit used in this project contains a voltage regulation diode package which protects the microcontroller IO pins from an unexpected voltage event (shown as D4 in Figure 3.22). This setup uses a SDM40E20LS diode package with 0.3V forward voltage on both diodes. This means that if the sensor signal line voltage exceeds the -0.3V to 3.6V range, the appropriate diode will become forward biased and short the signal to the respective nearby logic level; hence preventing electrical stress on the corresponding microcontroller pin. Due to the circuit design for flexibility, it is possible for the user to bypass this protection diode by setting the sensor jumper appropriately. It is recommended that the user exercise caution when performing this action, as it increases the risk of damage to the microcontroller in unexpected voltage events on the sensor signal line.

Like the sensor protection diodes, the CAN network transceiver circuit contains voltage level regulation using the PESD1CAN215 diode package (shown as D5 in Figure 3.26). This single SOT23 package is specifically designed to protect two CAN bus lines from electrostatic discharge and surge pulses. As CAN operates using a differential signal, this protection diode protects the lines from exceeding a maximum voltage difference of 24V. In such an event, the diodes will become forward biased and the extra voltage will be dissipated to the ground plane.

Finally, the connection points to the hall effect sensor and stepper motor have been designed to use 1-way keyed header plugs. This provides protection against reverse polarity connection, which could destroy the sensor or disable the motor. Additionally, this securely locks both connectors in place, which prevents the hardware from becoming unexpectedly disconnected and causing errors in the motor feedback loop.

3.5.9 Placement and Routing

The process of routing the PCB was performed based on a set of basic design principles which intended to improve the device efficiency and simplify testing and rework. Many of the integrated circuit components specified to use decoupling capacitors between the power supply rails. These were arranged to minimise the distance between the capacitors and the respective components. This was especially important for the microprocessor and oscillator, which both required stable and very low-noise power lines. Figure 3.30

displays a section of the PCB design demonstrating the microcontroller, oscillator and corresponding decoupling capacitor layout with minimised distance to the decoupling capacitors.

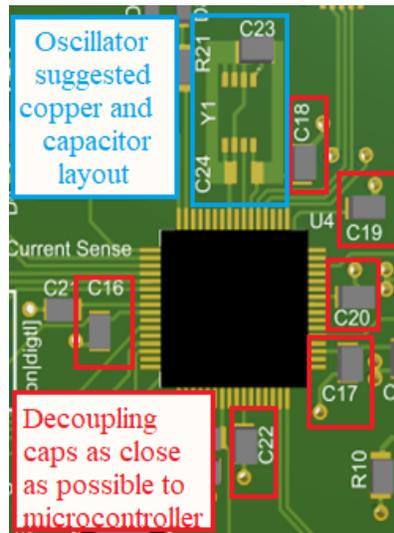


Figure 3.30: Example Arrangement of PCB Decoupling Capacitors at Minimised Distance to Microcontroller and Oscillator Layout

Another technique employed to reduce noise on sensitive lines was to reduce the length of low voltage traces and traces on the underside of the board. This was especially important for the CAN communication lines, which have been placed directly adjacent to the DB9 connector. Keeping such traces to a minimum length reduced the probability for noise and signal reflections to appear within the lines, which could be detrimental to the signal integrity. Figure 3.31 displays the PCB layout of the CAN components. Additionally, most of the passive components that include a ground connection were wired directly to the ground plane on the underside of the board using a short track and an individual via. The wiring configuration was also arranged such that the number and length of tracks within the ground plane were minimised. Overall, only five tracks needed to use the underside of the board, with the longest measuring 5mm.

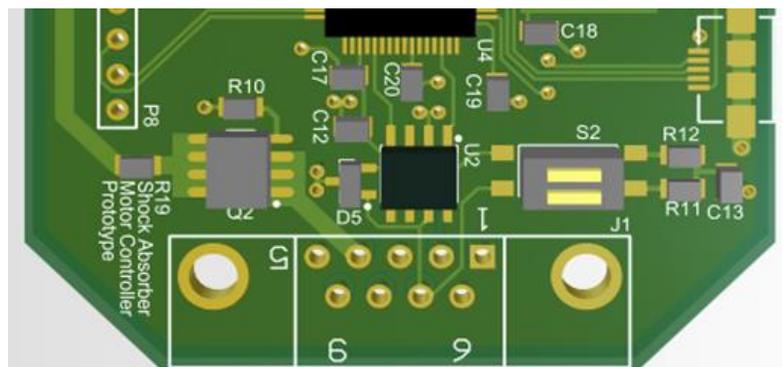


Figure 3.31: CAN Component Placement Adjacent to DB9 Connector for Minimised Noise Risk

Among the design features towards increased flexibility was the decision to use jumped header pins to control hardware switching instead of dip switches. This was implemented for the microstepping selection, microcontroller power source and sensor filter bypass, totalling five separate jumpers. This could have been implemented instead by using dip switches, however header pins were chosen instead as they are cheaper and use less space. A future iteration of this board may use dip switches or software to perform these switching actions.

The microstepping selection pins have also been arranged with extra spacing between the adjacent rows, which prevents the user from installing the jumper pins in the wrong place and creating a short circuit. Figure 3.33 demonstrates this arrangement, where the distance represented by the red indicator is intentionally larger than that of the blue indicator.

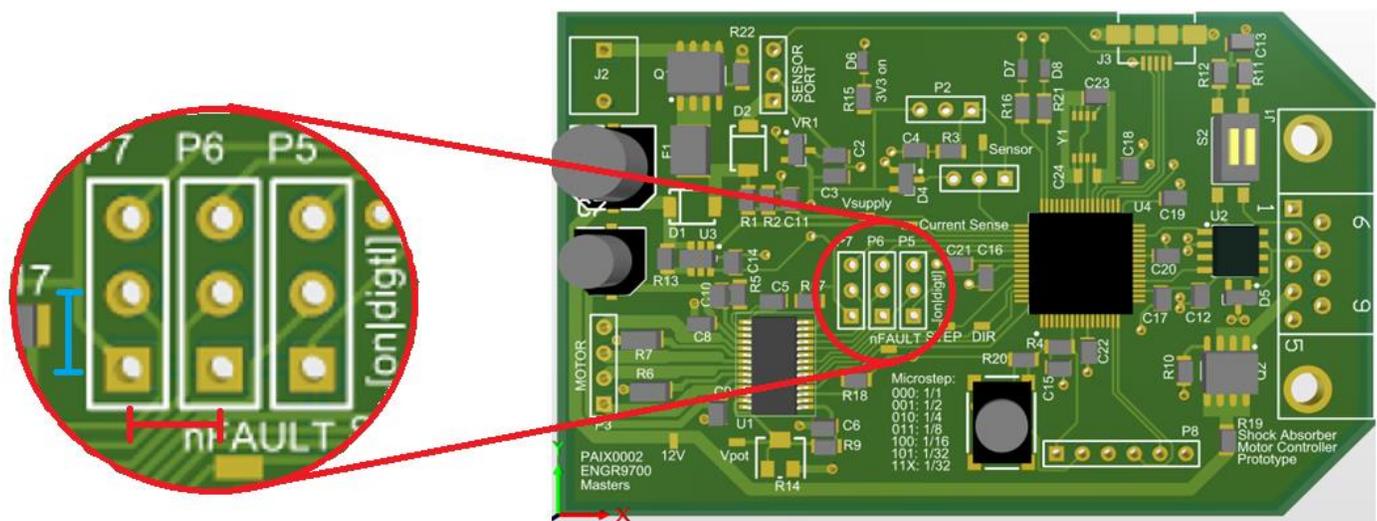


Figure 3.33: PCB Layout of Microstepping Mode Selection Headers with Extra Space Between Rows

3.5.10 Cost

Estimating the overall cost of this device can be broken down into different stages based on the different stages of design. The 3D printed components of the mechanical test rig were created using the university resources available for research purposes, however an estimated cost of printing can be calculated. Considering the \$46.90 cost of a 900g roll of 3D printing plastic (Imaginables, 2019), and 100g estimate of the 3D printed components, the total printing cost becomes \$5.20. Additionally, the machined aluminium valve components have been estimated to cost up to \$15 of material and \$50 for one hour of machinist labour and supplied by the mechanical workshop at Supashock. Adding to this, the cost of the shaft coupler is \$7.99 from eBay and the ring magnets are \$4.25 each from frenergy.com.au. Hence the total cost of the mechanical components would be \$90.94.

The cost of the electronic components is broken down into individual detail within the bill of materials in Appendix 6 and equates to \$66.06 plus the \$24.86 stepper motor and \$2.59 hall effect sensor. Ordering the custom PCBs from PCBway.com incurs a cost of US\$5 (AU\$7.22) for five PCBs (minimum order), US\$15 (AU\$21.64) for a stencil and US\$50 (AU\$72.16) for delivery via DHL. This forms the average cost estimate of US\$14 (AU\$20.21) per blank PCB. The assembly labour for the PCBs would be performed by the author and primary researcher, which therefore would cost \$0.

Overall the cost estimate of a single test rig and PCB would be \$204.66. In this case, the minimum order requirement for printed PCBs is five boards so one single PCB cannot be produced at the above-mentioned cost. If five PCBs and one test rig were constructed, the price would be accurately \$520.85, which would leave four PCBs constructed but idle. If five test rigs were constructed to accommodate the five PCBs, the final cost would increase to \$994.41, which is beyond the \$600 allowed budget of this project. Hence, components for five PCBs will be ordered, but only one test rig will be constructed, at a final cost estimate of \$520.85. This provides multiple spare PCBs to allow for electrical bugs or Supashock demonstration models but remains cost efficient by saving on mechanical costs.

4. RESULTS

This section details the experiments performed during this project and forms discussions about the meaningful outcomes of each test. These were performed in three stages throughout the research process to validate designs and measure performances. Early experiments aimed to validate the chosen motor and sensor components for their use in the project, as well as the early mechanical designs. Next, the full device assembly was prototyped, and experiments were performed to validate the assembly process. Finally, the assembly was tested as a complete mechanical and electronic structure, and the performance was analysed.

4.1 Preliminary Testing

A set of experiments were performed in the project early stages to validate the initial designs and confirm their planned use in the overall assembly. This involved initial test 3D-printing of the plastic mounting components and familiarisation with the sensor and motor components. Early electronic experiments were performed using an Arduino Uno and CNC stepper motor driver shield to interface with the motor and sensor, along with the Arduino IDE for programming.

4.1.1 Mechanical

Early designs of the 3D printed mounting shell involved fewer smooth curves and a blockier shape, which was simpler to design but had notable disadvantages once it was fabricated. Figure 4.1 displays the inventor model and first 3D print of the first mounting shell, with highlights towards the main issues encountered.

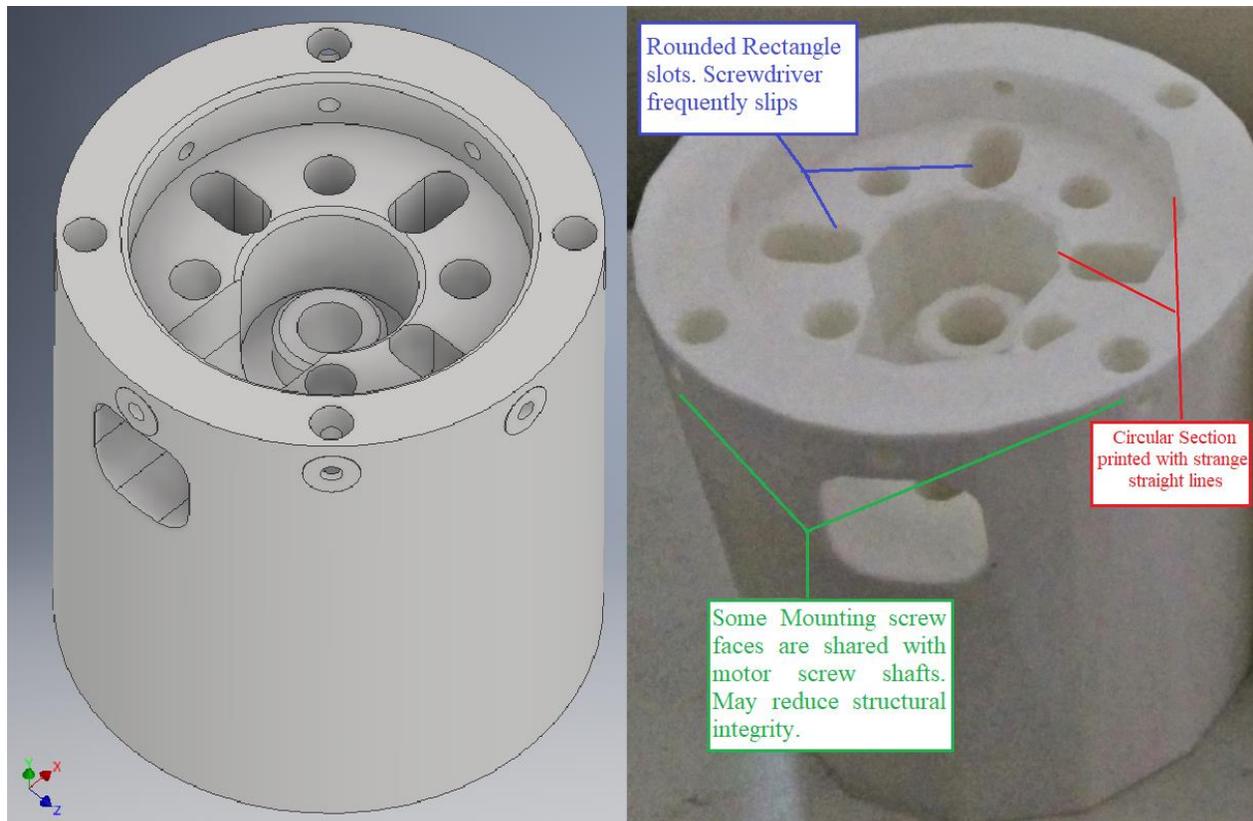


Figure 4.1: Design and 3D Print of First Mechanical Mounting Shell

The red indicator box in Figure 4.1 displays that the printer created the component with straight line sections on the expected circular curves. This was considered a major defect as the shape prevented interfacing with the circular valve component. Investigating this further found that this was caused by incorrect printer settings, where the slicing resolution was set to low but needed to be set to highest.

Additionally, half of the radial mounting screws for the valve block had intersecting shafts with the screw shafts for the NEMA23 motor mounts. This caused the mounting points for four of the eight radial screws to become very thin, which increased their risk of snapping when tightening the screws. Structural integrity was improved by adjusting the design to rotating the NEMA23 screw shafts such that they no longer intersected the valve mount shafts. Additionally, the thickness of the mounting points was reduced from 10mm to 5mm to increase the length of the screws that penetrates the valve block.

Another design issue realised by this 3D print was that there was no secure technique to mount the sensor bracket, which would greatly increase the risk of it shifting randomly. This was considered a major design flaw as a shifting sensor bracket would change the magnetic field strength detected by the hall effect sensor, and thus would cause inconsistencies in the data readout. To resolve this, the next iteration of the mechanical

design included a slot in the shell which would allow the sensor bracket to share a valve mount screw and prevent it from slipping during operation.

During testing of this shell, it was also discovered that the mounting points for the motor screws had been designed too thick, which reduced the available screw length of the 15mm screw to just 2mm to mount to the motor (shown in Figure 4.2). This was considered a mechanical hazard as the short mounting distance increased the risk of the heavy motor snapping off. Additionally, while testing the motor-mount screws, it was discovered that the rounded-rectangle shaped shaft shared by the NEMA11 and NEMA17 mounting holes (noted by blue in Figure 4.1) caused the screwdriver to frequently slip and miss the intended screw head. This was considered a mild inconvenience and would be addressed in the next design iteration to improve the simplicity in assembling the device.

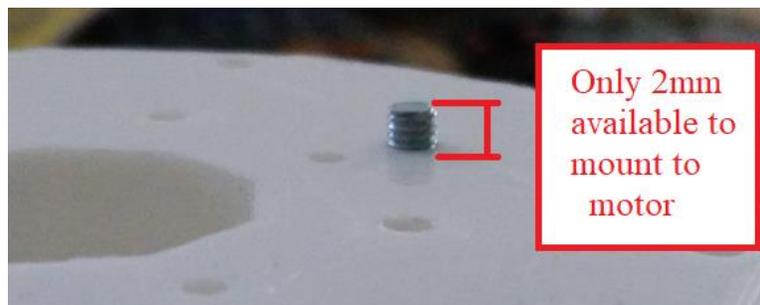


Figure 4.2: Remaining Screw Length Too short

After testing the first shell module and documenting the design flaws, an updated design was created and printed which addressed the issues. Figure 4.3 displays the model and print of this second-generation shell, with indicators towards the resolved problems. To overcome the shaft-sharing issue, the NEMA23 mounting angle was rotated by 22.5° , which positioned both the motor shaft and valve mounting on solid plastic regions. Increasing the resolution of the print was found to solve the non-circular circles, where now they could successfully mate with the circular valve parts. Additionally, the valve mounting screw locations were indented to increase the length of the screw penetrating the valve and a slot was added to the top of the sensor bracket, so it could share the one valve screw there. Finally, the thickness of the motor mounting slots was decreased such that the screws would penetrate 5mm into the motor sockets, rather than 2mm, and the screwdriver access shafts were changed from rounded-rectangle to figure-8 shape to reduce screwdriver slipping.

The updates to the designs of the mount shell were found to create a much stronger, safer and easier to assemble component.

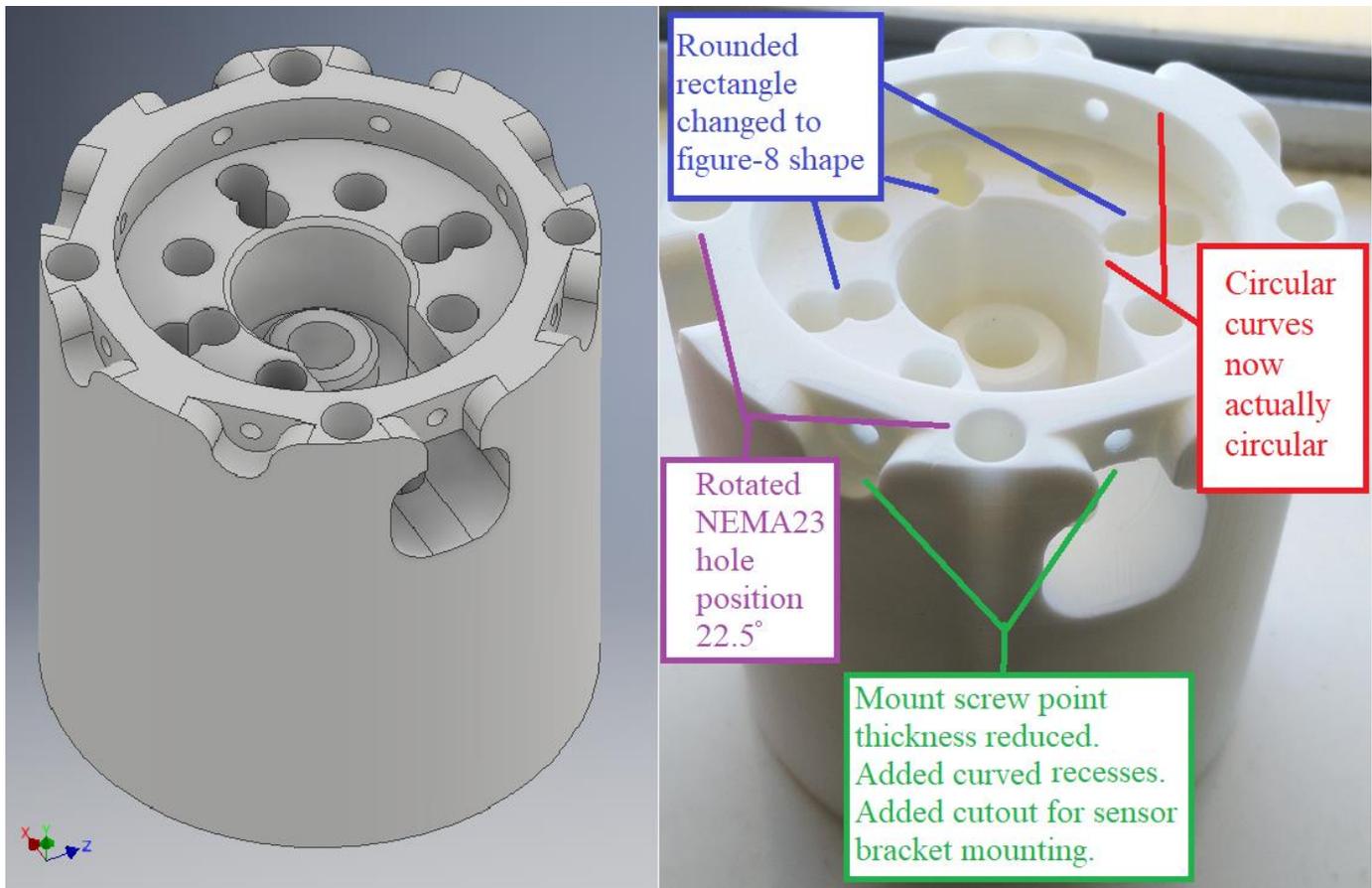


Figure 4.3: Second Generation Mounting Shell Model and Print with Improvements

4.1.2 Electronic

To ensure that the electronic components would perform as expected for the final assembly, they were tested individually in simple experiments with generic test equipment. The motor was tested using an Arduino Uno with a stepper motor CNC shield and the sensor was tested using the Arduino analog I/O pins. The following sections explain the corresponding motor and sensor preliminary tests and their outcomes.

