MECH 490 Capstone Project Final Report for a Camless Engine

Team 21

By

Alex Boehler	6258743
James Bracken	6251420
Julien Pagé	6319548
Maria Papadopoulos	9350845
Brandon Robinson	6292100
Ben Tétreault	6263054

A report
submitted in partial fulfillment
of the requirements of the MECH 490 Capstone Project
Concordia University
Supervisor: Dr. Henry Hong
25 March 2015

ABSTRACT

Midterm Report for a Camless Engine

Alex Boehler, James Bracken, Julien Pagé, Maria Papadopoulos, Brandon Robinson and Ben Tétreault

A camshaft in an internal combustion engine allows for only fixed timing of valve events. There is a need to reduce fuel consumption, increase power, and reduce emissions. The objective of this Concordia Mechanical Engineering Capstone project is to mimic the camshaft profile by implementing electrical actuators to a camless single cylinder four-stroke engine and control the intake and exhaust valve timings. It will then be possible to offset the camshaft lobe separation angle to any desired position and to simulate variable valve timing. The results will be directly compared to a camshaft's profile obtained from testing a 6.5 HP OHV gas engine. The valve timing for early or late, opening and closing valve operation will be controlled by a microcontroller.

To this date, the primary objective has been met and the system was able to mimic the valve timing events for the specified camshaft for various ranges of engine speeds and achieved a low error of 11% area open, with minimal system lateness due to engine speed compensation. The camless system has been mounted on an engine, run, tested; which achieved a maximum of eight seconds due to intermittent combustion. Modifications were made to the engine head to allow for a variable intake valve, which yielded promising results.

Keywords: variable valve timing (VVT), electrical actuators

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

1 INTRODUCTION

A camshaft on an internal combustion engine (ICE) allows for only fixed timing of valve events. The objective of this Concordia Mechanical Engineering Capstone project is to replicate the camshaft profile by implementing electrical actuators to a camless single cylinder four-stroke engine to control the intake and exhaust valve timings which will increase its efficiency and reduce emissions [1]. It will then be possible to offset the camshaft lobe separation angle to any desired position, to simulate variable valve timing with electrical actuators, and the results will be compared to the camshaft's profile obtained from engine valve simulation laboratory results for a 6.5 HP OHV gas engine.

1.1 Background

The camless engine operates the valves by solenoids and allows for variable valve events. The engine can therefore be operated at ideal valve timing for all engine speeds which allows it to run at a higher efficiency, increasing fuel economy and reducing hazardous emissions [1]. Swedish auto manufacture Koenigsegg and Cargine, designed a Free Valve system that can reduce fuel consumption by 25%, reduce harmful emissions by 50%, and increase power by 30% [2].

1.2 Software

The software and microcontroller are the driving components behind this Capstone project, it is responsible for all control of the actuators and engine performance. The microcontroller will serve as the interface between the operator and the hardware.

1.3 Hardware

Hardware is considered all electrical and mechanical equipment that will be purchased for the project. It is a necessary component of the project as not all components can be manufactured due to time and cost. The solenoids were purchased based on specifications suitable to the 6.5 HP OHV gas engine.

1.4 Modeling and Fabrication

1.4.1 Primary Objective

To mimic the valves' lift, open duration, and offset through a microcontroller that tracks and displays the angular position of the crankshaft and the valve lift from a linear sensor and magnet, the engine's head is to behave as its own system with a valve actuation system. Modifications to the head include a finned housing for the solenoids. Printed circuit boards for the linear sensor and solenoid driver were also added to the design and fastened to a plastic bracket. The rotary sensor's printed circuit board was fastened to a 3D printed mount for a small DC motor to simulate the engine's crankshaft position.

1.4.2 Secondary Objective

The project is implemented to a larger scale. A test stand was designed for the 6.5 HP OHV gas engine when performing tests and will enable one to move the full project around without having to disassemble it while retrofitting sensors and equipment to the engine. To reduce vibration and to mount the engine, a large steel section of C-channel is used as a base. The base plate also includes a manifold air pressure (MAP) sensor, a guard, an alternator assembly, and a welded steel plate on which the rotary sensor stand assembly is fastened. Moreover, the base plate is clamped to a cart for safety during testing.

2 PROBLEM IDENTITIFICATION AND INFORMATION GATHERING

2.1 Problem Description

Drastic measures are gradually taken to adopt environmentally friendly alternatives. Engines are used extensively in industry. This design project will demonstrate that emissions can be reduced by implementing electrical actuators to a camless single cylinder four-stroke engine to control the intake and exhaust valve timings. Trials will be conducted on a 6.5 HP OHV engine. Pressure and MAP sensors will enable one to compare output from an engine with a camshaft to a camless design.

2.2 Results from Prior Experimentation

A Swedish automotive company, Koenigsegg, retrofit a camless engine on a SAAB vehicle [3]. Koenigsegg used a pneumatic-electric valve designed by Cargine to actuate the intake and exhaust valves for variable valve actuation (VVA) [3]. The valves had electric solenoid to control the flow of the air, and the compressed air would act as the force to move the intake or exhaust valves.

The Electromagnetic Valve Actuator (EVA) system which uses an electromagnet actuator to move the valve system was also studied. Previous designs have been found with the U.S Army in 1998 using a Kohler engine [4]. The system used a Valve outside Rotating Armature Design (VORAD); where the valve was attached by a rotating rocker to spring assembly, and the rocker arm would be rotated by applying current to either an electromagnet on top or bottom of the rocker arm armature.

Lastly, EVA was explored by Girish Parvate-Patil at Concordia Univeristy. A single solenoid was used for valve operation to eliminate the rocker arm mechanism [1].