4.1.2.1 Sensor

Testing the SS49E hall effect sensor was performed using an Arduino Uno and a magnet. This simply provided 3.3V power and ground to the sensor and measured the analog voltage on analog pin A0. The voltage was measured using the internal 5V 10-bit ADC inside the Arduino microcontroller and visualised using the Arduino IDE serial monitor using code shown in Figure 4.4.

```

const int sensorPin = A0;           //analog sensor pin

void setup() {
  int sensorValue;                 // value read from the sensor
  Serial.begin(9600);              //initialise serial monitor
}

void loop() {
  int sensorValue = analogRead(sensorPin); //read sensor value
  Serial.println(sensorValue);          //display to serial window
  delay(50);                            //wait 100m
}

```

Figure 4.4: Arduino C Code Used to Read and Display Analog Voltage on Arduino Pin A0 Using Arduino IDE Serial Monitor

This experiment was performed to provide an understanding of the expected range of voltage and digital values that would be returned by the hall effect sensor. At this stage of the experiment, the planned ring magnets had not arrived, so the magnetic field was applied using a repurposed computer hard drive magnet. The sensor was measured for values of no flux, maximum positive flux strength and maximum negative flux strength. These were measured by placing the magnet as close as possible to the sensor and observing the response for both sides of the magnet. Table 4-1 displays the output response of the sensor with information in both raw values and volts. Conversion of raw data to volts was performed using Equation (19). Additionally, Figure 4.5 demonstrates an Arduino serial plotter waveform chart of the ADC output under minimum, maximum and neutral magnetic field strength conditions.

$$V_0 = \frac{\text{raw}}{2^{10} - 1} \times 5V \quad (19)$$

Table 4-1: Voltage Response of Hall Effect Sensor with Different Magnetic Field Conditions

Magnet Position	Raw ADC Value	Equivalent Voltage (V)	Difference from Neutral (V)
Maximum North	514	2.51	+0.83
None (neutral)	344	1.68	0
Maximum South	169	0.83	-0.85

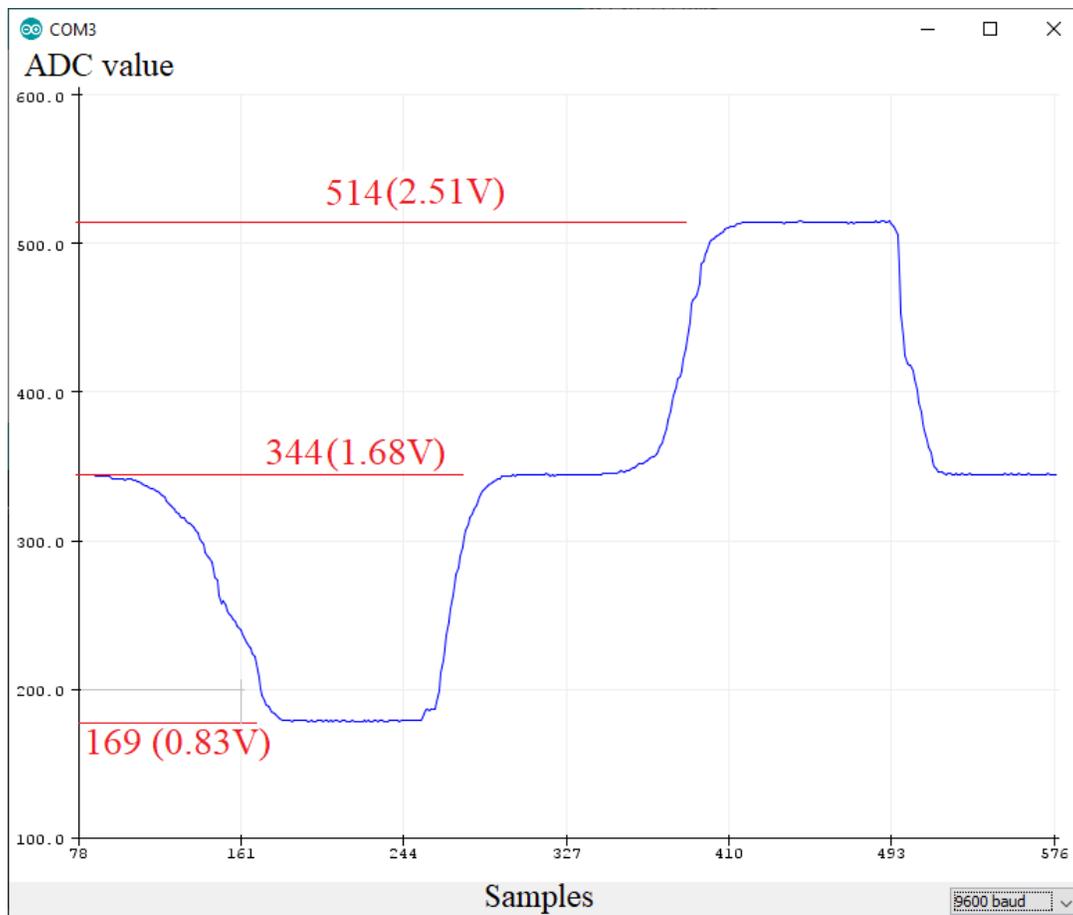


Figure 4.5: Serial Plotter Graph of Hall Effect Sensor Response at Minimum, Maximum and Neutral Magnetic Field Strength

This demonstrated that the SS49E hall effect sensor could perform as intended for this project, where the output voltage would vary linearly from 0.55V to 1.66V between maximum south and maximum north magnetic conditions respectively. The output waveform was also observed to encounter very low noise, where the ADC raw output would oscillate by only ± 1 (3.2mV). It is worth noting that the magnetic fields will not be swinging from maximum closed to maximum open in the final project assembly, so the output voltage range will likely be smaller at both extremes. This data was measured using the 10-bit ADC present on the Arduino, which has the same precision as that on the PIC microcontroller planned to be used on the final PCB; hence a similar response should be expected in the final assembly.

4.1.2.2 Stepper Motor

Like the sensor, the 42BYGHM809 stepper motor was acquired in the early stages of the project for early testing to ensure it would perform appropriately for the desired task. This was tested using an Arduino CNC stepper motor driver shield with a DRV8825 motor driver card. Conveniently, this driver IC is the same as

that chosen for the custom PCB, which means that the preliminary motor experiments can validate both the motor and driver choice simultaneously.

Testing was performed using the code displayed in Figure 4.6 to drive the motor at different stepping rates and observe the performance. The circuit was connected using a lab bench power supply with 12V output and a 1.8A current limit, and the current draw was observed. It was observed that the motor could operate smoothly within 500-1600 steps per second (1.25 – 4 rotations per second), where higher speeds caused slip and lower speeds caused jitter. Within this ideal speed range, the current draw for an unloaded motor shaft was 250mA. It is important to consider that this will likely be higher in the final assembly when the shaft is mechanically loaded.

```

#define direction_pin 5
#define step_pin 2
const int steps_per_rot = 400; //constant for this motor
void step(boolean dir, byte delayus, int steps);

void setup() {
  pinMode(8, OUTPUT); //set CNC shield enable pin output
  pinMode(step_pin, OUTPUT); //set step pin output
  pinMode(5, OUTPUT); //set direction pin output
  digitalWrite(8,LOW); //enable cnc shield module
}
void loop() {
  bool dir = true; //which direction to rotate
  int rotations = 5; //number of rotations
  int steps = rotations * steps_per_rot; //number of steps to perform
  int delays = 1;
  int delayus = 1200; //delay time sets motor speed

  //-----rotate motor for specified speed, direction and number of rotations-----
  step(dir,delayus,steps);
  delay(1);
}

//square wave pattern from delay
void step(boolean dir, int wait, int steps)
{
  digitalWrite(direction_pin, dir); //set direction
  for (int i = 0; i < steps; i++) {
    digitalWrite(step_pin, HIGH);
    delayMicroseconds(wait);
    digitalWrite(step_pin, LOW);
    delayMicroseconds(wait); }
}

```

Figure 4.6: Basic Motor Test Code with Variable Speed and Rotation Count Setting

Another noteworthy observation made during this experiment was that there was a brief current surge of 1.8A at one instance of the motor starting, which burst the fuse on the CNC driver shield. This further highlighted the need to include safety mechanisms in the custom PCB, as this could have caused more significant damage to more sensitive components if the fuse had not been in place. The reason for the current surge is unknown, although it was suspected to be caused by enabling the power supply during the drive loop of the microcontroller.

Table 4-2 displays a summary of the rated and observed characteristics of the stepper motor intended for use with this project. The measured speed and current characteristics were determined to be acceptable for use with the final assembly. Although the data measured here describes the characteristics of an unloaded shaft, they are far within the maximum ratings of the motor so would likely still be acceptable with a mechanical load. For example, the current draw at maximum smooth motor speed was measured to be 250mA, which is far from the maximum limit of 1.7A.

Overall this preliminary testing validated the plans for the use of a 42BYGHM809 stepper motor and DRV8825 driver.

Table 4-2: Table of Rated and Observed Characteristics of Project Stepper Motor

Stepper Motor (42BYGHM809) Tested with 12V supply and unloaded shaft	
Rated Characteristics	
Max Current (A)	1.7
Steps/rev	400
Torque (Nm)	0.48
Size	NEMA17
Winding Type	Bipolar
Observed Characteristics	
Max Speed Without Slip (steps/s)	1664

Max Speed Without Slip (RPS)	4.16
Smooth Stable Speed Range (steps/s):	500-1600
Smooth Stable Speed Range (RPS)	1.25 - 4
Current (A) at Hold	0.52
Current (A) at Max Speed	0.25
Current (A) When Fuse Popped	1.8

4.2 Assembly Prototyping

After completing the design of the test rig assembly, the device components were fabricated and assembled to validate their functioning in the real world. This was performed using the same 3D printing and Arduino processes used in the preliminary validation tests, with a focus towards the capabilities of the assembly as a complete unit. This section describes the process of fabricating and assembling the test rig prototype, as well as the related experiments performed to validate the assembly, and their significance to the project overall.

4.2.1 Mechanical

During the assembly stage of the motor shaft mechanism, it was reported by the Supashock mechanical engineering team that the custom hex socket introduced in Figure 3.7 was too complex and could not be manufactured. This was due to the hexagonal socket end, which required specialised tools and excessive labour. This meant that an alternative solution or redesign needed to be proposed which could provide the same mechanical motion. The solution was found with a repurposed hand tool piece of a multi-screw driver set owned by the author, shown in Figure 4.7. Although this provided a suitable socket, the 4mm shaft was too long and needed to be reduced. The red marking in Figure 4.7 shows the planned cut point, which left 1cm of material attached to the socket; enough to mount flush within the flexible shaft coupler.

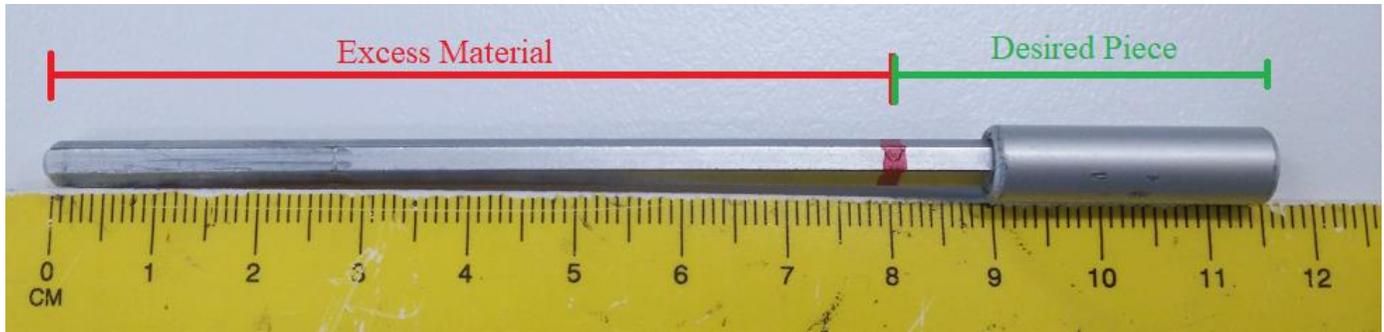


Figure 4.7: Screwdriver Hand Piece with 4mm Hex Shaft and Socket End

When it was initially designed, the custom hex socket was intended to use a circular 4mm tail to mate with the circular hex shaft, however the repurposed hand tool contained a 4mm hexagonal shaft instead. This meant that the largest diameter was 4.6mm, which was too large to mount to the 4mm circular shaft coupler as desired. To overcome this, the 4mm hole of the shaft coupler was carefully widened using a 4.6mm drill bit and the custom hex socket was tapped into place using a rubber mallet.

Due to this modified hole and mallet technique, the hex socket became permanently mounted to the shaft coupler with a very tight fit. As a result, the grub screws are not necessary for tightness, although they were secured anyway to prevent the shaft from rattling during operation. This alternative solution proved successful as it could still perform the original intended task and was very secure. Figure 4.8 displays the final flexible shaft and hex socket assembly and Figure 4.9 displays the assembly mounted on the stepper motor.



Figure 4.8: Complete Flexible Shaft Coupler and Hex Socket Assembly



Figure 4.9: Stepper Motor and Shaft Assembly

Finally, the motor shaft assembly was mounted to the mount bracket piece using four 3mm screws and is displayed in Figure 4.10. Inspecting this assembly found that the fit was suitable, where the motor connection was very sturdy and had no observable wobble. Additionally, the length of the shaft mechanism aligned exactly with the top of the mount shell as intended. It is worth noting that a slight shaft concentricity misalignment was observed, however this was not considered an issue as this would become corrected by the flexibility of the shaft coupler once the valve was connected.

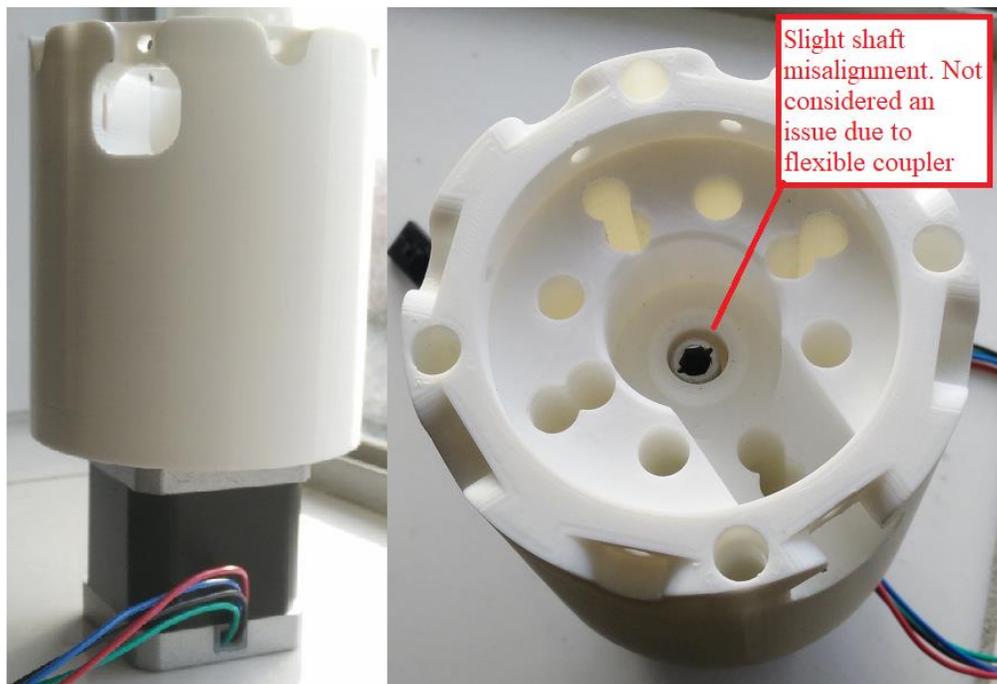


Figure 4.10: Motor and Shaft Mechanism Successfully Screwed to Mounting Shell

After the engineering drawings for the custom aluminium valve were submitted to the mechanical workshop at Supashock, the parts were also 3D printed to allow for basic testing while waiting for the real components to arrive. Figure 4.11 shows the front and top views of the 3D printed valve and Figure 4.12 shows the needle with the ring magnets connected. This demonstrates that the needle fitted successfully into the appropriate socket and could be observed through the oil fitting holes to successfully open and close the valve. Although the mechanical function appeared to operate correctly, this was not intended to be tested with actual oil pressure as the 3D printed material was not designed to handle a hydraulic system.

It is important to note that Autodesk Inventor does not natively support creating threads in STL files, which means that this test printed without any screw threads on either component. This aspect would need to be further tested with the aluminium valve, which would be machined to include threads.

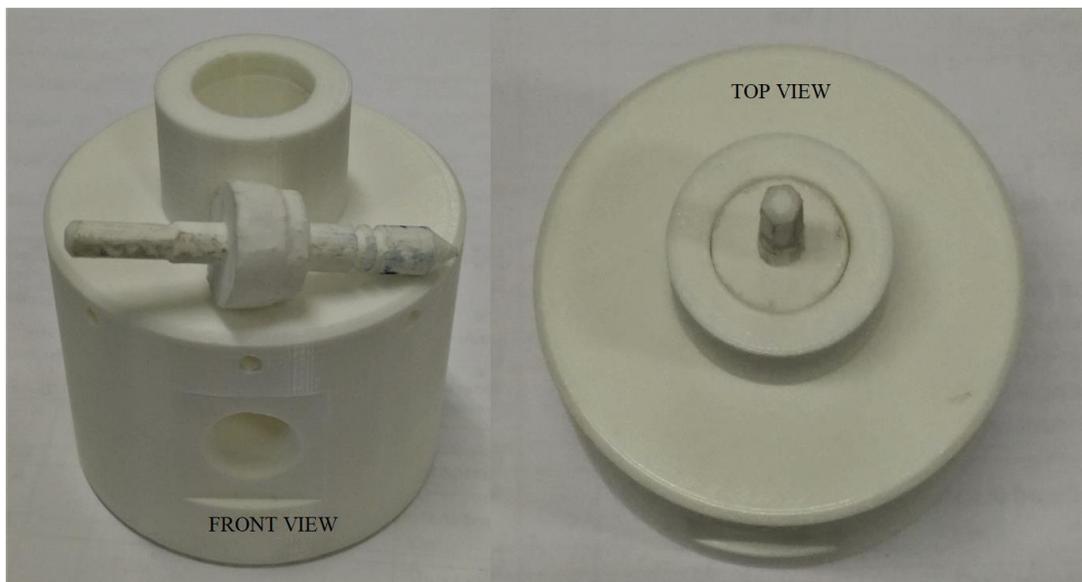


Figure 4.11: 3D Printed Version of Custom Valve from Front and Top Views with Needle Separate and Inserted

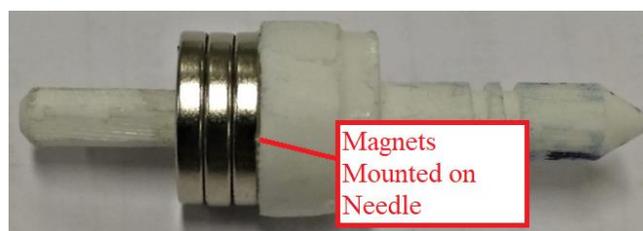


Figure 4.12: 3D Printed Valve Needle with Ring Magnets Attached

Next, the 3D printed valve was mounted to the remaining assembly as shown in Figure 4.13. This was found to produce a sturdy fit with no wobble or misalignment. Additionally, the needle was found to successfully mate with the shaft hex socket to provide rotational coupling and linear sliding as intended. When attempting to screw the eight 3mm mounting screws in place, however, it was found that the 3mm holes on the 3D printed valve had shrunk during the process, so could not fit the screws. This was not considered a significant issue though, as the aluminium valve would be machined more precisely so would not encounter this error.



Figure 4.13: 3D Printed Valve and Motor Mounted to Assembly

Like the other structural components, the sensor mounting bracket was 3D printed and the sensor was installed. This was performed by soldering extension leads onto the sensor (shown in Figure 4.14), with one small heat shrink tube piece on only the centre leg and pulling them through the cable access hole. Additionally, a 0.1uF 0805 surface mount capacitor was soldered between the power lines of the sensor to reduce the effects of any noise developed over the long wires. Due to the small size of the cable hole, the sensor was able to be mounted using only the tightness of the single heat shrink, which provided a very sturdy fit (shown in Figure 4.15).

After the cables were routed through the bracket, a header plug was attached to the other end to allow connection with a PCB and to prevent the sensor from slipping out. Finally, this was mounted into the test rig assembly in the assigned slot and attached with the single mount screw. This completed the assembly of the test rig prototype with the temporary 3D printed valve.

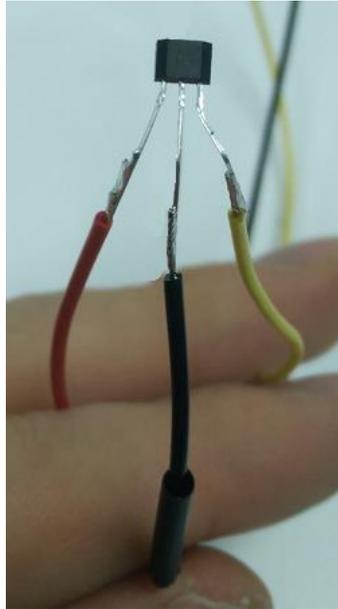


Figure 4.14: Hall Effect Sensor with Soldered Extension Leads

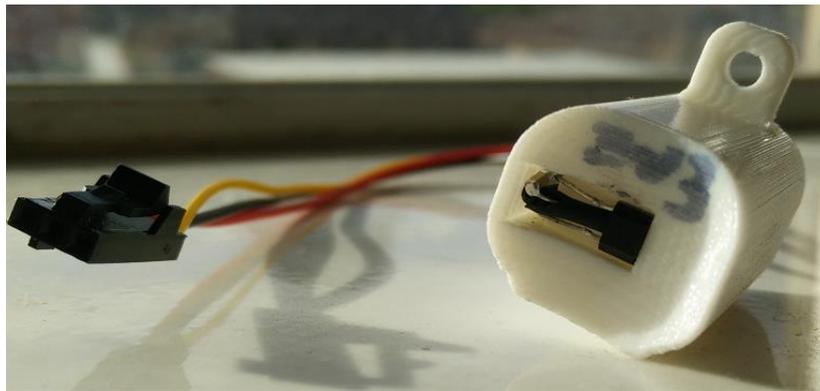


Figure 4.15: Hall Effect Sensor Installed into Mount Bracket with Header

An additional test performed for research purposes only was to design a custom long needle, which would be 3d printed and used to observe the needle position from outside the valve. This was designed as identical to the main needle but with an additional 35mm extrusion from the front, which would poke out of the valve through the oil-inlet hole and allow the user to identify the position of the needle for testing purposes. This would only be implemented on the 3D printed test valve, as it is not a necessary part of the final valve assembly. Furthermore, this needle was marked with dots spaced 1mm apart to allow for measurement of the

position. This was used to analyse the response of the hall effect sensor over valve displacement, which is discussed further in Section 4.2.2.

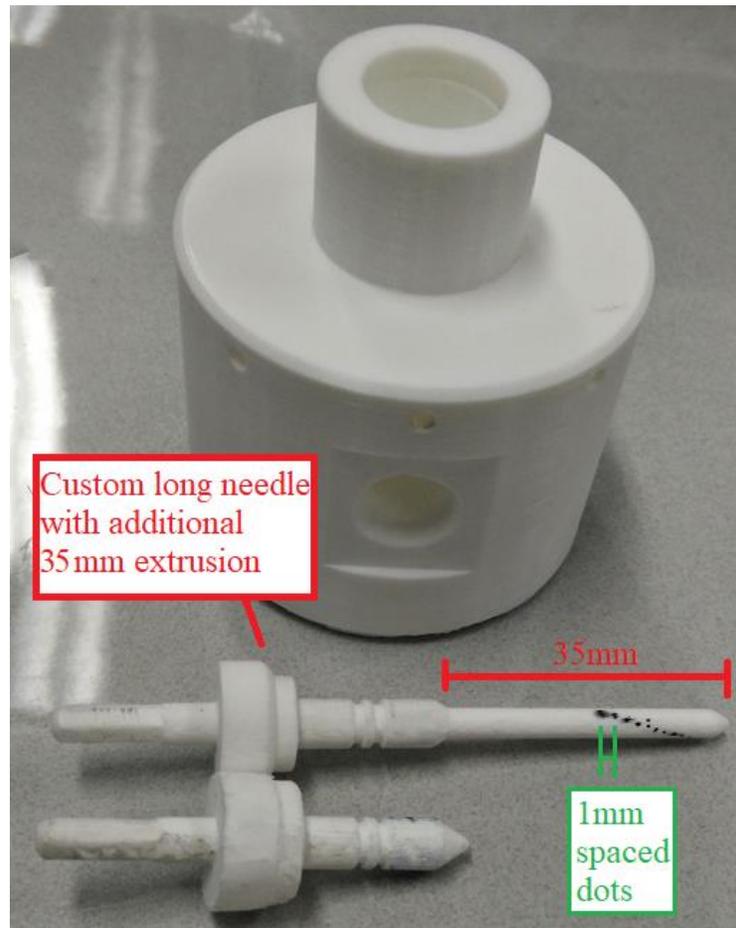


Figure 4.16: 3D Printed Valve with Additional Custom Long Needle

4.2.2 Electronic

Once the prototype test rig was fully assembled, the hall effect sensor was tested to analyse its response during different needle positions. The purpose of this experiment was to form a model of the sensor voltage over the linear range of movement of the needle. Understanding this would allow the user to accurately know the position of the needle through the sensor, and thus to model the valve response at different motor positions.

This experiment was performed by assembling the test rig including the ring magnets and hall effect sensor with a lab bench power supply and oscilloscope, as shown in Figure 4.17. The Arduino Uno was connected to the hall effect sensor with the output pin on analog pin A0 and the code introduced in Figure 4.4 was used

again here to measure the sensor output. Beginning with the valve position at fully opened, the hall effect sensor output was measured, and the needle was shifted by 1mm increments until the valve was fully closed.

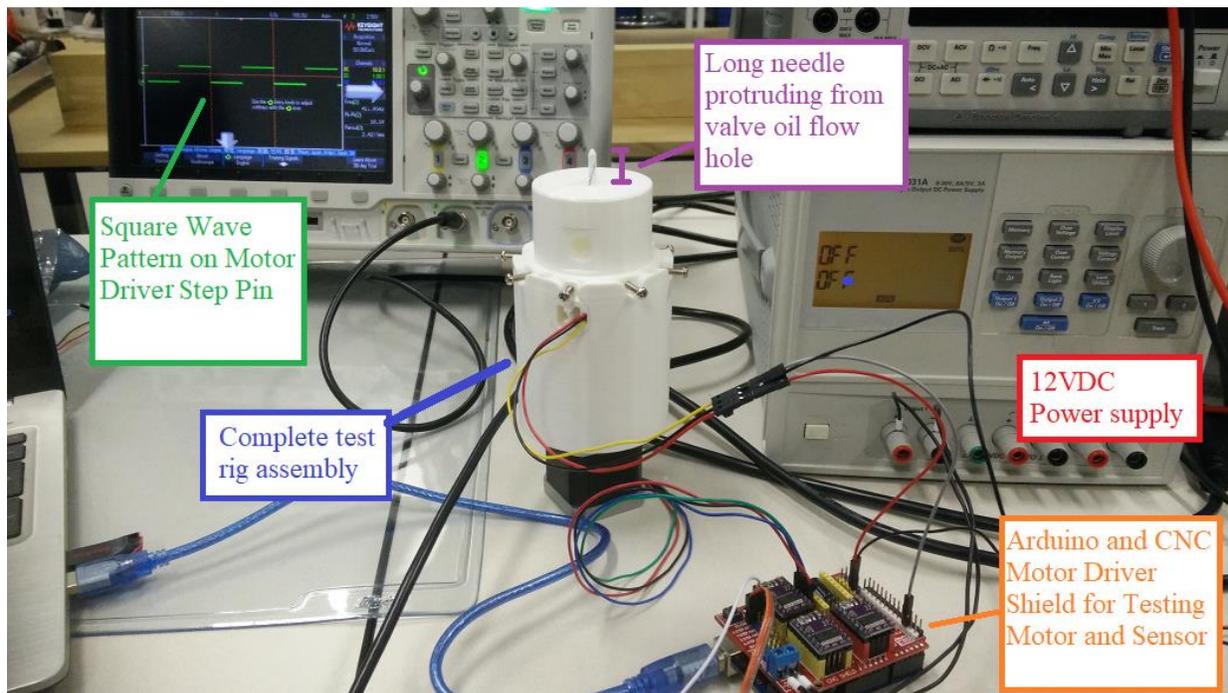


Figure 4.17: Test Setup of Complete Test Rig Assembly

Figure 4.18 displays the graph of sensor output VS distance from fully opened valve position, measured by the hall effect sensor. This demonstrates that the sensor produces a linear output with needle movement and a voltage range of 0.27V between fully opened and fully closed position. Interestingly, the hypothesis that the voltage range would decrease compared to the maximum and minimum sensor value test (Figure 4.5) was validated by the data measured here. This is due to the orientation and distance of the ring magnets, which cannot exert as much magnetic flux onto the hall effect sensor compared to the full contact position of the repurposed hard drive magnet.

Furthermore, Figure 4.19 and Figure 4.20 display the results of repeating this experiment with the motor rotating at 1.5 and 3 rotations per second respectively. Observing these three graphs demonstrates that the sensor produces the same output response independent of the motor speed. This indicates that the sensor is not affected by vibration, power fluctuations or magnetic fields of the motor, which means that the system has good reliability.

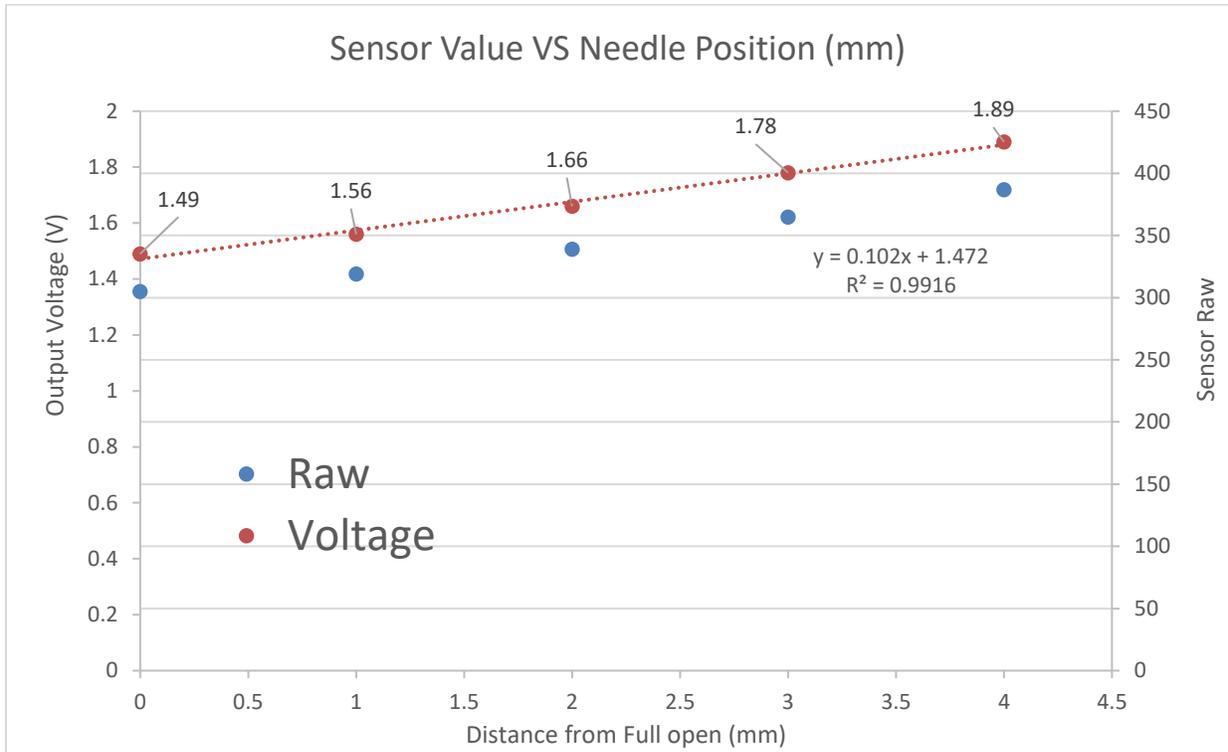


Figure 4.18: Sensor Output (raw, V) VS Distance (mm) from Fully Opened Valve

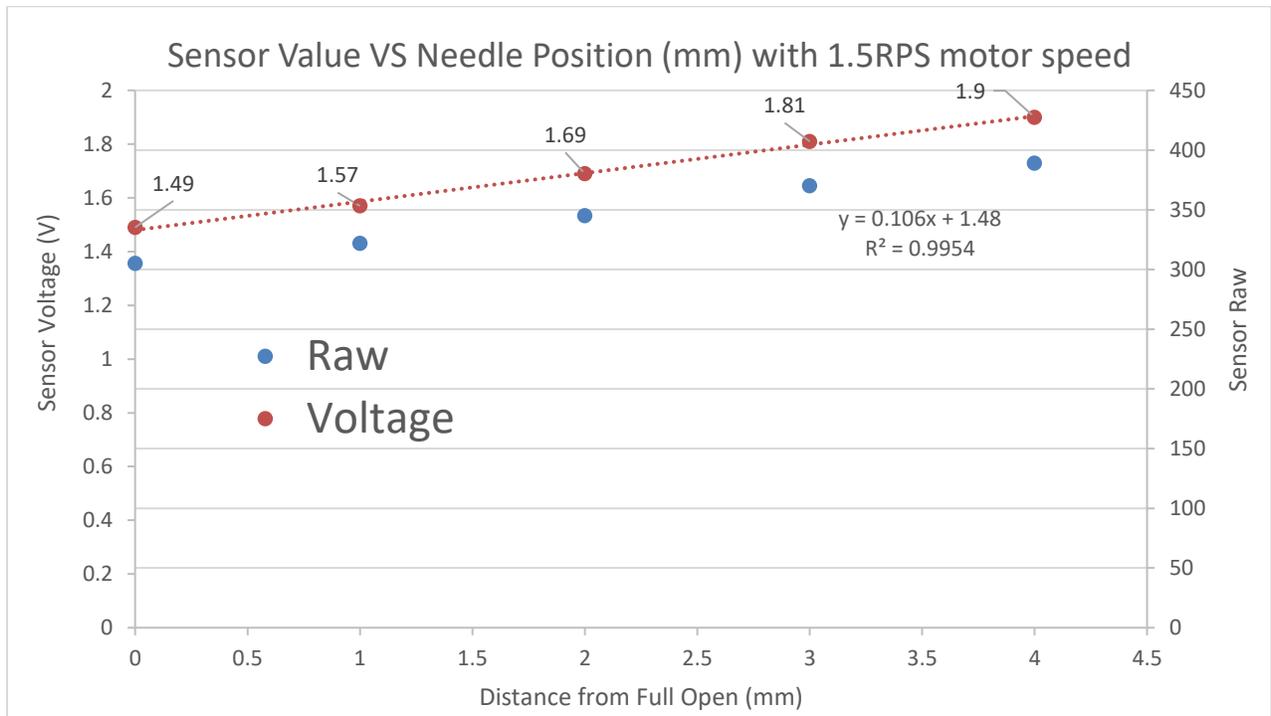


Figure 4.19: Sensor Output (raw, V) VS Distance (mm) from Fully Opened Valve with 1.5RPS Motor Rotation

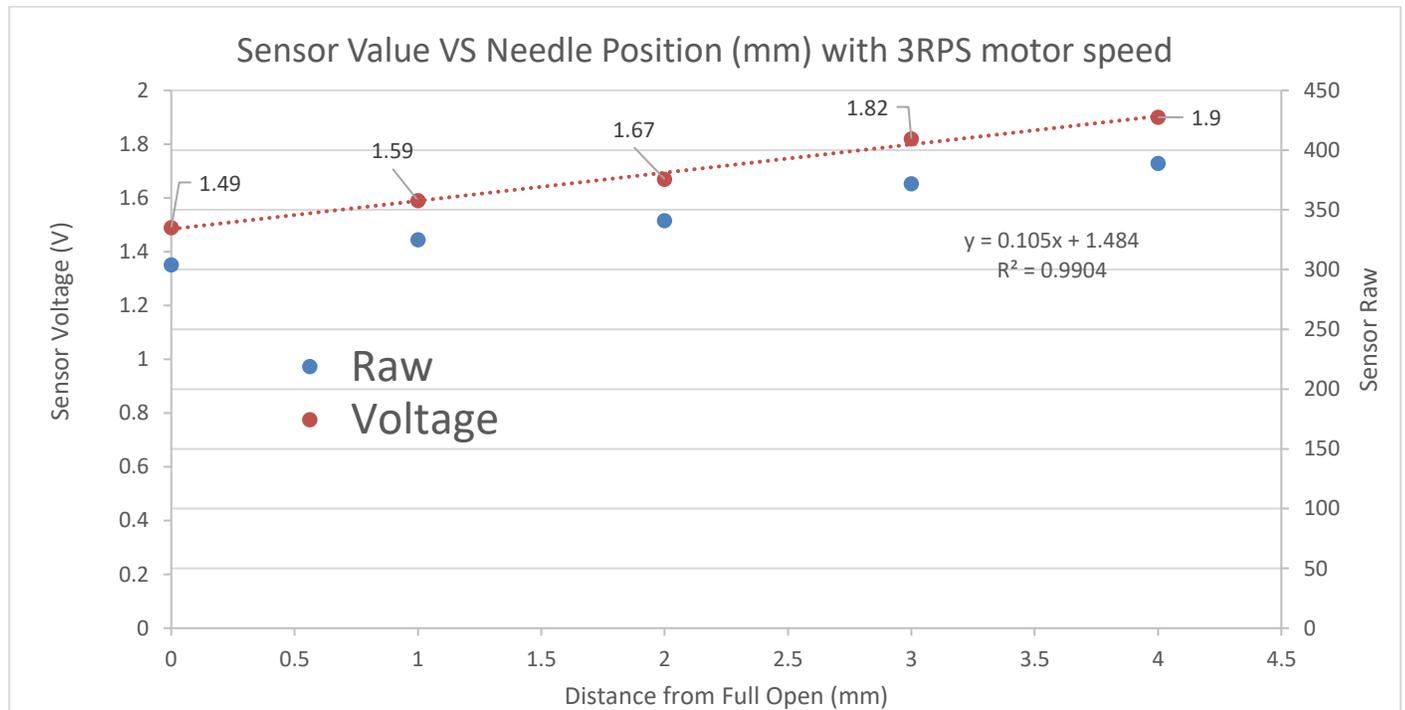


Figure 4.20: Sensor Output (raw, V) VS Distance (mm) from Fully Opened Valve with 3RPS Motor Rotation

4.3 Final Assembly

The final stages of the project mainly involved analysing the performance of the test rig as a complete assembly. This section describes the experiments performed to finalise the overall assembly, as well construction and programming of the custom PCB.

4.3.1 Mechanical

Towards the later stages of experimentation, while the custom aluminium valve was still being manufactured, several additional experiments were performed regarding 3D printing the valve with a threaded interface with the. The purpose of this was to create a demonstration model to prove the ability of the motor to control the valve, without access to the machined valve. This would allow closing of the feedback loop between the expected motor position and the actual position as measured by the sensor. The thread was added to the CAD file by using the “CoolOrange” program extension, which could create appropriate thread models from the standard Inventor thread tools.

The first attempt at 3D printing a custom valve involved simply printing the designs submitted to the machine workshop, including a 1mm pitched thread. Unfortunately, attempting to print such a fine thread caused several significant issues as shown in Figure 4.21. Due to design features of both the needle and block, it was necessary to use supports to prevent sections from collapsing mid-print. However, this also caused the printer to attempt to print supports all throughout the threads, which completely clogged them to the point of failure. This is most noticeable in the block structure, where the slot has been completely clogged with support and is incapable of being used at all. Similarly, the external thread on the needle experienced significant clogging and warping, which prevented it from functioning as desired.

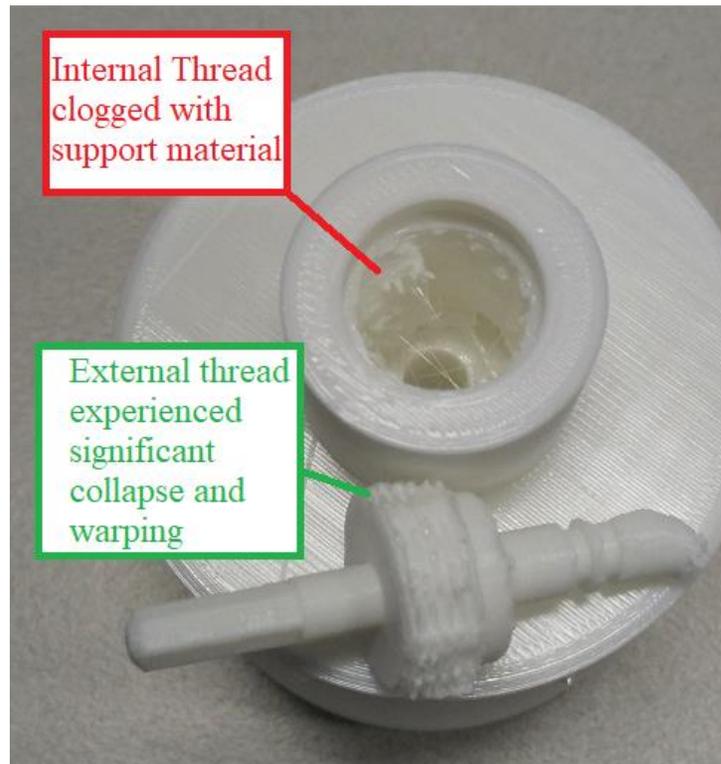


Figure 4.21: Failed First Attempt at 3D Printing Custom Valve with 1mm Threads

This procedure was reattempted with a 2mm thread instead of 1mm and a redesigned block and needle to remove the need for printing supports. Figure 4.22 displays the results of the second attempt, which produced a more successful threaded valve. This required removing some of the features that would have been included in the aluminium block to remove the need to print with plastic supports. The socket intended to let oil out of the valve needed to be plugged with a generic extrusion, so it did not print and sag. Additionally, the nose section of the needle needed to be entirely removed so the print could start flat from the threaded section.

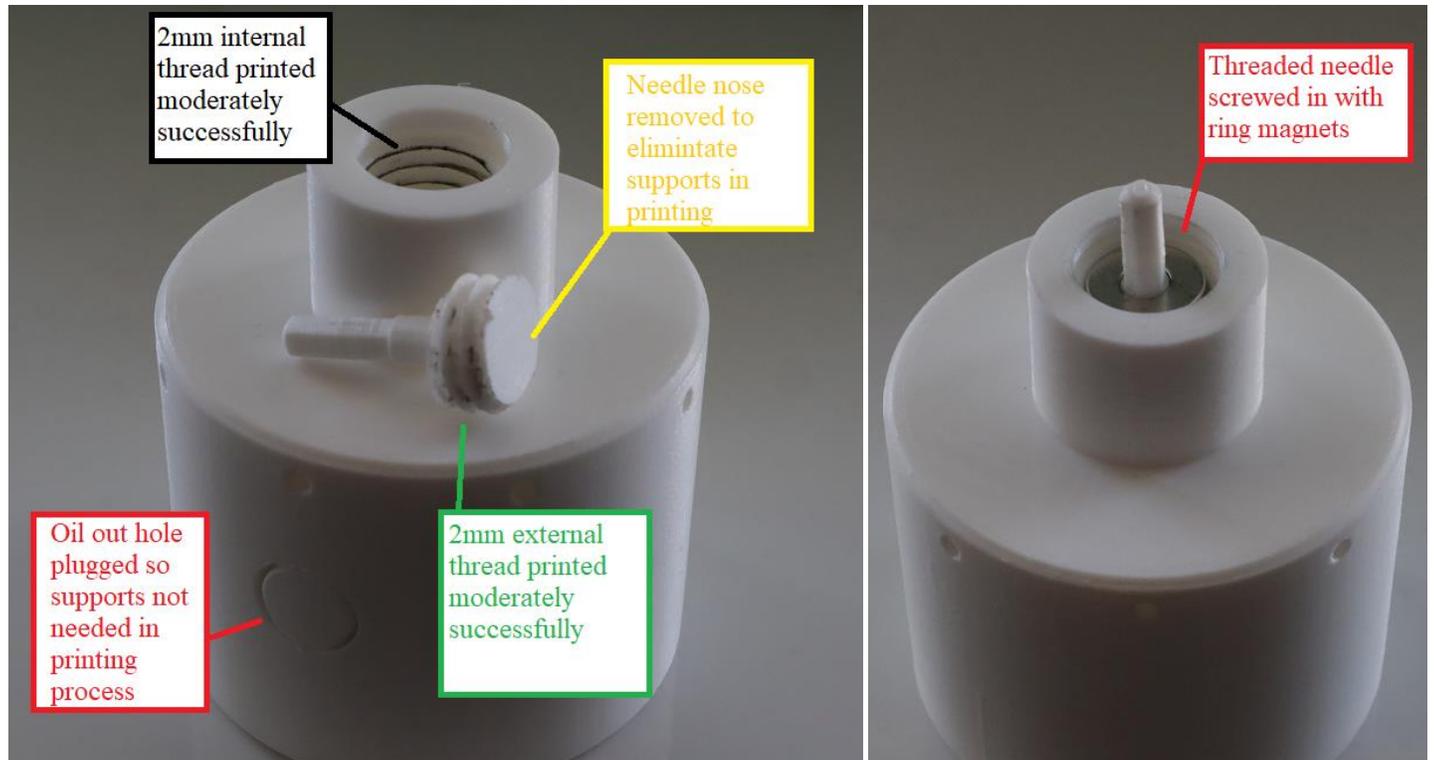


Figure 4.22: Second Valve 3D Print Attempt with 2mm Threads

Such modifications clearly remove the ability of the valve to function as a valve, however they provide a demonstration of the motor control ability and the screw motion. The half needle is capable of winding through the entire thread length, however the thread is of low quality. Much of the thread surface is grainy and unclear from the print, which makes certain areas of the thread motion very “crunchy” feeling and difficult to move. Such a low thread quality may react non-linearly with a motor driving it, as certain regions of the thread will require more torque to overcome.

This alternative solution created a successful but non-ideal demonstration of the motor and thread motion to test with while awaiting the production of the aluminium valve. Overall the 3D printed thread was functional but low quality, where certain regions required much larger torque to rotate. Additionally, the 2mm thread

pitch is a suitable but non-ideal representation of the 1mm thread pitch in the original machined block designs. This has the downside of halved precision control, where one rotation controls 2mm of linear movement instead of 1mm.

4.3.2 PCB

The electronic section of this project was managed by a single custom PCB. This was designed to integrate components for the motor, sensor and other beneficial peripherals. This section details the assembly and fault-finding stages of the PCB construction. Figure 4.23 displays the blank custom PCBs before any work has been performed.

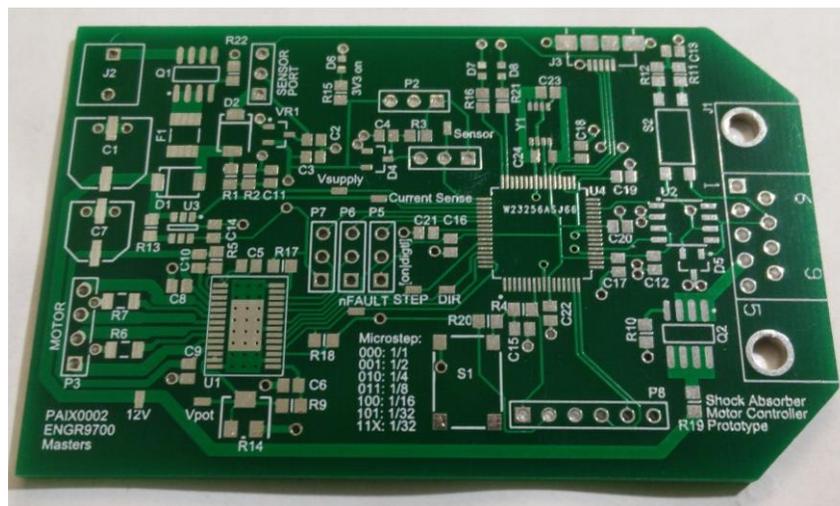


Figure 4.23: Blank Custom PCB

4.3.2.1 Assembly Process

Assembling the PCBs was performed using a combination of soldering tools. Overall, components for five boards were supplied but only two boards were produced to save time. Firstly, the pads of the surface mounted components were coated with solder paste by using the included stencil and an available solder-spreading tool. Next the surface components were individually placed using tweezers, beginning with the large multi-pin integrated circuits. After all the surface components were placed, the boards were cooked with a reflow oven.

When the boards were removed, there were several solder bridge short circuits that had formed between legs on both the microcontroller and the motor driver. These were removed using a dry solder soldering iron to

carefully extract the excess solder and scrape it into solder cleaning sponge. After the surface mounted components had been completed, the thru-hole components were connected using a soldering iron.

4.3.2.2 Fault Finding

During the process of assembling the PCBs, multiple faults were discovered with severity ranging from benign to significant. While assembling the PCB, it was noted that the expected 4.12K resistors for R9 were not provided. This was suspected to be a stock order error and was solved by stacking two 8.2K resistors on top of each other in parallel and hand soldering them together as shown in Figure 4.24.

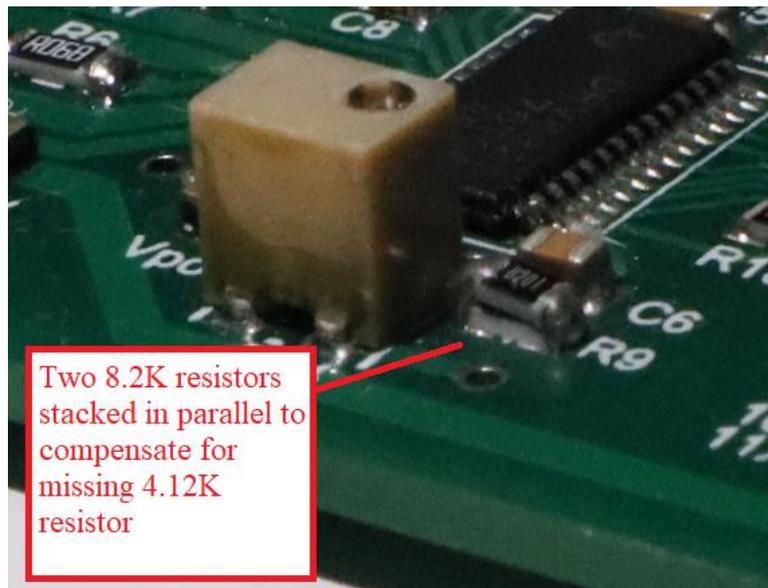


Figure 4.24: Stacking Two 8.2K Resistors to Compensate for Missing 4.12K Resistor R9

Additionally, another suspected parts order error caused there to be only one switch provided for S2, which meant that only one PCB could include a switch. This was approached by soldering bridging wires across the switch terminals on PCB 2 to make it permanently on. The purpose of this switch is to allow the user to select whether this device will be the final node in a CAN network, in which case the 118 Ω end termination resistance (R11 and R12) will be engaged. In the case for PCB2, the device has become a permanent end node, however this can be disconnected by cutting or desoldering the bridging wires.

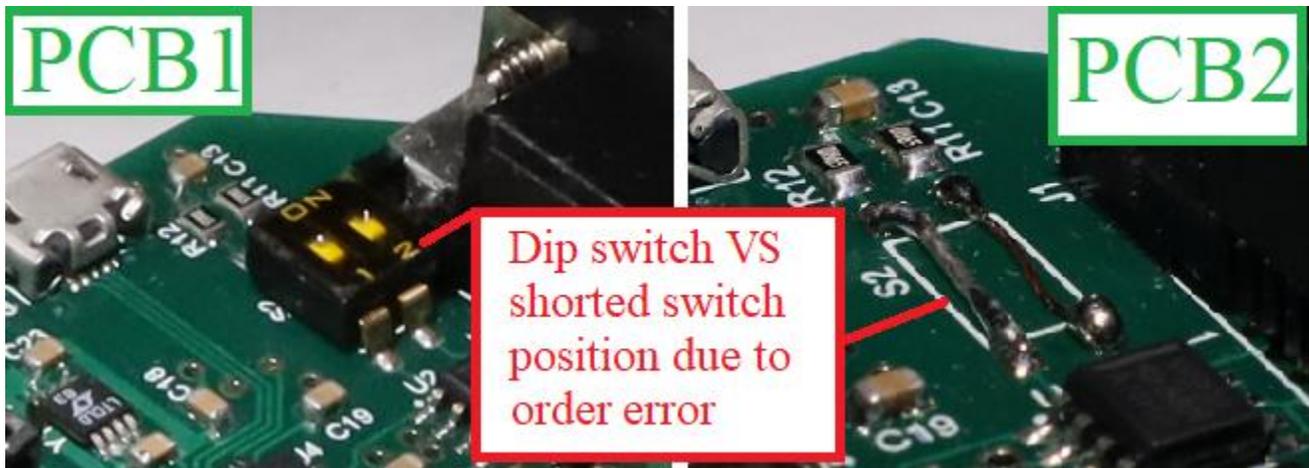


Figure 4.25: PCB1 and PCB2 Demonstrating Dip Switch S2 and Shorted Switch Terminals

After the PCBs had been fully assembled, they were powered on using a 12V lab bench power supply and current limited to 100mA to check for short circuits. On first power-up, the main power LED illuminated, and the board was measured consuming 10mA of current, which implied that there were no short circuits present. Following this, the motor and sensor components were tested by plugging in the corresponding peripherals and attempting to operate them.

Testing the sensor revealed no observable errors, however attempting to control the motor appeared to produce no output. This was performed by powering the PCB with a 12V and current limit of 1.7A, then applying a 3.3V square wave to the step pin via the appropriate test point. A function generator was used for this purpose and was intended to bypass the microcontroller, so any errors could be more easily isolated. It was observed that the stepper motor driver was successfully being powered, as the 3.3V charge pump on pin 15 was active, however it was not outputting any current to the motor.

Eventually it was discovered that the source of the error was an incorrectly wired reset pin on the driver. Pin 16 represents the reset pin, which disables the device and zeroes the step count when pulled low. Unfortunately, during the design process this had been wired directly to the grounded power pad, when it instead should have been wired to the 3.3V charge pump on the adjacent pin 15 (see Figure 4.26). This meant that the device was constantly held in reset and was not able to function, which was considered a significant design fault as it prevented a major function of the PCB from operating.

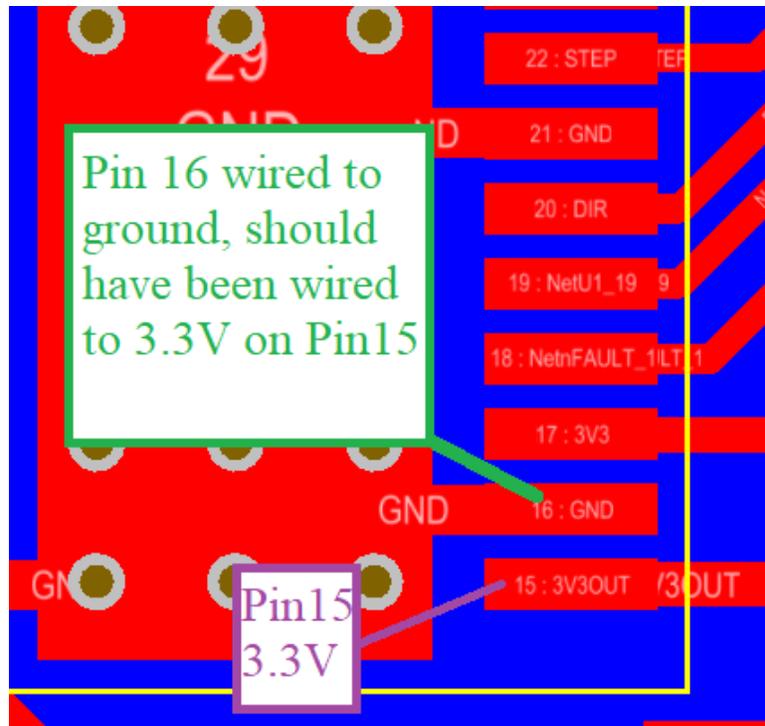


Figure 4.26: Pin 16 Wired to Grounded Power Pad Instead of 3.3V Pin 15

As the driver chips had already been soldered to the pads, this meant that solving the issue would either require very careful and precise rework under the chip or removing the entire chip to expose the area more. It was decided that this would be approached without removing the chip, which was performed using very sharp tweezers, a scalpel tip and a microscope. These were used to carefully scrape away the unwanted ground trace while preserving the driver pin. Due to the nature of this error, it was very important that the pin was completely disconnected from the ground trace, otherwise the 3.3V charge pump on pin 15 would be shorted to the ground plane and cause more errors.

Eventually, the trace was fragmented and scraped away enough that the pin was no longer connected to ground. This was confirmed using the continuity mode of a multimeter. Following this, pin 16 was carefully bent towards pin 15 and shorted to it with a blob of solder. This procedure was repeated for PCB2 (Figure 4.27) and the remaining blank PCBs were corrected by digging up the offending pads and traces to prevent the same issue reoccurring in future board assembly (Figure 4.28).

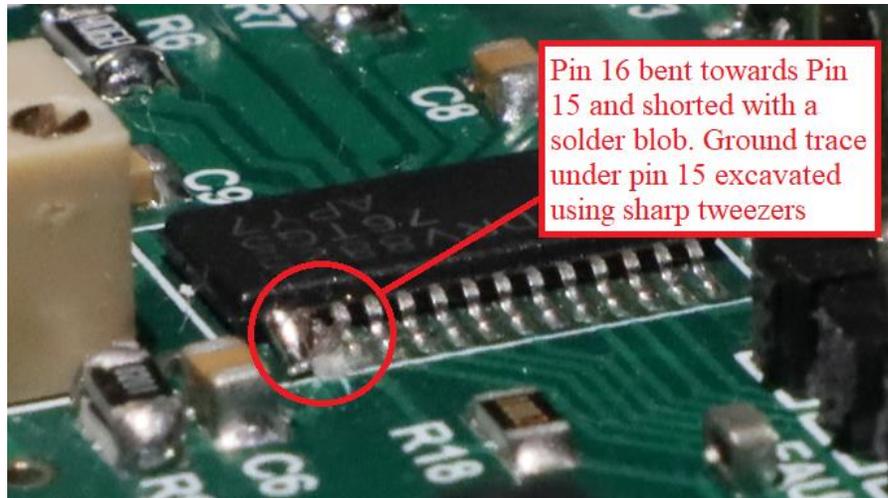


Figure 4.27: Aftermath of Removing Unwanted Ground Trace and Shorting Reset Pin to Adjacent Charge Pump Pin

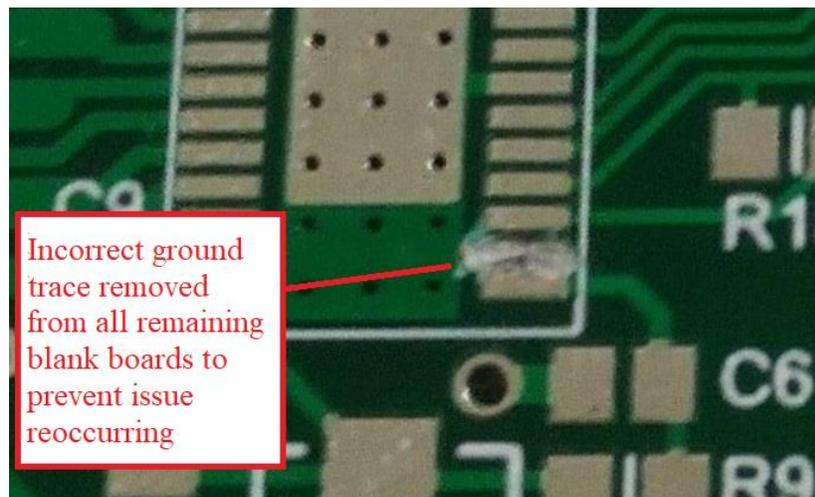


Figure 4.28: Removed Ground Trace from Remaining Blank Boards to Prevent Issue from Reoccurring

This section detailed the identification and resolution of the faults within the PCB, after which the board could perform all intended functions.

4.3.2.3 Sensor

After powering the device and confirming there were no short circuits, the hall effect sensor circuitry was tested. This was performed by using the same test performed previously to measure the sensor voltage response across the range of valve needle movement. The test rig assembly was formed with the 3D printed long needle and was shifted by 1mm increments between valve fully opened and full closed. Measurement was performed by bridging between the sensor test point and the Arduino Uno analog pin A0, then using the previous sensor test code.

Figure 4.29 displays the measured output response of the sensor over the valve range of motion. This demonstrates that the sensor output still responds linearly with a similar range compared to the non-filtered test, however the absolute voltages have decreased slightly. Previously, the sensor voltage range was measured between 1.49V and 1.89V, which decreased to 1.34V to 1.73V with the PCB low-pass filter setup. This minor difference was considered negligible as it did not affect the ability to measure the sensor in the complete assembly.

Another interesting observation formed during testing of the low pass filter circuitry was that the sensor noise appeared to increase compared to the non-filtered circuit. This was unexpected as the filter was specifically intended to reduce noise. This was observed as larger value fluctuations of ± 10 (32mV) on the Arduino serial monitor, which was 10x larger than the ± 1 (3.2mV) fluctuations measured without the PCB filtering. Such an observation lead to the trade-off that if a highly precise sensor value is required during operation, the filtering circuitry should be bypassed using the jumper on header P1. Otherwise, if this device is likely to encounter electrostatic discharge and random surges, it is more important that the filter remain engaged to allow the protection diode D4 to act.

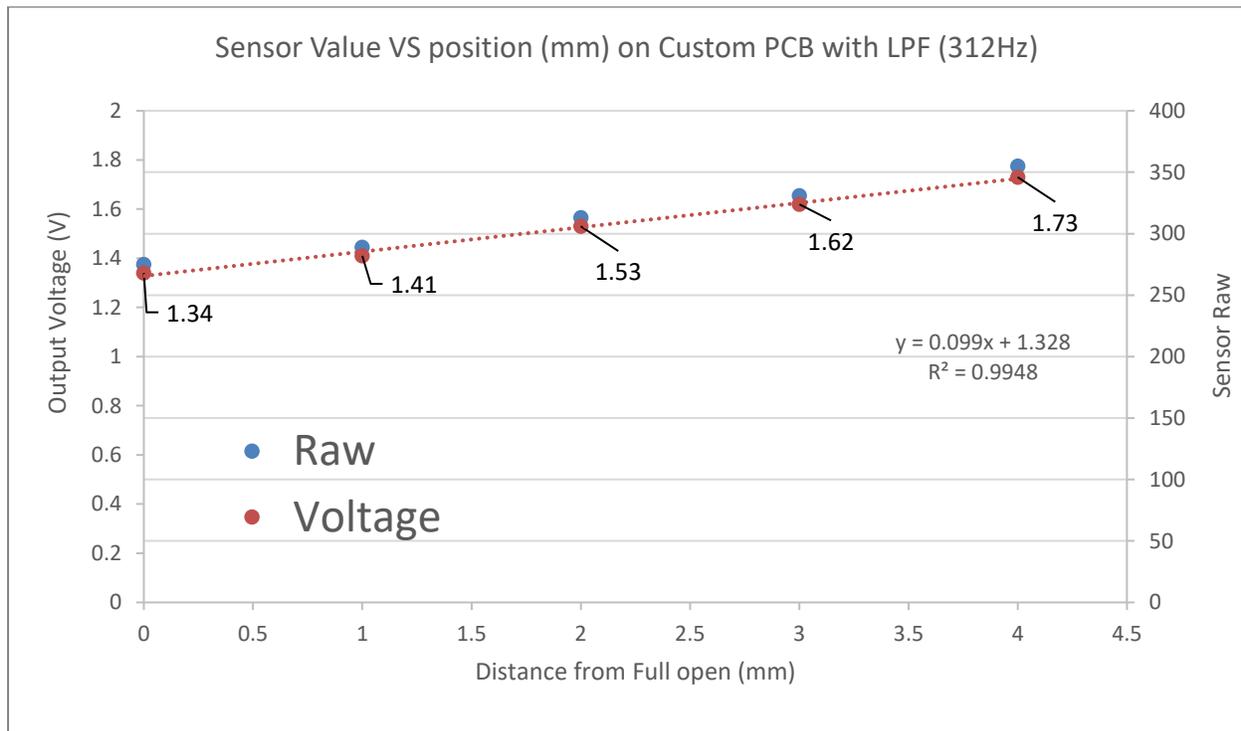


Figure 4.29: Sensor Value and Voltage Over Valve Motion Through Custom PCB Filtering Hardware

4.3.2.4 Motor

Once the motor driver trace error had been corrected, the driver operation could be tested. Initially, it was required that the driver output current was set by configuring the voltage on test point Vpot. This was performed by probing the Vpot test point with the device powered and adjusting the trimpot until the voltage read 0.11V; corresponding to 1.62A.

Next, the motor was connected to the header socket and the driver was activated. This was performed again by bypassing the microcontroller and applying a 3.3V square wave to the step pin via the test point with a function generator. It was observed that the motor driver could successfully drive the motor and consumed 250mA during smooth operation. Following this, a 3.3V voltage was applied to the direction pin while the motor was rotating, which caused it to change direction as expected. Finally, the microstepping headers were adjusted to test all possible microstepping settings, which successfully caused the motor to operate at each corresponding microstep rate.

This confirmed that, after correcting the reset pin wiring issue, the motor driver was successfully able to drive the motor with all expected functions.

4.4 Software

After testing the peripheral components of the PCB, the microcontroller was programmed and tested. This was performed using the MPLABX IDE to write the software, and individually PICKit3 and ICD3 debuggers to interface with the programming port. This section explores the design choices made in the microcontroller software and their relation towards the operation of the device. The complete code listing can be found in Appendix 8.

4.4.1 Initialisation

Initialising the microcontroller hardware and defining macros was performed using the code shown in Figure 4.30. This configured the microcontroller to use the external 8MHz oscillator with PLL enabled to increase the base speed. The PLL settings were configured to multiply the base oscillator frequency up to 48MHz, which was the highest possible combination for this hardware arrangement within the maximum limit of 50MHz.

```

#pragma config FNOSC =      PRIPLL      //FRCPLL for internal PRIPLL for external
#pragma config FSOSCEN =   OFF
#pragma config IESO =      ON
#pragma config POSCMOD =   XT           //OFF for internal, XT for crystal
#pragma config OSCIOFNC =  OFF
#pragma config FPBDIV =    DIV_1
#pragma config FCKSM =     CSDCMD
#pragma config WDTPS =     PS1048576
#pragma config FWDTEN =    OFF
#pragma config WINDIS =    OFF
#pragma config FWDTWINSZ = WINSZ_25
#pragma config FPLLIDIV =  DIV_2       //clock is now 4MHz
#pragma config FPLLMUL =   MUL_24     //multiply up to 96MHz
#pragma config FPLLODIV =  DIV_2       //divide down to 48MHz

#define dir LATEbits.LATE1
#define decay LATEbits.LATE2
#define step LATFbits.LATF1
#define driverFault PORTEbits.RE3
#define LED_RED LATCbits.LATC14
#define LED_YEL LATCbits.LATC13
#define clockwise 1
#define anticlockwise 0
#define SYSFREQ 4800000L                //system clock at 48MHz

```

Figure 4.30: Main Setup Code for Microcontroller Hardware and Macro Definitions

Following this, the software initialisation subroutine was written and is displayed in Figure 4.31. This performed output assignments of the digital I/O pins being used and configured the appropriate hardware features as desired. Timer 1 and 2 were both individually initialised with 1/256 pre-scaler, 16-bit counting and internal pulse source. Both were used later in the program to act as delay sources for the program.

Also included in this section is the initialisation of the analog to digital converter (ADC), which was configured to cycle repeatedly through analog pins AN24, AN25 and AN26. These corresponded to the hall effect sensor, scaled input voltage and current sensor voltage respectively. The ADC was configured to automatically acquire and convert for each measurement period, which simplified the reading process.

```

void init(void)
{
    TRISCbits.TRISC14 = 0;           //set RC14 output
    TRISCbits.TRISC13 = 0;           //set RC13 output
    TRISEbits.TRISE1 = 0;           // set DIR pin output
    TRISEbits.TRISE2 = 0;           // set Decay pin output
    TRISFbits.TRISF1 = 0;           // set step pin output
    TRISDbits.TRISD4 = 0;           // set microstep pin 0 output
    TRISDbits.TRISD5 = 0;           // set microstep pin 1 output
    TRISDbits.TRISD6 = 0;           // set microstep pin 2 output
    PORTD = 0;                       // set micro step size 1

    OC1CON = 0x5;                     //OCC pwm mode
    T1CON = 0;                         //TIMER 1 1/256 PRESCALE, INTERNAL, 16bit
    T2CON = 0;                         //timer 2, 1/256 pre, internal, 16bit

    LED_RED = 0;                       //red led off
    LED_YEL = 0;                       //yel led off
    step = 0;                           //set step pin low default
    dir = clockwise;                   //set direction clockwise default
    decay = 0;                         //set default decay mode = 0

    //----- ADC initialisation -----
    AD1CSSLbits.CSSL24 = 1;           //configure A2D to read AN24, AN25, AN26
    AD1CSSLbits.CSSL25 = 1;           //
    AD1CSSLbits.CSSL26 = 1;           //

    AD1CON1 = 0b0000000011100100; //16bit integer, auto convert,
    AD1CON2 = 0b0000010000010000; //scan all muxA, 16bit buffers
    AD1CON3 = 0b0000111100000001; //clock from PBCLK, 31TAD
    AD1CON1bits.ADON = 1;           // turn on the A2D
}

```

Figure 4.31: Software Initialisation Subroutine

4.4.2 Motor Driving

The first peripheral focus for the custom PCB was the ability to drive the motor, which effectively required a square wave signal on the driver step pin with configurable frequency. This was performed by defining two delay routines as shown in Figure 4.32, which respectively used timer 1 and timer 2 to wait for 1ms and 5 μ s. The 5 μ s timer was chosen at this timeframe as it was the smallest achievable time frame that remained accurate at high time delays. Both functions operated using an input count value, which acted as a multiplier for each timer cycle. These timing limit values were determined experimentally using an oscilloscope to probe the step pin and adjusting the delay until accurate.

```

void wait_ms (int count)           //delay 1ms using timer1
{
    int i;
    T1CONbits.TCKPS1 = 0;          //set timer prescale to 1/1
    T1CONbits.TCKPS0 = 0;
    T1CONbits.ON = 1;             //turn on timer
    for(i = 0; i < count; i++)
    {
        while(TMR1 < 47985);      //ticks for 1ms: 47985
        TMR1 = 0;
    }
    T1CONbits.ON = 0;             //turn off timer
}

void wait_5us (int count)          //delay 100us using timer1
{
    int i;
    T2CONbits.ON = 1;             //turn on timer 2
    for(i = 0; i < count; i++)
    {
        while(TMR2 < 111);       //ticks for 5us: 111
        TMR2 = 0;
    }
    T2CONbits.ON = 0;             //turn off timer
}

```

Figure 4.32: Wait 1 ms and Wait 5 μ s Delay Subroutines

Next, the function shown in Figure 4.33 was created and could be called from the main loop. This allowed the user to specify the number of steps, direction of rotation and speed in steps per second of the motor motion. Furthermore, Figure 4.34 provided a function to configure the microstepping setting using a single integer input. Additionally, this function contained corrections for if the user entered a number that was out of bounds or did not match a possible microstepping size. If the selected setting exceeded the allowable range, it was rounded back to the nearest possible limit. Similarly, if an invalid size was requested, the function defaulted to size 1.

Testing these functions successfully caused the motor to rotate, which validated the capability of the microcontroller to drive the motor. It was also observed that since the enable pin of the motor driver is hard wired to a 3.3V source, whenever the motor is not being actively stepped, it remains in hold mode with strong torque. This is desirable as it means the motor can actively hold position against the hydraulic pressure of the valve in a test system.

```

void step_motor(int steps, int direction, int steps_per_sec)
{
    int delay = 202000 / steps_per_sec;
    int i = 0;
    dir = direction;
    for(i = 0; i < steps; i++){
        step = 1;
        wait_5us(delay);
        step = 0;
        wait_5us(delay);}
}

```

Figure 4.33: Step_motor Function to Move Motor at Specified Distance, Direction and Speed

```

void set_microstep_size(int size_change)
{
    //set microstep size. if outside bounds, bring in range. if invalid, choose 1/1
    //must have all headers set to "digt1" on PCB for full digital control.
    microstep_size = size_change;
    if(microstep_size < 1) microstep_size = 1;
    if(microstep_size > 32) microstep_size = 32;
    switch(microstep_size)
    {
        case 1:
            PORTD = 0b0000000;
        case 2:
            PORTD = 0b1000000;
        case 4:
            PORTD = 0b0100000;
        case 8:
            PORTD = 0b1100000;
        case 16:
            PORTD = 0b0010000;
        case 32:
            PORTD = 0b1010000;
        default:
            PORTD = 0b0000000;
    }
}

```

Figure 4.34: Set Microstepping Size with Corrections for Forbidden Values

4.4.3 Sensor

Once the motor was operating successfully with controllable speed, direction and distance, the system was extended to measure the hall effect sensor response across the valve motion range. This was performed by manually configuring the plastic threaded valve in fully closed position, then using the motor to fully open it. At the same time, the sensor with filtering hardware engaged was continuously polled using the code shown in Figure 4.35 and the response was plotted using the Arduino serial monitor.

```

void read_analog(void)
{
    while(!IFS0bits.AD1IF); // wait until all sampling complete
    AD1CON1bits.ASAM = 0; // disable sampling
    sensor = ADC1BUF0; // record hall effect sensor voltage
    current = ADC1BUF2; // record current sensor voltage
    voltage = ADC1BUF1; // record power supply voltage
    AD1CON1bits.ASAM = 1; // reenale sampling
    IFS0bits.AD1IF = 0; // clear interrupt flag created by sampling completion
}

```

Figure 4.35: Save Contents of ADC Buffers to Data Registers

Figure 4.36 displays the sensor voltage measured for a motion cycle of fully closed to fully opened, waiting three seconds, then returning to fully closed using the fastest possible motor speed of 1600 steps per second. The structure of this plastic valve meant that the total motion distance was 6mm and three rotations. The sensor response in Figure 4.36 demonstrates that the closed position produces a sensor value of 1.86V, open position produces 1.58V and the fastest possible transition time was 0.75 seconds. Interestingly, this voltage range was higher than that measured for Figure 4.29; more closely resembling the values measured for Figure 4.19.

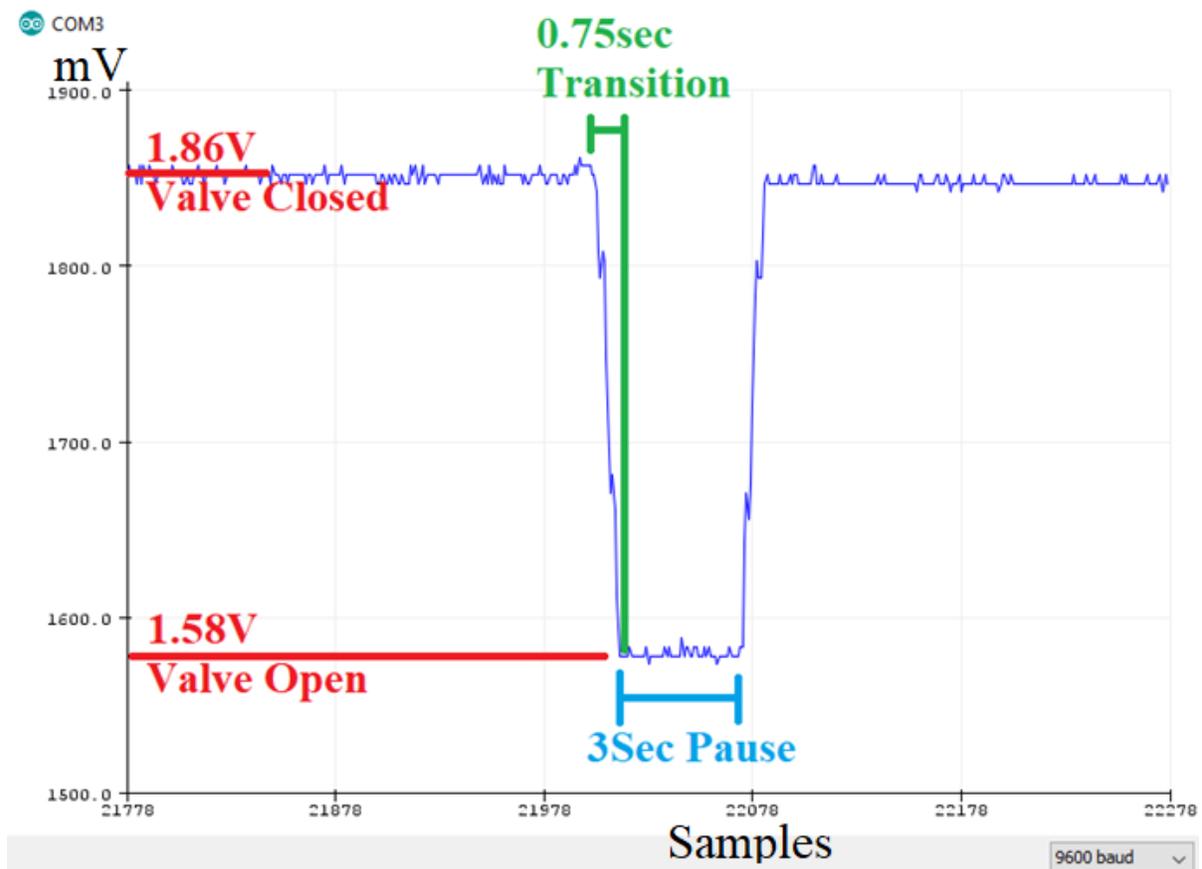


Figure 4.36: Sensor Response (mV) from Valve Full Sweep Motion

It is interesting to note that there were some observable non-linear voltage spikes during the transition range. This was explored further by repeating the valve position cycle over ten seconds, 1/4 microstepping rate and 480 steps per second speed. Figure 4.37 displays the sensor voltage response for these settings, which demonstrates the non-linear regions more significantly. These present an undesirable disturbance to the sensor response, as it makes it ambiguous to accurately determine the needle positions within those regions. It is likely that this is an effect of the poor print quality of the valve, where the magnet rings may be getting caught on the thread. This should be retested with the aluminium valve once it arrives to verify the source of the nonlinearity.

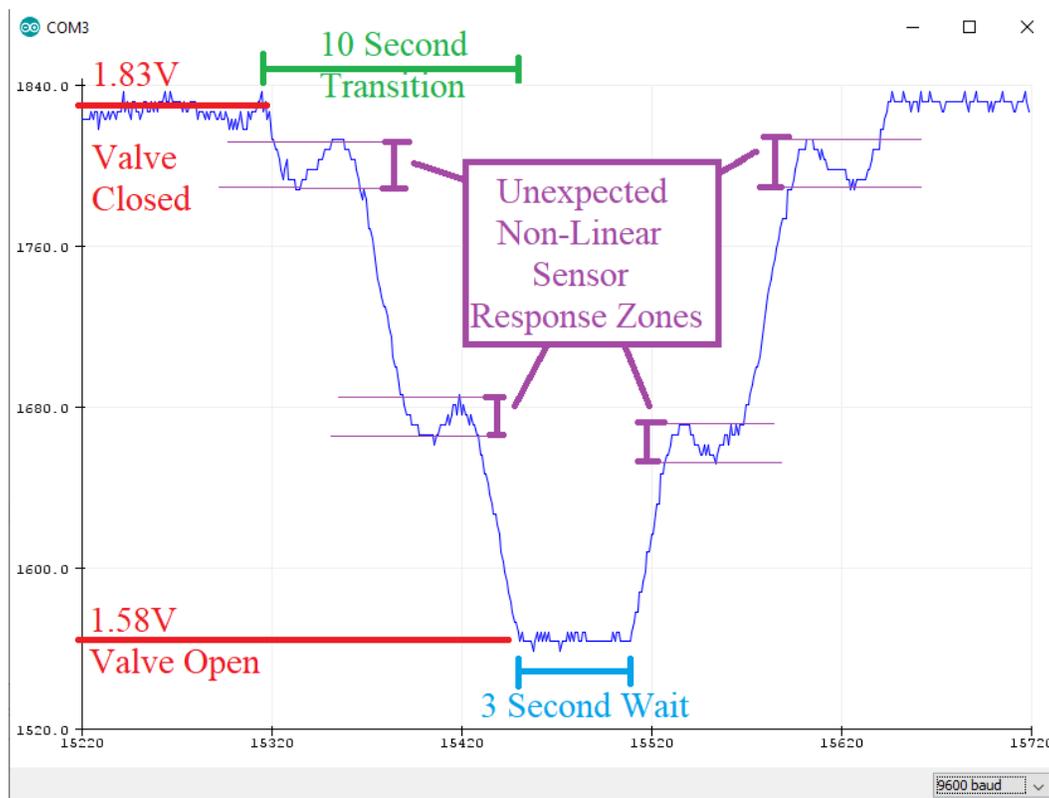


Figure 4.37: Non-Linear Sensor Response More Significant for Slow Motion

A further test was performed involving the sensor response with different motor microstepping settings. Figure 4.38 displays the sensor response for 1/1 to 1/16 microstepping at maximum speed with one second delays between each sweep. It is interesting to note that the faster speeds tend to be less affected by the nonlinear regions, where the 1/1 step size test was almost entirely linear. Additionally, the motion of closing the valve appeared consistently worse than opening for sensor linearity. This is likely because the positioning of the setup meant that valve opening was assisted by gravity, which made the motion smoother and encounter less resistance.

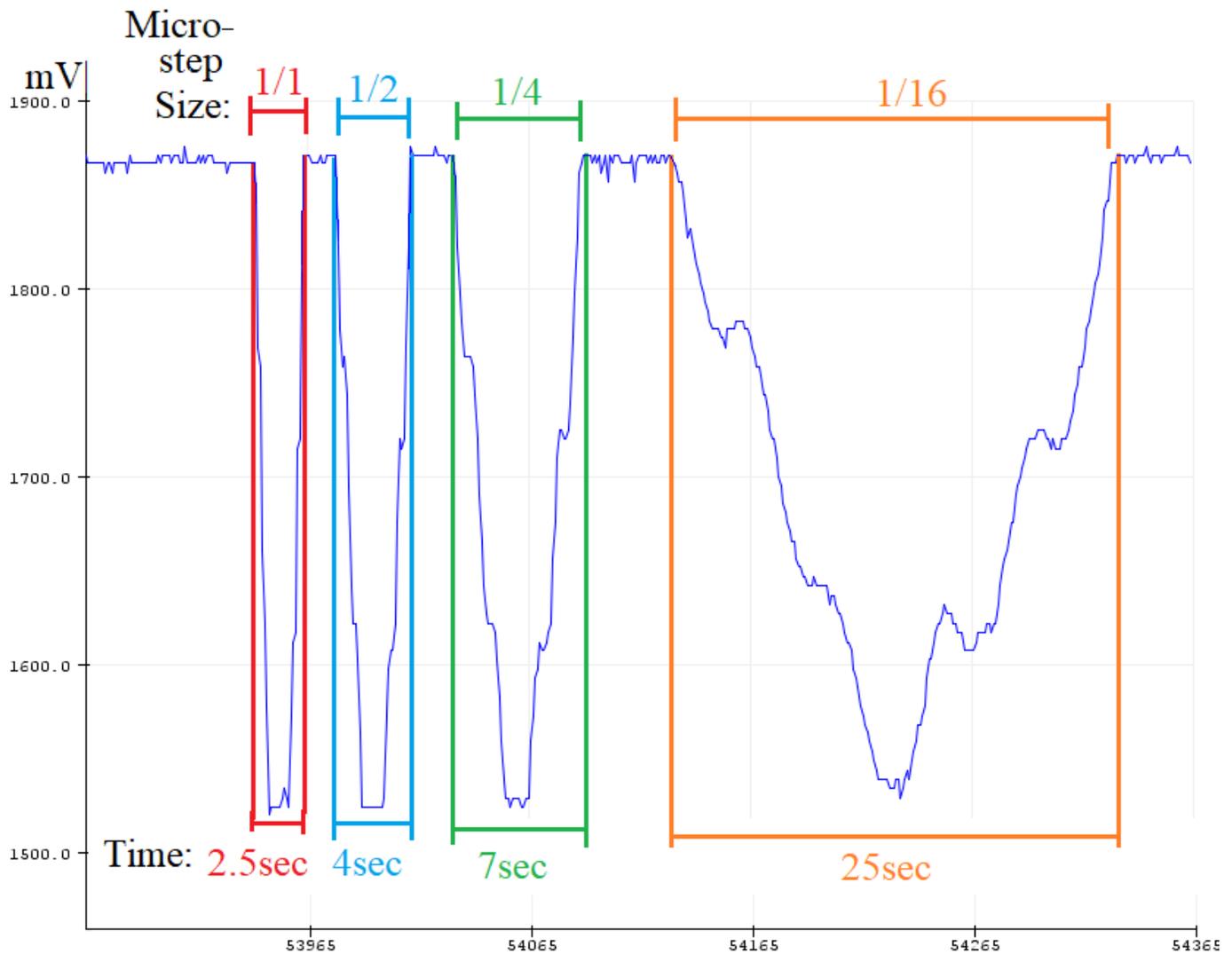


Figure 4.38: Valve Cycle Sensor Response for 1, 1/2, 1/4 and 1/16 Microstepping

These experiments demonstrate successful operation of the valve test rig. The motor was able to operate at a maximum speed of 1600 pulses per second, which equated to 0.75 seconds from full open to close in the plastic valve. The sensor responded as expected, although nonlinear regions appeared in low speed operation, which may lead to measurement ambiguities. This was hypothesised to be caused by the low quality of the 3D printed thread and expected that it will not occur in the machined aluminium valve.

4.4.4 Testing and Debugging Features

After testing the motor and sensor, other features of the PCB were tested. The V_{supply} signal on analog pin AN25 was measured using the ADC and produced an output value of 356. Using Equations 20-22, this ADC raw value has been used to calculate the supply voltage to the PCB. The expected input voltage was 12V, but the measured input voltage was 11.59V, which is considered acceptable as this voltage is still usable by the motor driver and linear regulator.

$$\text{ADC } V_{\text{supply}} = 356 \quad (20)$$

$$\therefore V_{\text{supply}} [\text{V}] = \frac{356}{2^{10} - 1} \times 3.3\text{V} = 1.148\text{V} \quad (21)$$

Given that this comes from a voltage divider with ratio 1.1:11.1

$$\therefore \frac{148 \times 11.1}{1.1} = 11.59\text{V} \quad (22)$$

Additionally, the motor driver current was measured using the analog voltage on pin AN26. Interestingly, this produced an excessively noisy waveform between 0-2.4V with no discernible information about the current, as shown in Figure 4.39. The signal shown in this figure was generated for the same valve microstepping sweep that produced the sensor data in Figure 4.38. It is unclear why the current measurement signal is so noisy, although unlike the other analog signals, it does not include any low pass filtering. This would likely need to be resolved in a future hardware iteration or compensated with low pass filtering in software.

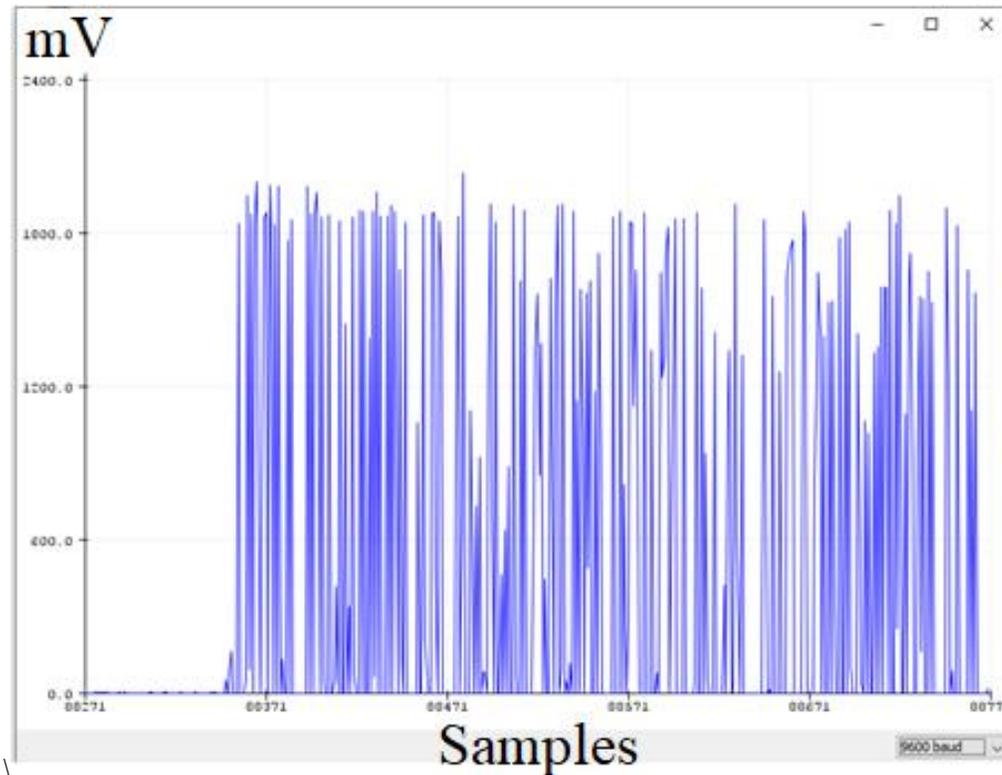


Figure 4.39: Current Sensor Output Signal Excessively Noisy

Another feature which was included in the software was to use the LEDs as notification conditions. Example uses of this were configuring the yellow LED to illuminate solid while the motor is rotating and the red LED to flash during error conditions. The error check was performed using the main loop structure and function displayed in Figure 4.40, where the presence of an error condition would cause the main loop to exit the action stages and flash the red LED three times quickly until the error was solved. The structure of this design is also easily expandable, where any additional error condition checks can be easily included in the error check subroutine. A future expansion of this feature may be to add unique identifier codes to each error source, which causes a unique flashing pattern on the LED to indicate for a specific error.

```
while(1)
{
    //-----check error condition-----
    if(!error_check())
    {
        //main code here
    }
    else
    {
        //flash red LED 3 times quickly if error true
        LED_YEL = 0;
        LED_RED = 0;
        for(i = 0; i<3; i++)
        {
            LED_RED = 1;
            wait_ms(100);
            LED_RED = 0;
            wait_ms(100);
        }
        wait_ms(500);
    }
}

int error_check(void)
{
    if(!driverFault) errorcode = 1; //if motor driver fault pin active, set error

    //include more error checks here

    return errorcode;
}
```

Figure 4.40: Main Loop Structure and Error Check Subroutine to Analyse for Faults and Flash Red LED If True

Figure 4.41 displays examples of the yellow and red LEDs active during motor motion and fault conditions respectively. In this case, the fault condition was triggered by disconnecting the plug during motor operation, which triggered the nFAULT pin on the motor driver.

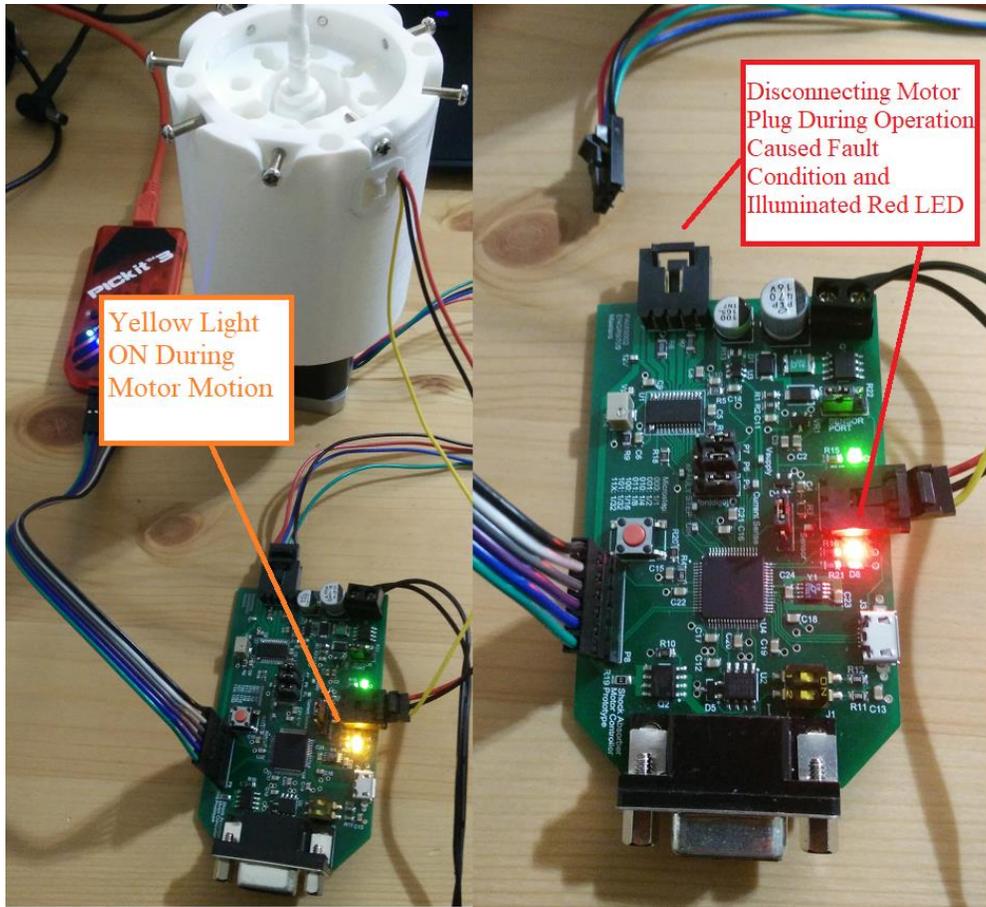


Figure 4.41: Yellow LED Illuminates During Motor Operation and Red During Fault Condition

5. CONCLUSIONS

This project explored the development of a PCB and custom test rig for Supashock to demonstrate the capability of electromechanical control of dynamic shock absorber parameters. This was performed in distinct mechanical and electronic design stages and three distinct assembly stages. Resources were used from both Flinders University and Supashock.

The mechanical test rig was designed to form a demonstration of a motor to control hydraulic oil pressure. This was performed with a custom hydraulic valve and corresponding motor and shaft mechanism. The complete assembly was successfully prototyped with 3D printed components and was demonstrated as fully capable of performing the intended mechanical motion. The mechanical structure of the test rig could successfully control the valve position using the stepper motor, however the performance was limited by poor quality 3D printed threads. Further testing will be performed with the higher-quality machined aluminium valve once machining has been completed.

A custom PCB was designed and fabricated to provide control of the stepper motor and a hall effect sensor within the mechanical assembly. This was initially prototyped using Arduino and related hardware, which successfully validated the intended function. Initial testing of the PCB revealed several notable faults; however, these were all resolved by PCB rework. After repairs, the PCB could successfully control the motor and read the analog sensor. Overall, the valve could be driven at a maximum speed of 1600 pulses per second, which equated to 0.75 seconds for a full 6mm sweep of the 3D printed valve.

The hall effect feedback sensor was demonstrated to provide a useful response to needle position between the voltage range of 1.58V to 1.86V, however there were also observable nonlinear ambiguous regions within the motion range. This was suspected to be a result of the poor thread quality, where the ring magnets were suspected getting caught on thread sections and disturbing the sensor value. This was expected to be overcome in future experiments on the more precisely machined aluminium valve. Further testing of the PCB features was performed, where the main input source voltage was calculated as 11.59V using the PIC ADC and the current sense amplifier was found to produce voltage signals with excessive noise and no discernible information.

Programming the onboard microcontroller was performed using the MPLABX IDE with PICkit 3 and ICD3 debuggers, with which a basic functional valve position cycling program was implemented. The final PIC program instructed the valve to cycle between full open and fully closed at maximum motor speed, with

varying microstepping sizes. This presented a suitable demonstration of the successful functioning of this test rig.

Overall, hardware and software mechanisms were developed which allow the for an electromechanical system to dynamically control shock absorber parameters. These demonstrated successful operation of both mechanical and electronic features, with further tests to be performed with hydraulic systems once additional components arrive.

5.1 Resource Estimate

During the experimenting process, unexpected circumstances and decisions caused changes to the original manufacturing plans. This meant that cost and resource estimates formed during the design stage do not accurately represent the true costs incurred. This section provides a current summary of the project costs and resource usage.

As of the time of writing, research is waiting for the manufacture of the custom aluminium valve. The original cost estimates for production remain accurate, however have not been applied at this stage. This means that no resources have yet been spent on this component, so the cost has been considered as \$0 at this stage.

Due to unexpected design faults, multiple 3D printed prototypes needed to be reprinted, which increased the total printing cost estimate. Two of the mounting shells were printed, which brings a total estimate of \$10.4 for 200g. Furthermore, the custom valve was printed three times to compensate for non-threaded versions and printing failures. Each valve block used approximately 33g of printing material, which meant a \$5.15 cost for 99g. Accompanying the valve blocks was one regular needle, one long needle and one half-needle. These respectively used 2.2g, 3.3g and 1g of material, which totals \$0.34. The 3D printed sensor bracket used 4.4g, which estimated \$0.23.

The additional assembly components, including the flexible shaft coupler, three ring magnets, stepper motor and hall effect sensor were purchased at their expected price of \$48.19.

As intended, the PCBs were purchased at a minimum order of five, with components to construct each of them. This meant that the \$72.16 PCB cost and $5 \times \$66.06$ component cost formed a total of \$402.46, which used funds allocated by Supashock. Although only two of the PCBs were constructed due to time limitations, the blank PCBs and components remain property of Supashock so can be assembled at any time in the future.

Overall, due to changes in the design plans, the total cost of the project resources was \$466.77, which was \$54.08 cheaper than the expected cost of \$520.85 and within the \$600 allocated budget.

5.2 Future Work and Recommendations

5.2.1 Testing

Further recommendations towards the project research are presented here. As the current stage of the project is awaiting the delivery of the custom aluminium valve, experiments involving this have not yet been performed. Once this is available and is applied to the test rig, a series of experiments should be performed to analyse the device performance with hydraulic pressure.

Experiments should be performed again to model the sensor response across the valve motion range. This has been suggested as the aluminium metal of the valve may affect the properties of the magnetic flux, which would affect the voltage response of the sensor. Additionally, the plastic valve was observed to encounter sensor non-linearities, which were attributed to poor quality 3D printed threads. This should be investigated further to ensure that the sensor responds linearly and non-ambiguously within the aluminium valve.

Once the properties of the test rig have been modelled in lab conditions, the system should be connected to the hydraulic testing setup proposed initially in Figure 1.2. Experiments should be performed to identify the limits of the device such as maximum allowable oil pressure, as well as modelling the motor speed and torque against a range of different pressures to indicate the performance of such a system in an actual vehicle.

5.2.2 Design Recommendations

During the development stages of the custom PCB, multiple aspects were identified for improvement in a potential future revision. As the PCB provides the most research benefit to Supashock and would form the basis of a potential future product, it is advantageous to consider as many potential useful improvements as possible. This section describes the recommended design corrections and additional features for the PCB to greatly improve the performance of a future application.

The most significant design errors to the performance of this PCB involved the stepper motor driver. It was found during testing that the reset pin had been wired to ground rather than 3.3V, which caused it to remain permanently in reset mode and thus unusable. This should be rewired such that this pin is connected appropriately to a 3.3V source or an available digital line of the microcontroller. Wiring both the enable and reset pins to microcontroller I/O pins would allow for better software control of the driver, which means the

motor could be switched off to save power. Currently the system uses a permanent-on configuration, which means that the motor is constantly applying power and is therefore either rotating or applying holding torque. The ability to disable motor power if desired may be advantageous to a shock absorber controller if a low power mode is desired.

Along with the enable and reset pins, the microstepping selection pins should be redesigned to only use software control. Currently, the system allows for hardware or software control via jumpers, which is advantageous in prototyping, but consumes extra space that may not be available in a vehicle system. Furthermore, a future version of this PCB that is mounted in a vehicle may not be physically accessible to change jumper positions, which reduces their usefulness. Using software-only control allows for easy adjustment of microstepping settings without needing to consider if the hardware jumper is configured correctly.

In the development of this project, the microstepping settings were demonstrated as operable but not implemented as part of a complete algorithm. In a future extension of this project that applies the system to a shock absorber, it may be advantageous to use microstepping functionality more extensively in the software to accurately control speed and position. An example of this would be to apply maximum speed when travelling a larger distance, then engaging half or quarter microstepping within a certain distance from the target to increase precision. The driver chosen for this project allows up to 1/32 microstepping size, which provides very high rotational precision, so the potential performance could be easily explored.

Many of the features of this PCB have been included for prototyping purposes and may not be necessary in a final product. Reducing or removing many of the features will allow for much of the circuitry to be reduced in size and power consumption, which is beneficial for miniaturisation for a vehicle system. All the selection jumpers on this PCB could be removed and replaced with circuitry for simplicity. The microcontroller power selection jumper allows the user to specify whether the microcontroller will receive power from the 12V input lines or the USB 5V supply. This could be improved in a future revision by applying both voltage sources to the input of the linear regulator and including switching circuitry to prevent 12V on the USB bus or 5V on the motor driver supply. Additionally, the voltage-set potentiometer on the motor driver could be replaced with a fixed resistor to permanently configure the motor current to the desired value, which would further reduce the PCB size.

The main 12V power supply connection applied to this device is a simple 2-pin screw terminal, which was chosen as it allows for maximum flexibility in providing power connections. This could be improved in a

future revision by replacing it with a dedicated power plug such as a DC barrel jack. Using a dedicated plug would reduce the need for reverse polarity protection, which could simplify the circuit overall.

Currently the mechanical design of the PCB does not contain facilities for any mounting. There were no mounting holes included in the design and the edges of the board mostly occupy external plugs, which makes clamping them difficult. A future revision of this board could improve upon this by including four mounting holes to support PCB standoffs and provide a more secure holding system. The result of the current lack of mounting options means that the board must be free floating adjacent to the test assembly without a secure attachment to anything. Alternatively, a custom enclosure box may be designed for the PCB, which would provide mounting and very high protection from environmental effects. As the automotive industry often involves contaminants and dirty environments, exposure may negatively affect the PCB. An enclosure box would therefore shield the circuitry from hazards and likely improve the lifespan.

Another possible upgrade to the circuitry for a future revision would be to include an integrated circuit temperature sensor. This would allow the user to easily measure the temperature of the surrounding environment, which would help to reduce the risk of overheating. Interestingly, some PIC microcontrollers contain a built-in temperature sensor diode, which can be accessed using the same ADC, however this was not the case for the microcontroller chosen for this design. If a different microcontroller was chosen which included a temperature sensor, this would reduce the need for external sensor components. In this case, the most significant heat source in the circuit is the motor driver, as it is handling up to 2.5A of current. Therefore, it is recommended that any additional temperature sensor ICs should be placed near to the motor driver for best accuracy.

Finally, this project forms the basis and demonstration for an electromechanical device to control a single shock absorber parameter. This does not provide any capability towards regulating the response of multiple shock absorbers to the road conditions, rather a separate top-level controller should be investigated as an extension project to perform master control of all shock absorbers. The device provided by this research contains a variety of hardware capabilities such as CAN communication and current sensing, so should be easily capable and simple to implement as a key stepping stone in a larger vehicle network project.

6. BIBLIOGRAPHY

- Bielski, S., Feichtinger, K., Spies, A., Strasser, E., Mitterreiter, J., Meyer, H., & Schroter, A. (1999). Multiturn rotary encoder with multiple code carriers coupled by a reduction gear, Patent number (US6542088B1)
- Condit, R., & Jones, D. (2004). *Stepping Motors Fundamentals*
- Digi-Key. (2016). How to Power and Control Brushless DC Motors. Retrieved from <https://www.digikey.com.au/en/articles/techzone/2016/dec/how-to-power-and-control-brushless-dc-motors>
- Dixon, J. C. (2007). *The shock absorber handbook* (2nd ed. ed.). Hoboken, N.J.: Wiley ; Chichester : John Wiley [distributor].
- Els, P., Uys, P., Snyman, J., & Thoresson, M. (2005). Gradient-based approximation methods applied to the optimal design of vehicle suspension systems using computational models with severe inherent noise. *Mathematical and Computer Modelling*, 43, 787-801.
- Frenergy.com.au, Neodymium Ring Magnet OD13 x H2 x ID5 mm N35, Part No: R-OD13H2ID5-N35, <https://www.frenergy.com.au/neodymium-ring-magnet-od13-x-h2-x-id5-mm-n35.html>
- Ganssle, J. (2012). A Designer's Guide to Encoders. Retrieved from <https://www.digikey.com/en/articles/techzone/2012/apr/a-designers-guide-to-encoders>
- Gordaninejad, F., & Kelso, S. (2000). Fail-Safe Magneto-Rheological Fluid Dampers for Off-Highway, High-Payload Vehicles. *Journal of Intelligent Material Systems and Structures*.
- Honeywell, (2015), SS39ET/SS49E/SS59ET Series Linear Hall-effect Sensor ICs, Component Datasheet, https://sensing.honeywell.com/index.php?ci_id=50359
- Hopkins, T. (2012). *Stepper Motor Driving*. STMicroelectronics
- Howard, M. (n.d.). *Incremental encoders, absolute encoders & pseudo-absolute encoders*
- Imaginables, 2019, Australian PLA- White, <https://imaginables.com.au/collections/filaments/products/australian-pla-white>
- Ivers, D., & Miller, L. (1991). *Semiactive Suspension Technology: An Evolutionary View*. ASME.
- Jolly, M., Bender, J., & Carlson, D. (1999). Properties and Applications of Commercial Magnetorheological Fluids. *Journal of Intelligent Material Systems and Structures*, 10.
- Keeping, S. (2013). *An Introduction to Brushless DC Motor Control*.
- Khan, T., Taj, T., & Ijaz, I. (2014). Hybrid stepper motor and its controlling techniques a survey. from IEEE

- Khan, Z. (2015). FAQ: What is the difference between absolute and incremental encoders? Retrieved from <https://www.motioncontroltips.com/faq-what-is-the-difference-between-absolute-and-incremental-encoders/>
- LittleBird. (2018). VL53L0X Time-of-Flight Distance Sensor Carrier with Voltage Regulator, 200cm Max. Retrieved from <https://www.littlebird.com.au/vl53l0x-time-of-flight-distance-sensor-carrier-with-voltage-regulator-200cm-max>
- National_Instruments. (2018). Measuring Position and Displacement with LVDTs. Retrieved from <http://www.ni.com/white-paper/3638/en/>
- Pololu. (2019). Stepper Motor: Unipolar/Bipolar, 200 Steps/Rev, 57×56mm, 3.6V, 2 A/Phase. <https://www.pololu.com/product/1473>
- Potter, J., Neild, S., & Wagg, D. (2010). Generalisation and Optimisation of Semi-active, on-off switching controllers for single degree-of-freedom systems. *Journal of Sound and Vibration*, 329(13).
- RobotShop. (2019). Banebots RS-540 12V 17200 RPM Brushed DC Motor, <https://www.robotshop.com/en/banebots-rs-540-motor.html>
- RSComponents. (2019a). Alps Electric RK1631110TWA Linear Carbon Film Potentiometer with a 6 mm Dia. Shaft, Panel Mount (Through Hole), 10k Ω . Retrieved from <https://au.rs-online.com/web/p/potentiometers/2499238/>
- RSComponents. (2019b). Vishay 534B1103JL 10 Turn 1 Gang Rotary Wirewound Potentiometer with a 6.35 mm Dia. Shaft, 10k Ω , $\pm 5\%$, 2W, Linear. Retrieved from <https://au.rs-online.com/web/p/potentiometers/1679487/>
- Schweber, B. (n.d.). Don't Ignore the Humble Brushed DC Motor. *Motor Control*.
- SeeedStudio.com. (2019). Emax XA2212 12V DC 980KV Brushless Motor, <https://www.seeedstudio.com/Emax-XA2212-12V-DC-980KV-Brushless-Motor-p-1899.html>
- Sensorland. (n.d.). THE LINEAR VARIABLE DIFFERENTIAL TRANSFORMER (LVDT). Retrieved from <http://www.sensorland.com/HowPage006.html>
- STMicroelectronics. (2017). Advanced 256 microsteps integrated motor driver with step-clock and direction interface.
- STMicroelectronics. (2018). DMOS driver for bipolar stepper motor.
- Storey, I. (2011). Performance Measures and Control Laws for Active and Semi-Active Suspensions. (Ph.D), The Royal Melbourne Institute of Technology (RMIT),
- Strasser, E. (2000). Multiturn rotary encoder with detector devices on a shared side of a printed circuit board, Patent (US6683545B2)
- Tein. (2013). EDFC Active, Retrieved from https://www.tein.com/products/edfc_active.html
- Texas_Instruments. (2014). DRV8825 Stepper Motor Controller IC

Todd, C. (1975). *The Potentiometer Handbook*: Bourns Inc.

Wagner, J., Liu, P., & Liu, X. (2000). *Nonlinear Modeling and Control of Automotive Vibration Isolation Systems*. Paper presented at the American Control Conference, Chicao, Illinois.

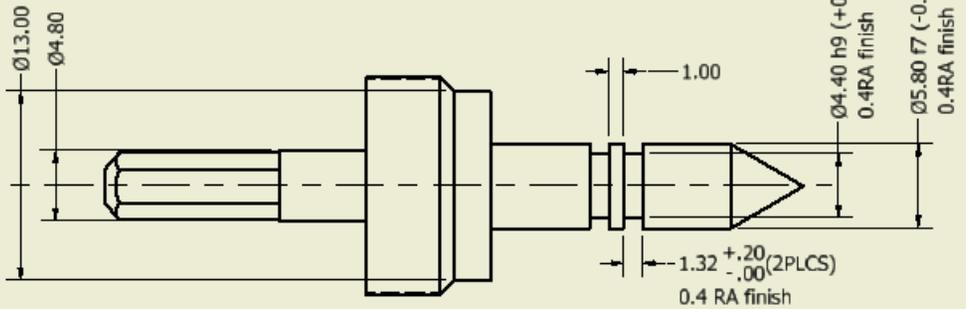
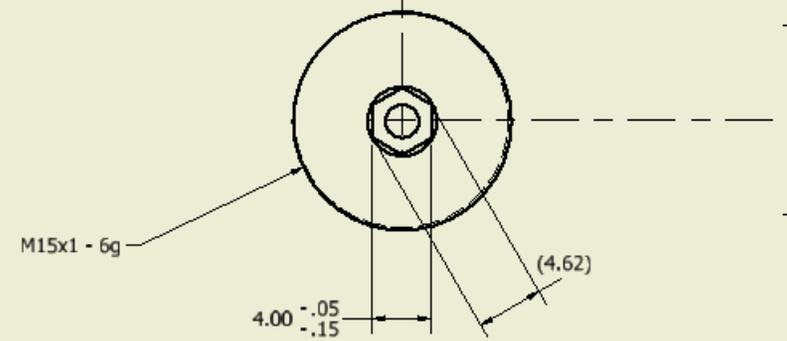
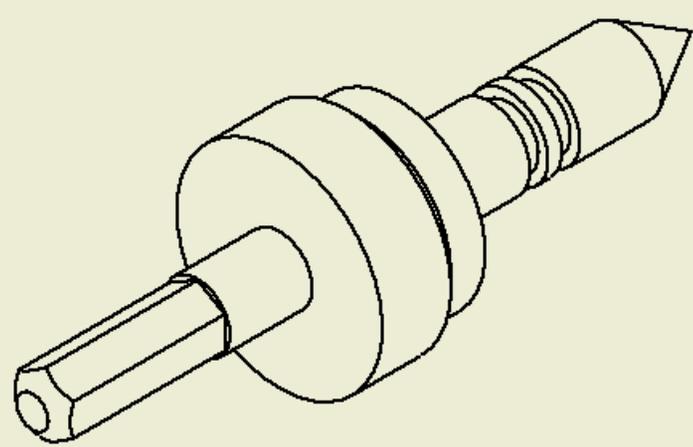
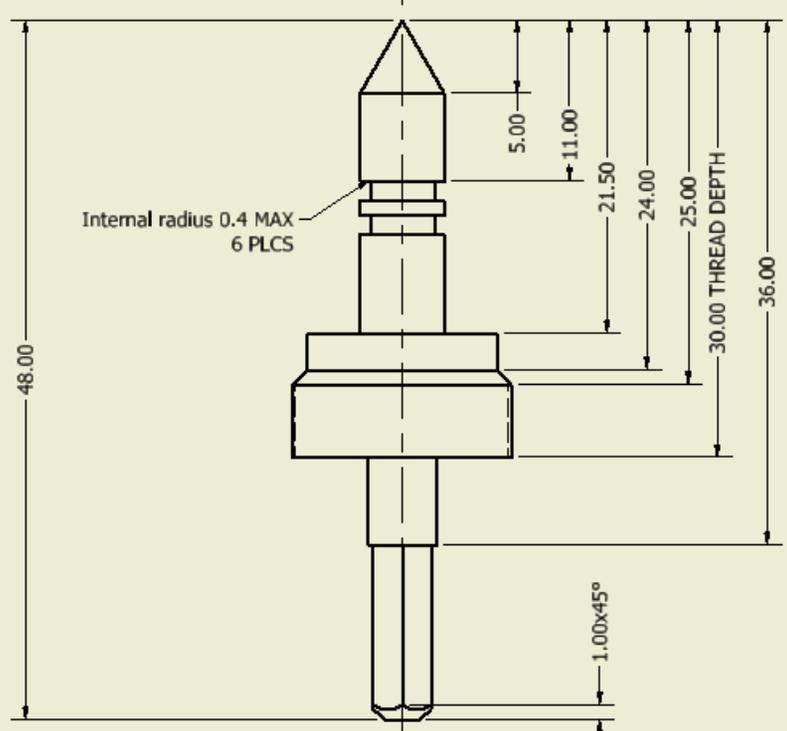
Wong, W. F., & Lee, S. K. (2008). *Inductive multi-turn encoder*, Patent (US7579829B1)

Zhao, J., & Yu, Y. (Producer). (2011). *Brushless DC Motor Fundamentals Application Note*.

7. APPENDICES

- 1) Valve Mechanical Drawings
- 2) Valve Seal
- 3) Valve Bill of Materials
- 4) Assembly Bill of Materials
- 5) PCB Schematics
- 6) PCB Bill of Materials
- 7) Sensor and Motor Test Code
- 8) Device Code

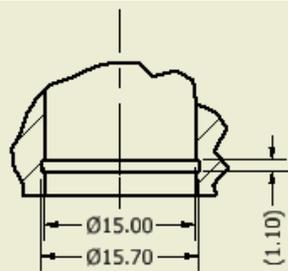
7.1 Valve Mechanical Drawings



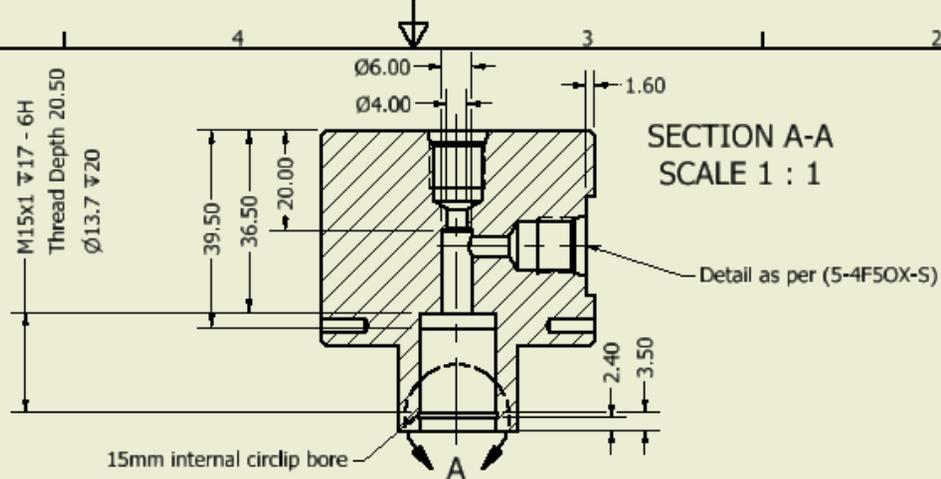
Mating components:
 -VB1: M15x1 thread
 -SC1: (provided) 4.00mm (4.62mm diagonals) hex plug, allow rotational coupling but linear sliding

Material: Aluminium 6061 (T6)
 Finish: 1.6 RA unless otherwise specified
 Tolerances unless otherwise stated:
 Linear: $\pm 0.1\text{mm}$
 angular: $\pm 0.5^\circ$
 all unspecified internal radii : 0.4mm

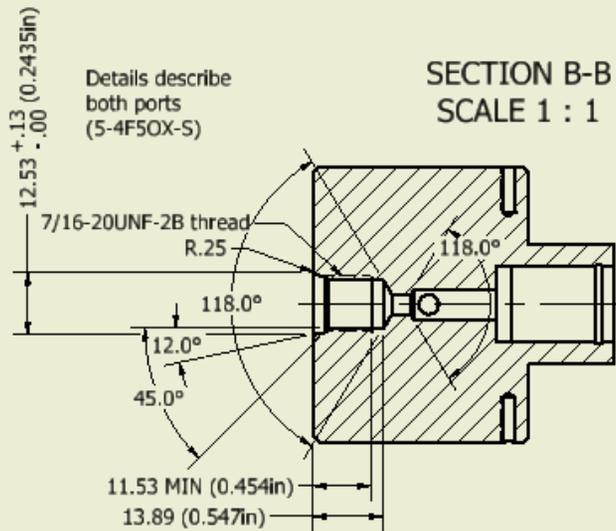
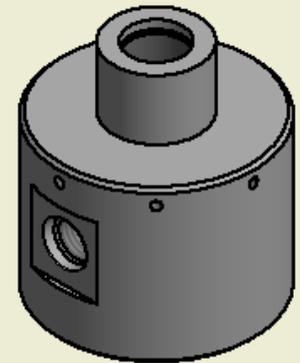
DRAWN Lucas Paix	18/04/2019	PART NUMBER: VN1	
CHECKED		TITLE	
QA		Needle	
MFG			
APPROVED			
		SIZE A3	DWG NO Valve Demo
			REV 2
		SHEET 1 OF 4	



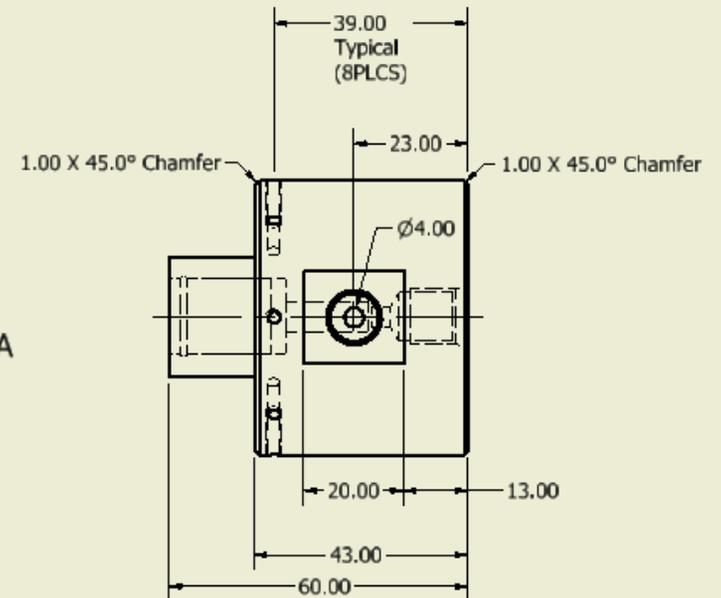
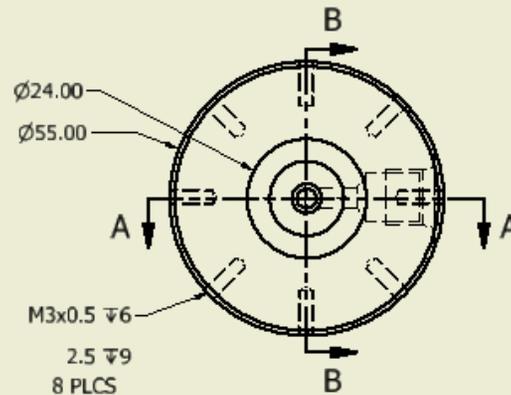
DETAIL A
SCALE 2 : 1



SECTION A-A
SCALE 1 : 1



SECTION B-B
SCALE 1 : 1



Mating components:
-VN1: M15x1 thread

Material: Aluminium 6061 (T6)
Finish: 1.6 RA unless otherwise specified
Tolerances unless otherwise stated:
Linear: ±0.1mm
angular: ±0.5°
all unspecified
internal radii : 0.4mm

DRAWN Lucas Paix		18/04/2019		PART NUMBER: VB1	
CHECKED				TITLE	
QA				Valve Block	
MFG					
APPROVED					
		SIZE	DWG NO	REV	
		A3	Valve Demo	3	
				SHEET 2 OF 4	

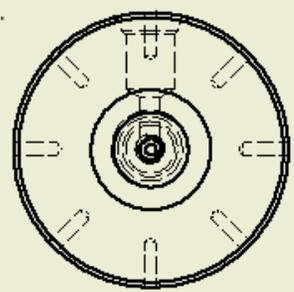
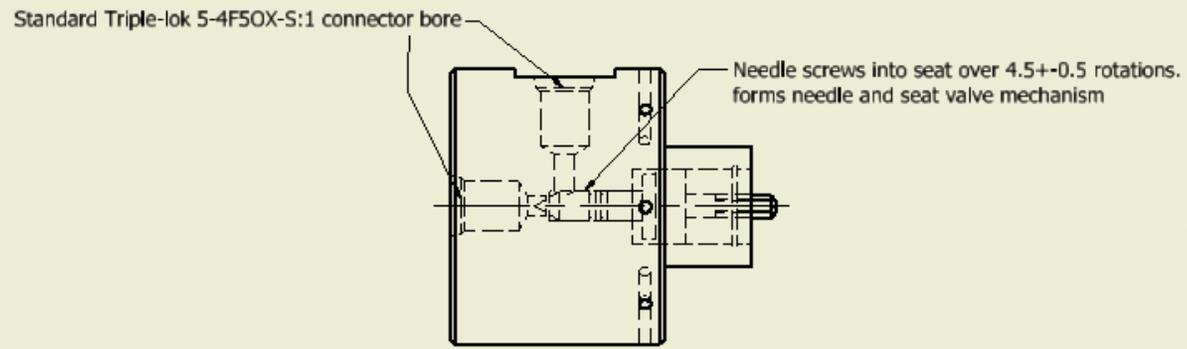
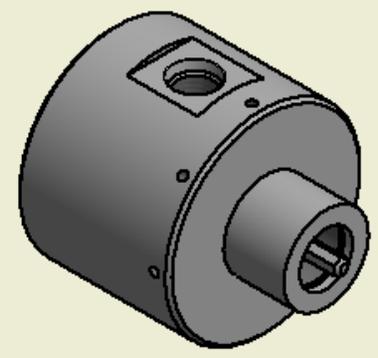
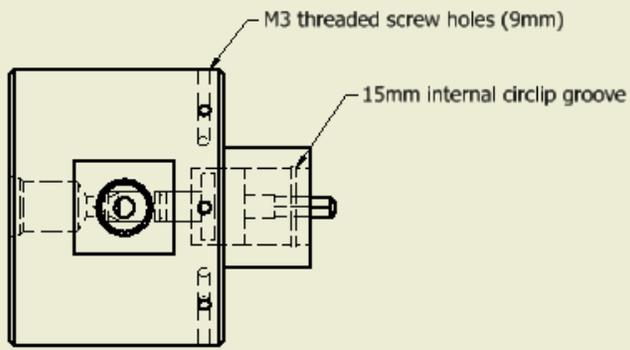
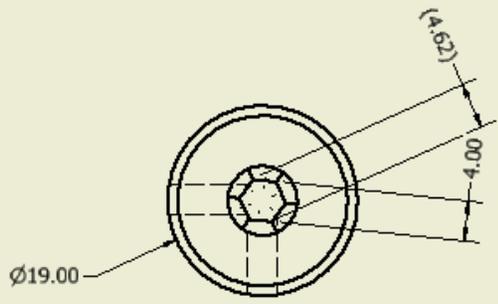
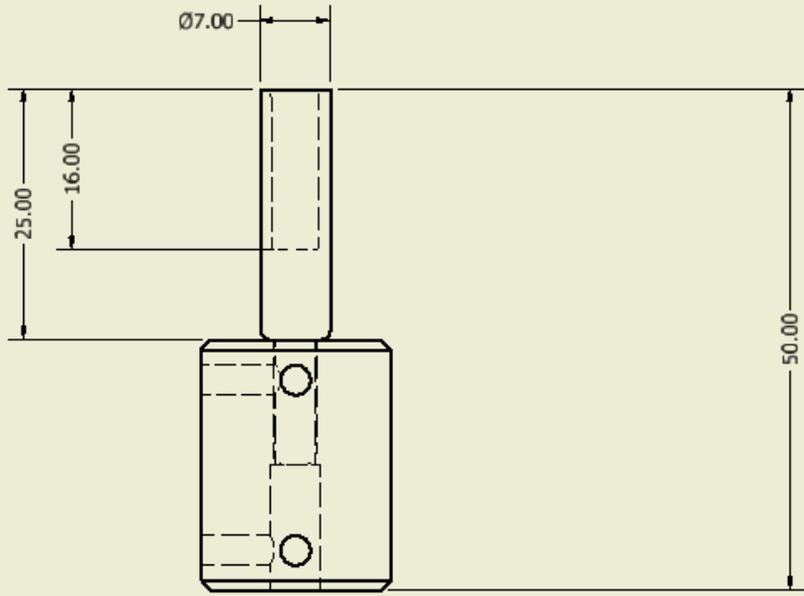
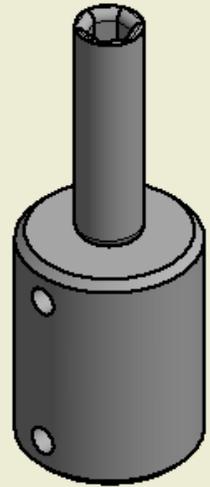


TABLE	
Material	QTY
Aluminium 6061	2

DRAWN Lucas Paix	18/04/2019		
CHECKED		TITLE	
QA		Valve assembly	
MFG			
APPROVED			
		SIZE A3	DWG NO Valve Demo
			REV
SHEET 3 OF 4			



Significant details are those describing the Hex socket



Mating components:
 -VN1: 4mm (4.62mm) Hex socket to allow rotational coupling but linear sliding

This part (SC1) has already been created, drawing provided for reference for mating with part VN1

DRAFT DO NOT MANUFACTURE

DRAWN Lucas Paix	18/04/2019	PART NUMBER: SC1		
CHECKED		TITLE		
QA		Shaft Coupler		
MFG				
APPROVED				
		SIZE A3	DWG NO Valve Demo	REV
		SHEET 4 OF 4		

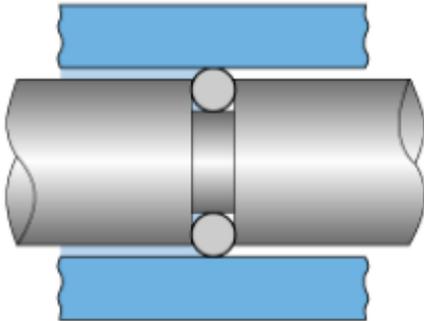
7.2 Valve Seal

ERIKS

know-how makes the difference

| o-ring.info

| o-ring design



Piston - groove dimensions				
	+	-		
material	Steel (Ck45 etc.)			
free groove volume at inst (%)	20			
bore diameter (mm)	6.00	0.018	0	H8
piston diameter (mm)	5.80	-0.01	-0.022	f7
groove diameter (mm)	4.40	0	-0.03	h9
groove width (mm)	1.32	0.2	0suggestion	
radius (mm)	0.2			
inner diameter stretch at inst (%)	0			

Application		O-Ring			
			±		
Sealing principle	Piston	compound	NBR		
design	Design Groove	chemical volume swell (%)	0		
temperature (°C)	21	inner diameter (mm)	4.5	0.15	ISO 3601
movement	static	cross section diameter (mm)	1	0.08	ISO 3601
pressure	over pressure				
compression (%)	20				

Results at Installation	min.	nom.	max.
Calculated Values at Central Position of Piston:			
O-Ring Compression (%)	8.93	20.00	25.93
Free Groove Volume (%)	2.79	20.00	45.38
O-Ring Inner Diameter Stretch (%)	- 6.02	- 2.22	1.15
Results at Service	min.	nom.	max.
Calculated Values at Central Position of Piston:			
O-Ring Compression (%)	8.93	20.00	25.93
Free Groove Volume (%)	2.79	20.00	45.38
O-Ring Inner Diameter Stretch (%)	- 6.02	- 2.22	1.15
Groove Depth incl. Gap (mm)	0.80	0.80	0.82
Sealing Gap (mm)	0.10	0.10	0.12
Calculated Values at Excentral Position of Piston:			
O-Ring Compression (%)	- 4.33		36.20
Groove Depth incl. Gap (mm)	0.69		0.94
Sealing Gap (mm)	0.00		0.24

Comments

Results at Installation

Caution, excessive groove fill may result in decreased part life, extrusion, or part degradation

Results at Service

Caution, excessive groove fill may result in decreased part life, extrusion, or part degradation

Disclaimer

This information is, to the best of our knowledge, accurate and reliable to the date indicated. The above mentioned data have been obtained by tests we consider as reliable. We don't assure that the same results can be obtained in other laboratories, using different conditions by the preparation and evaluation of the samples.

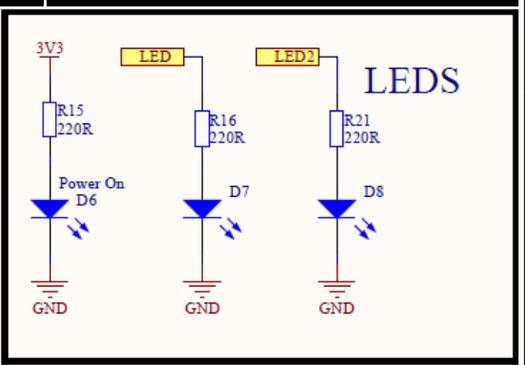
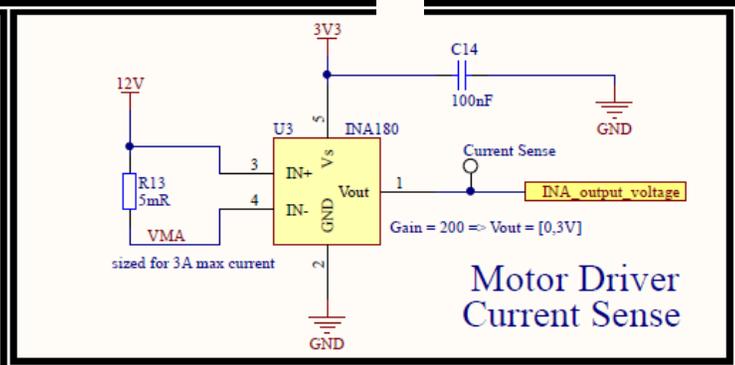
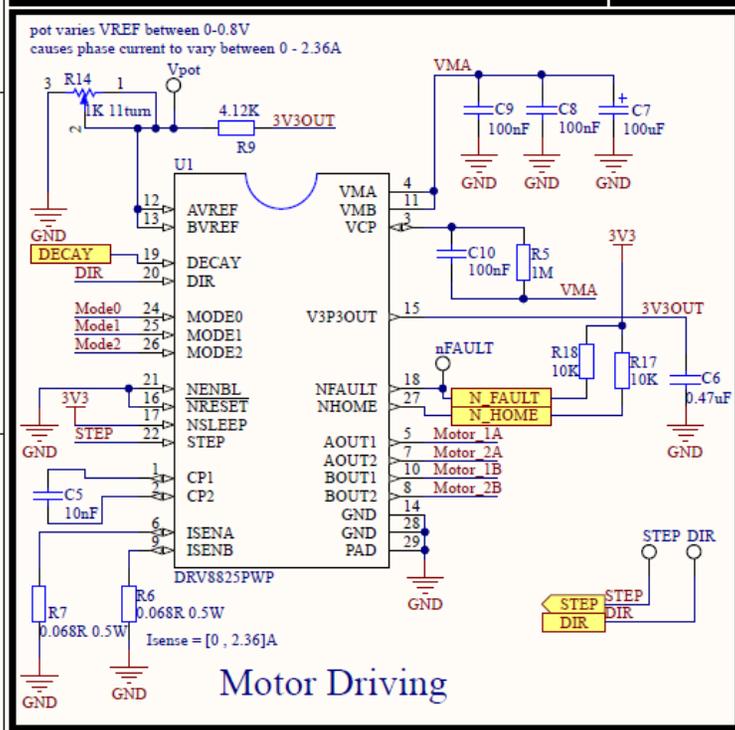
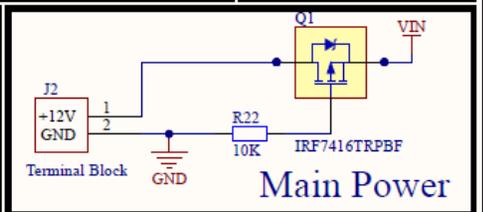
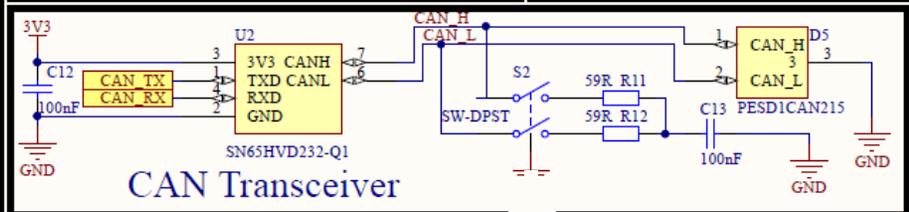
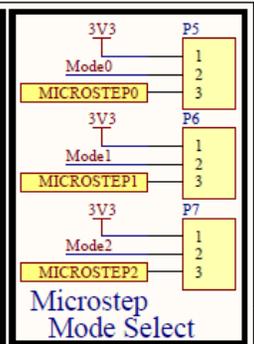
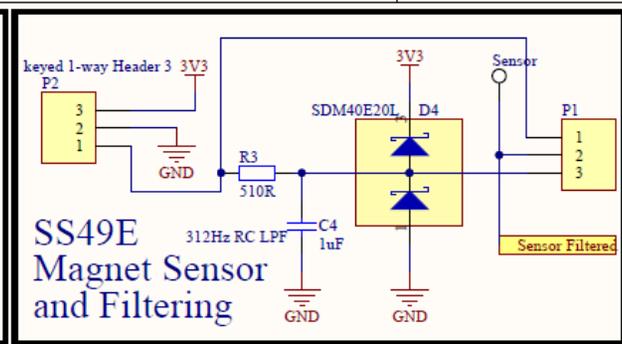
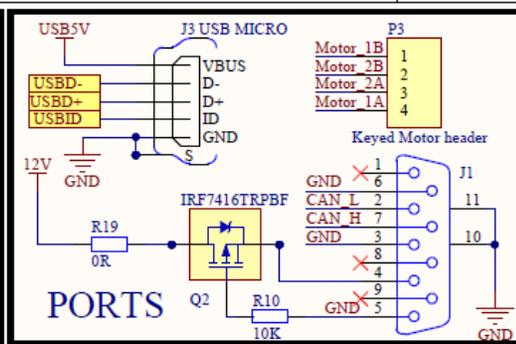
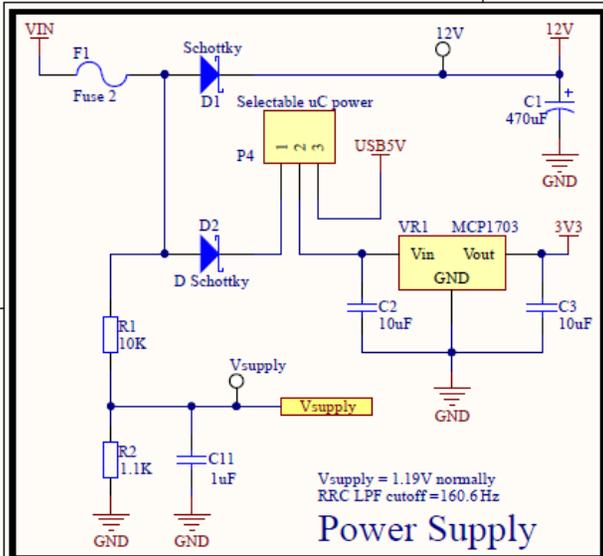
7.3 Valve Bill of Materials

Part	Part no.	Supplier	Qty	Type	Material	Size
Needle	VN1	Supashock	1		Aluminium 6061	see drawings
Block	VB1	Supashock	1		Aluminium 6061	see drawings
O-Ring seal		ERIKS	2	NBR		4.5mm ID, 1mm cross
circlip		ANY	1	internal		15mm nominal
ring magnet	R-OD13H2ID5-N35	frenergy.com.au	3	N35	Neodymium	OD13 x H2 x ID5 mm

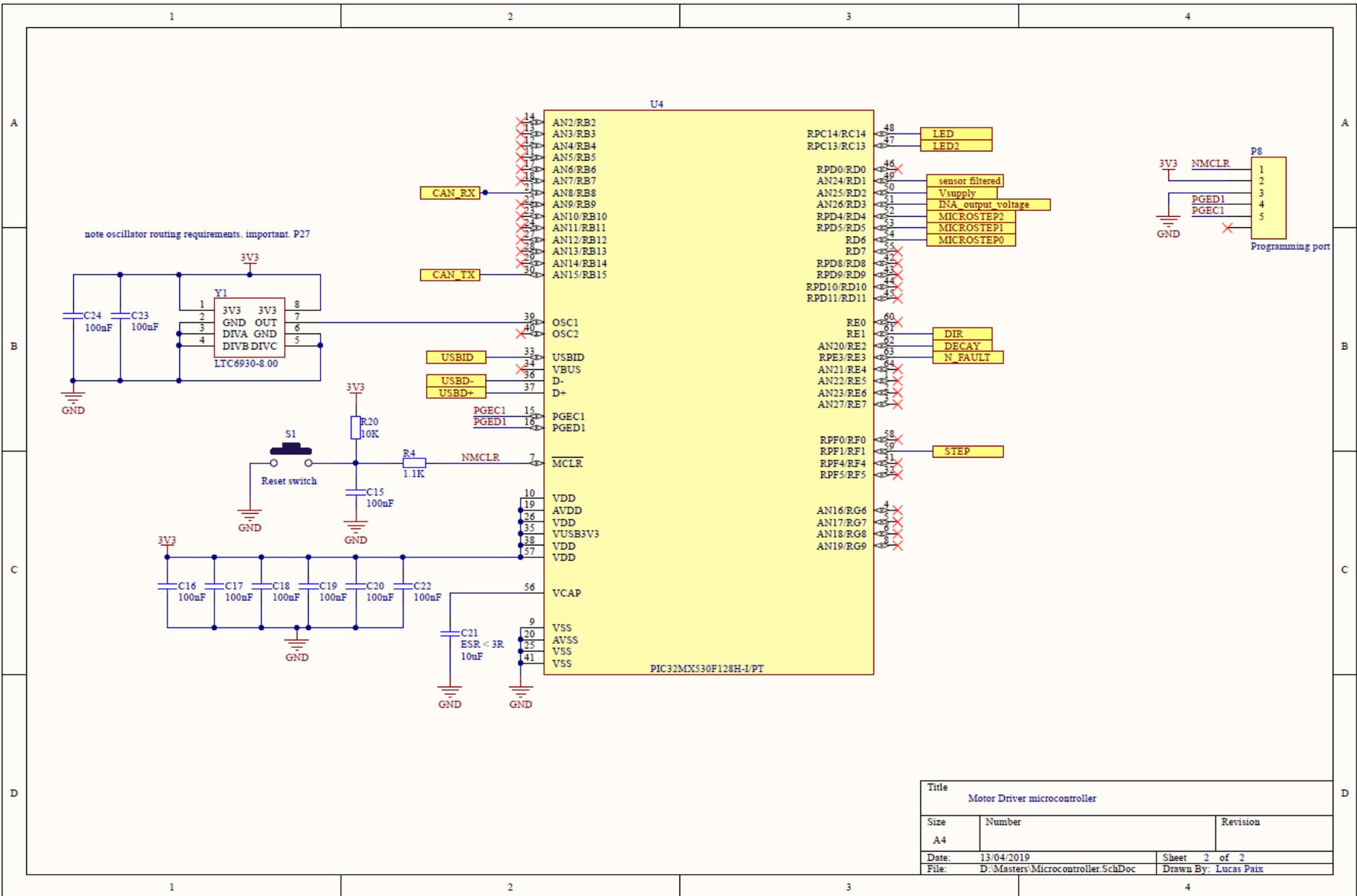
7.4 Test Rig Assembly Bill of Materials

Part	Part no.	Supplier	Qty	Type	Material	Size
Needle	VN1	Supashock	1		Aluminium 6061	see drawings
Block	VB1	Supashock	1		Aluminium 6061	see drawings
O-Ring seal		ERIKS	2	NBR		4.5mm ID, 1mm cross
circlip	ANY	ANY	1	internal		15mm nominal
ring magnet	R-OD13H2ID5- N35	frenergy.com.au	3	N35	Neodymium	OD13 x H2 x ID5 mm
Stepper motor	42BYGHM809	Element14/Digik ey	1	1.7A, 400 steps/rot	-	NEMA17
Custom Hex socket			1			
Flexible Shaft Coupler	ANY	ANY	1	Flexible		4mm – 5mm
Mount shell			1	3D printed	PLA	
Sensor Mount Bracket			1	3D printed	PLA	
Hall Effect Sensor	SS49E	Element14/Digik ey	1	Linear ratiometric		Flat TO-92
Mount screws	ANY	ANY	12			3mm Dia x 15mm L

7.5 PCB Schematics



Title			Motor driver peripherals		
Size	A4	Number			Revision
Date:	13/04/2019	Sheet 1			of 2
File:	D:\Masters\mainboard1.SchDoc		Drawn By:	Lucas Paix	



Title		
Motor Driver microcontroller		
Size	Number	Revision
A4		
Date:	13/04/2019	Sheet 2 of 2
File:	D:\Masters\Microcontroller.SchDoc	Drawn By: Lucas Paix

7.6 PCB Bill of Materials

PCB Bill of Materials (Per PCB)

Manufacturer pt. no	digikey pt. no	Value	Description	Designator	Footprint	Qty	Price ea. \$	Price total \$
UWT1C471MNL1GS	493-2181-1-ND	470uF	Polarized Capacitor (Radial)	C1	8.3 x 8.3	1	0.78	0.78
EEE-1CA101WP	PCE3880CT-ND	100uF	Polarized Capacitor (Radial)	C7	6.6 x 6.6	1	0.63	0.63
CL21A106KOQNNNG	1276-6455-1-ND	10uF, 16V	Capacitor SMD ceramic	C2, C3, C21	C0805	5	0.25	1.25
GCM219R71C105KA37D	490-8043-1-ND	1uF, 16V	Capacitor SMD ceramic	C4, C11	C0805	2	0.32	0.64
C0805C104K4RACTU	399-1167-1-ND	0.1uF, 16V	Capacitor SMD ceramic	C8, C9, C10, C12, C13, C14, C15, C16, C17, C18, C19, C20, C22, C23, C24	C0805	16	0.32	5.12
C0805C103K4RACTU	399-7995-1-ND	10nF, 16V	Capacitor SMD ceramic	C5	C0805	2	0.25	0.5
0805YC474KAT2A	478-1403-1-ND	0.47uF, 16V	Capacitor SMD ceramic	C6	C0805	1	0.47	0.47
PMEG3050BEP,115	1727-5207-1-ND	30V, 5A	Schottky Diode	D1, D2	4.4 x 3.4	2	0.65	1.3
SDM40E20LS-7-F	SDM40E20LS-FDICT-ND	20V, 400mA	Schottky Diode dual package	D4	SOT23_N	1	0.66	0.66
PESD1CAN215	1727-3817-1-ND	24V, 3A	CAN bus ESD protection diode	D5	SOT23_N	1	0.72	0.72
SML-D12U1WT86	SML-D12U1WT86CT-ND	2.2V	Red SMD led	D6	1608 (0603)	1	0.32	0.32
SML-D12Y1WT86	SML-D12Y1WT86CT-ND	2.2V	Yellow SMD LED	D7	1608 (0603)	1	0.32	0.32
SML-D12M8WT86	511-1578-1-ND	2.2V	Green SMD LED	D8	1608 (0603)	1	0.50	0.5

OZCG0260BF2B	507-1780-1-ND	5A trip, 2.6A hold, 16V	SMD fuse self- resetting	F1	1812 (4532)	1	0.41	0.41
A-DF 09 A/KG-T2S	AE10921-ND		Receptacle Assembly, 9 Position, Right Angle	J1	DSUB1. 385-2H9	1	1.13	1.13
OSTVN02A150	ED10561-ND		16A terminal block for power - OSTTA020161	J2	HDR1X3	1	1.33	1.33
10118193-0001LF	609-4616-1-ND		USB micro B socket	J3	USB MICRO SOCKE T	1	0.62	0.62
M22-2511005	952-1316-ND		Header pin male	P1, P4, P5, P6, P7	HDR1X3	2	0.89	1.78
5-103635-3	1777068 (Element14)	Keyed 1- way header	Keyed 1-way header PCB mount 4-pin	P3	HDR1X4	1	1.35	1.35
5-103634-2	2429542 (Element14)	Keyed 1- way header	Keyed 1-way header PCB mount 3-pin	P2	HDR1X3	1	1.61	1.61
1-103957-4	1126555 (element14)	Keyed 1- way header	Keyed 1-way header Plug 4-pin			1	2.12	2.12
104257-2	1777052 (Element14)	Keyed 1- way header	Keyed 1-way header Plug 3-pin			1	0.78	0.783
PPTC061LFBN-RC	S7004-ND		6-way header socket	P8	HDR1X6	1	0.76	0.76
IRF7416TRPBF	IRF7416TRPBFCT- ND	10A 30V	P-MOSFET	Q1, Q2	8-SOIC	2	1.53	3.06
RNCF0805DTE10K0	RNCF0805DTE10 K0CT-ND	10K, 0.125W	SMD resistor	R1, R17, R18, R22, R23	0805	7	0.25	1.75

RC0805FR-071K1L	311-1.10KCRCT-ND	1.1K, 0.125W	SMD resistor	R2, R4	0805	3	0.15	0.45
ESR10EZPJ511	RHM510KCT-ND	510R, 0.4W	SMD resistor	R3	0805	1	0.15	0.15
ERA-6AEB105V	P1MDACT-ND	1M, 0.125W	SMD resistor	R5	0805	1	0.53	0.53
RLP73N2BR068JTD	(element14) 2116423	0.068R, 0.5W	SMD current sense resistor	R6, R7	0805	2	0.62	1.24
RC0805FR-074K12L	311-4.12KCRCT-ND	4.12K, 0.125W	SMD resistor	R9	0805	1	0.15	0.15
RC0805FR-0759RL	311-59.0CRCT-ND	59R, 0.125W	SMD resistor	R11, R12	0805	3	0.15	0.45
ERJ-6LWFR005V	P19183CT-ND	0.005R, 0.5W	SMD current sense resistor	R13	0805	1	0.92	0.92
PVG5A102C03R00	490-2660-1-ND	1K 11turn 0.25W	Square Trimming Potentiometer 11-turn	R14	trimpot	1	4.68	4.68
CRGCQ0805F220R	A129741CT-ND	220R 0.125W	SMD resistor	R15, R16, R21	0805	3	0.15	0.45
RMCF0805ZT0R00	RMCF0805ZT0R00CT-ND	0R 0.125W	SMD resistor	R19	0805	1	0.15	0.15
PTS645SK43SMTR92 LFS	CKN9084CT-ND		Micro tactile reset switch	S1	MICROSWITCH	1	0.32	0.32
A6S-2102-H	(element14) 1455487	24V 25mA	Dual Dip switch SMD	S2	dip switches	1	2.17	2.17
RCU-0C	A106145CT-ND		Test Point, 1 Position SMD	TP1, TP2, TP3, TP4, TP5, TP6, TP7, TP8	1.6x0.8 mm pad	8	0.32	2.56
DRV8825PWPR	296-29503-1-ND	2.5A	2.5A Bipolar Stepper Motor Driver with On-Chip 1/32 Microstepping	U1	SOP65p 640x120	1	5.63	5.63
SN65HVD232DR	296-26344-1-ND	3.3V	3.3V CAN transceiver	U2	can transceiver	1	2.95	2.95

INA180A4IDBVR	296-47655-1-ND		Low- and high-side voltage output, current-sense amplifiers	U3	sot-23-6	1	0.88	0.88
PIC32MX530F128H-I/PT	PIC32MX530F128H-I/PT-ND		32-bit Microcontrollers (up to 512 KB Flash and 64 KB SRAM) with Audio/Graphics/Touch (HMI), CAN, USB, and Advanced Analog	U4	pic32MX520F128H footprint	1	5.21	5.21
MCP1703T-3302E/CBCT-ND	MCP1703T-3302E/CB	3.3V, 250mA	MCP1703 3.3V voltage regulator	VR1	SOT-23	1	0.82	0.82
LTC6930HMS8-8.00#PBF	LTC6930HMS8-8.00#PBF-ND	8MHz	Precision μ Power Oscillators	Y1		1	6.44	6.44
ROB-10846	1568-1106-ND	1.7A, 400 step/rot	bipolar hybrid stepper motor	-		1	24.86	24.86
SS49E	480-2006-ND		Sensor hall effect analog radial lead	-		1	2.59	2.59
							Total price:	\$ 93.51

7.7 Arduino Test Codes

```
//-----Read SS49E Hall Effect sensor Analog Voltage on Pin A0-----  
//this simply reads the analog voltage on pin A0 and displays it to the serial  
terminal  
const int sensorPin = A0; // analog sensor pin  
  
void setup() {  
  int sensorValue; // value read from the sensor  
  Serial.begin(9600); //initialise serial monitor  
}  
  
void loop() {  
  int sensorValue = analogRead(sensorPin); //read sensor value  
  Serial.println(sensorValue); //display to serial window  
  delay(50); //wait 100m  
}  
//----- END PROGRAM -----
```

```
//----- drive stepper motor test code -----  
//this program drives a stepper motor using pins 5 and 2 with step rate  
//determined by delays and delayus  
  
#define direction_pin 5  
#define step_pin 2  
const int steps_per_rot = 400; //constant for this motor  
void step(boolean dir, byte delayus, int steps);  
  
void setup() {  
  pinMode(8, OUTPUT); //set CNC shield enable pin as output  
  pinMode(step_pin, OUTPUT); //set step pin output  
  pinMode(5, OUTPUT); //set direction pin output  
  digitalWrite(8,HIGH); //enable cnc shield module  
}  
void loop() {  
  bool dir = true; //which direction to rotate  
  int rotations = 5; //number of rotations  
  int steps = rotations * steps_per_rot; //number of steps to perform  
  int delays = 1; //delay time sets motor speed  
  int delayus = 1200; //300 seems to be the lowest stable  
value  
  
  //rotate motor for specified speed, direction and number of rotations  
  step(dir,delayus,steps);  
  delay(1);  
}  
//create basic square wave pattern from delay time  
void step(boolean dir, int wait, int steps)  
{  
  digitalWrite(direction_pin, dir); //set direction  
  for (int i = 0; i < steps; i++) {  
    //wait = wait * (5-i);  
    digitalWrite(step_pin, HIGH);  
    delayMicroseconds(wait);  
    digitalWrite(step_pin, LOW);  
    delayMicroseconds(wait);  
  }  
}  
  
//----- END PROGRAM -----
```

```

//----- read sensor and drive motor simultaneously-----
//this program drives a stepper motor with specified delay and reads
//an analog sensor on pin A0 with each motor step

const int sensorPin = A0; //A0
int sensorValue = 0; // value read from the sensor

#define direction_pin 5
#define step_pin 2

int read_sensor();
void step(boolean dir, byte delayus, int steps);

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);

  pinMode(8, OUTPUT); //set CNC shield enable pin as output
  pinMode(step_pin, OUTPUT); //set step pin output
  pinMode(5, OUTPUT); //set direction pin output
  digitalWrite(8,HIGH); //enable cnc shield
}

void loop() {
  bool dir = true; //which direction to rotate
  int rotations = 10;
  int steps = rotations * 400; //number of steps to perform
  int delaysms = 1; //
  int delayus = 1200; //300 seems to be the lowest stable value
  int sensorval = 0; //magnetic sensor value

  step(dir,delayus,steps); //perform stepping and reading

  sensorValue = analogRead(sensorPin); //just read sensor
  Serial.println(sensorValue);
}

void step(boolean dir, int wait, int steps)
{
  digitalWrite(direction_pin, dir);
  for (int i = 0; i < steps; i++) {
    //wait = wait * (5-i);
    digitalWrite(step_pin, HIGH);
    delayMicroseconds(wait);
    digitalWrite(step_pin, LOW);
    delayMicroseconds(wait);

    int value = analogRead(sensorPin); //read sensor
    Serial.println(value); //print sensor to terminal
  }
}

//----- END PROGRAM -----

```

7.8 PIC Device Code

```
//----- PCB test program -----
//----- LUCAS PAIX -----
//----- ENGR9700: Masters Thesis -----
//----- Flinders University -----
//----- 13/5/19 -----

#pragma config FNOSC = PRIPLL //FRCPLL for internal PRIPLL for external
#pragma config FSOSCEN = OFF
#pragma config IESO = ON
#pragma config POSCMOD = XT // OFF for internal
#pragma config OSCIOFNC = OFF
#pragma config FPBDIV = DIV_1
#pragma config FCKSM = CSDCMD
#pragma config WDTPS = PS1048576
#pragma config FWDTEN = OFF
#pragma config WINDIS = OFF
#pragma config FWDTWINSZ = WINSZ_25
#pragma config FPLLIDIV = DIV_2 //clock is now 4MHz
#pragma config FPLLMUL = MUL_24 //multiply up to 96MHz
#pragma config FPLLODIV = DIV_2 //divide down to 48MHz

#include <xc.h>
#include <stdlib.h>

#define dir LATEbits.LATE1
#define decay LATEbits.LATE2
#define step LATFbits.LATF1
#define driverFault PORTEbits.RE3
#define LED_RED LATCbits.LATC14
#define LED_YEL LATCbits.LATC13
#define clockwise 1
#define anticlockwise 0
#define SYSFREQ 4800000L //system clock at 48MHz

int microstep_size = 1;
const int STEPS_PER_ROTATION = 400;
unsigned int n = 0;

void wait_ms (int);
void wait_5us (int);
void step_motor(int, int, int);
void set_microstep_size(int);
int error_check(void);
void read_analog(void);
void delayus(int count);

void init(void)
{
    TRISCbits.TRISC14 = 0; //set RC14 output
    TRISCbits.TRISC13 = 0; //set RC13 output
    TRISEbits.TRISE1 = 0; // set DIR pin output
    TRISEbits.TRISE2 = 0; // set Decay pin output
    TRISFbits.TRISF1 = 0; // set step pin output
    TRISDbits.TRISD4 = 0; // set microstep pin 0 output
    TRISDbits.TRISD5 = 0; // set microstep pin 1 output
}
```

```

TRISDbits.TRISD6 = 0;           // set microstep pin 2 output
PORTD = 0;                     // set micro step size 1

OC1CON = 0x5;                  //OCC pwm mode
T1CON = 0;                     //TIMER 1 1/256 PRESCALE, INTERNAL, 16bit
T2CON = 0;                     //timer 2, 1/256 pre, internal, 16bit

LED_RED = 0;                   //red led off
LED_YEL = 0;                   //yel led off
step = 0;                      //set step pin low default
dir = clockwise;              //set direction clockwise default
decay = 1;                     //set default decay mode = on

//-----ADC initialisation-----
AD1CSSLbits.CSSL24 = 1;        //configure A2D to read AN24, AN25, AN26
AD1CSSLbits.CSSL25 = 1;        //
AD1CSSLbits.CSSL26 = 1;        //

AD1CON1 = 0b0000000011100110; //16bit integer, auto convert,
AD1CON2 = 0b0000010010001000; //scan all muxA, 16bit buffers. change to 8bit if
more saving time needed
AD1CON3 = 0b0000111100000001; //clock from PBCLK, 31TAD
AD1CON1bits.ON = 1;           // turn on the A2D
}

int errorcode =0;
unsigned int sensor = 0, current = 0, voltage = 0;

void main(void)
{
    int i=0;
    init();
    set_microstep_size(1);
    while(1)
    {
        wait_ms(3000);

        if(!error_check())      //error condition check

        {
            //cycle valve full speed 1/1 microstepping
            set_microstep_size(1);
            step_motor(1200,clockwise,1600);
            wait_ms(1000);
            step_motor(1200,anticlockwise,1600);
            wait_ms(1000);

            //cycle valve full speed 1/2 microstepping
            set_microstep_size(2);
            step_motor(1200,clockwise,1600);
            wait_ms(1000);
            step_motor(1200,anticlockwise,1600);
            wait_ms(1000);

            //cycle valve full speed 1/4 microstepping
            set_microstep_size(4);
            step_motor(1200,clockwise,1600);
            wait_ms(1000);
            step_motor(1200,anticlockwise,1600);

```

```

        wait_ms(1000);

        //cycle valve full speed 1/16 microstepping
        set_microstep_size(16);
        step_motor(1200,clockwise,1600);
        wait_ms(1000);
        step_motor(1200,anticlockwise,1600);
        wait_ms(1000);

        LED_YEL = 1;
        while(1);

        //wait_ms(1000);
    }
    else
    {
        LED_YEL = 0;
        LED_RED = 0;

        for(i = 0; i<3; i++)
        {
            LED_RED = 1;
            wait_ms(100);
            LED_RED = 0;
            wait_ms(100);

        }
        wait_ms(500);
    }
}

void wait_ms (int count)                //delay 1ms using timer1
{
    int i;
    T1CONbits.TCKPS1 = 0;                //set timer prescale to 1/1
    T1CONbits.TCKPS0 = 0;
    T1CONbits.ON = 1;                    //turn on timer
    for(i = 0; i < count; i++)
    {
        while(TMR1 < 47985);             //pulses for 1ms: 47985
        TMR1 = 0;
    }
    T1CONbits.ON = 0;                    //turn off timer
}

void wait_5us (int count)                //delay 100us using timer1
{
    int i;
    T2CONbits.ON = 1;                    //turn on timer 2
    for(i = 0; i < count; i++)
    {
        while(TMR2 < 111);               // pulses for 5us: 111
        TMR2 = 0;
    }
    T2CONbits.ON = 0;                    //turn off timer
}

```

```

//void delayus(int count)
//{
//    unsigned int i = _CP0_GET_COUNT();
//    n = (CORE_TIMER_FREQ / 1000000 * count);
//    while(_CP0_GET_COUNT() -i < n);
//}
////Experimental, not counting exactly 1us as expected

int error_check(void)
{
    if(!driverFault) errorcode = 1;    //if fault pin from motor drive is active, error
    //include more error checks here

    //LED_RED = errorcode;            //if device error, shine red light
    return errorcode;
}

void set_microstep_size(int change_size)
{
    //set microstep size. if outside bounds, bring within range.
    //if invalid number, choose 1/1 size
    //must have all headers set to "digt1" on PCB for full digital control.
    microstep_size = change_size;
    if(microstep_size < 1) microstep_size = 1;
    if(microstep_size > 32) microstep_size = 32;
    switch(microstep_size)
    {
        case 1:
            PORTD = 0b00000000;
            break;
        case 2:
            PORTD = 0b10000000;
            break;
        case 4:
            PORTD = 0b01000000;
            break;
        case 8:
            PORTD = 0b11000000;
            break;
        case 16:
            PORTD = 0b00100000;
            break;
        case 32:
            PORTD = 0b10100000;
            break;
        default:
            PORTD = 0b00000000;
    }
}

void step_motor(int steps, int direction, int pulses_per_sec)
{
    int delay = 202000 / pulses_per_sec;
    int i = 0;
    dir = direction;
    for(i = 0; i < steps * microstep_size; i++)

```

```

    {
        step = 1;
        wait_5us(delay);
        step = 0;
        wait_5us(delay);

        //read_analog();
//      if(current > 70 && current < 130)
//      {
//          LED_YEL = 1;
//      }
//      LED_YEL = 0;
}

void read_analog(void)
{
    while(!IFS0bits.AD1IF); //wait until all sampling complete
    AD1CON1bits.ASAM = 0; //disable sampling
    sensor = ADC1BUF0;
    current = ADC1BUF2;
    voltage = ADC1BUF1;
    if(sensor < 460) //valve fully opened
    {
        LED_YEL = 1;
        LED_RED = 0;

        //wait_ms(100);
    }
    else if(sensor > 590) //valve fully closed
    {
        LED_RED = 1;
        LED_YEL = 0;

        //wait_ms(100);
    }
    else
    {
        LED_YEL = 0;
        LED_RED = 0;
    }
    AD1CON1bits.ASAM = 1; //enable sampling again
    IFS0bits.AD1IF = 0; //clear interrupt flag created by sampling
completion
}

```