2.3 Environmental Impact

A case study performed by Xing Lan and Yu Liu was conducted at the Chalmers University of Technology for the LC48VE Husqvarna lawn mower. It is stated that for a maintenance of 1000 m^2 lawn in Sweden during 10 years, "the maximum working hours for LC48VE is 250 hours while the grass growth period in Sweden is 5 months" [5]. In addition, according to the American Chemical Society, "the air pollution from cutting grass for an hour with a gasoline-powered lawn mower is about the same as that from a 100 mile automobile ride" [6] from a study conducted in 2001. It is also noted that "according to an EPA study prior to the Clean Air Act of 1990, small engines from lawn and garden equipment make up nearly 9 percent of some types of air pollution" [6]. The working hours for LC48VE per month is of 1.25. Figure 1 displays the input and output products of LC48VE over its life cycle per functional unit [5].

Table 1 Input and output of LC48VE's use during the whole life cycle per functional unit [20]

INPUT		**Unit**
Petrol	219.75	l
Engine oil	4.2	l
Blade	2	pieces
OUTPUT		
CO_2	173.08	kg
HC	3.07	kg
NO_X	0.77	kg

For the Capstone design project, if emissions are decreased by 10% when using actuators, by proof of concept, harmful emissions displayed in Figure 1 will decrease.

3 IDEA GENERATION AND SELECTION

3.1 Critique of Alternative Solutions

3.1.1 Hardware

3.1.1.1 Pneumatic Design

The pneumatic design uses compressed air to move the actuator position, it requires the use of an air pump, a tank (for compressed air), a minimum of two solenoids valves and actuators. By taking air at atmospheric pressure, compressing it in the pump and storing it in the tank, the valves can control the actuator. This design is proven to work in the private sector of engine development technologies. However, considering the limits of the project, the constraints of time, money and resources since the compensation for compressed gas would need to be evaluated in actuation timing, it was decided that it could not be done.

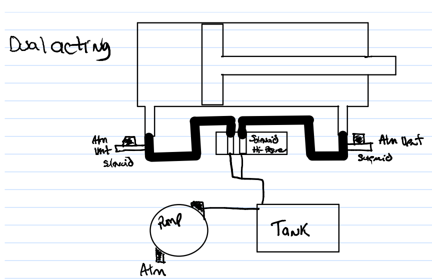


Figure 1: Pneumatic Valve

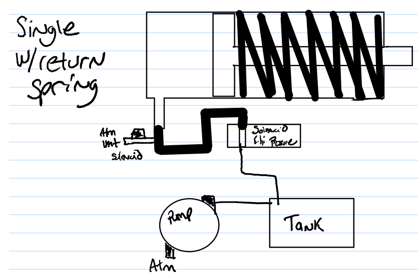


Figure 2: Pneumatic Valve Single Action

3.1.1.2 Hydraulic Design

The second design considered was to use hydraulic valve instead, as they are similar to pneumatic, but the fluid is much less compressible and would therefore require less compensation for compressibility.

The hydraulic design uses high-pressure oil to move the actuator position, it requires the use of a pump, two tanks (one for high pressure and low pressure return), a minimum of two solenoids valves and actuators. The pump takes oil from the reservoir at atmospheric pressure and compresses it into the high pressure storage tank. Through electronically controlled solenoid valves, the pressurized oil is then forced into its respective actuator. With a higher level of complexity, cost, the risk of high-pressure fluids and weight, cleaning and handling processes oil this system was considered to be an inferior design for the project scope.

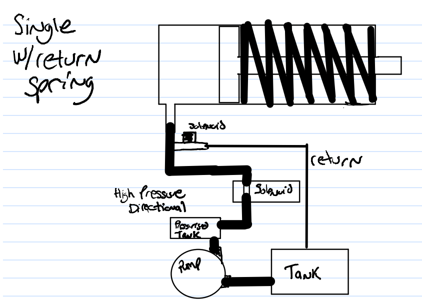


Figure 4: Hydraulic Valve Single Action

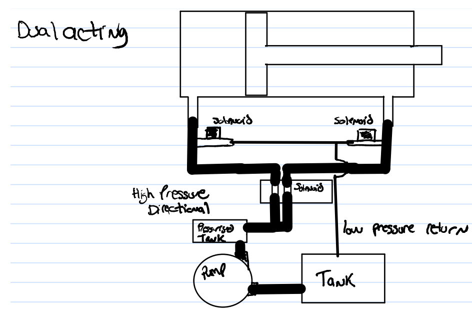
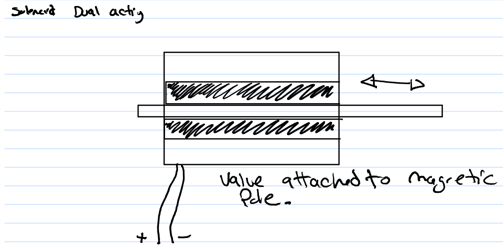


Figure 3: Hydraulic Valve Dual Action

3.1.2 Electrical Design

The electrical design uses voltage applied to solenoids to move the valves. Because solenoid prices can vary based on their size and performance it is important to ensure a proper project scope is used. Keeping lower costs, added mechanical simplicity (no pumps, lines or tanks) and easier integration, the electric solenoid design was chosen.



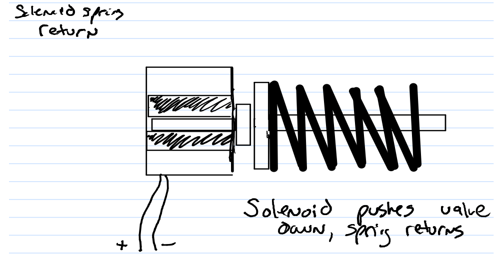


Figure 5: Electric Figure 6: Electrical Valve Single Action

3.1.2.1 Electromagnetic Valve Actuator (EVA)

EVA was explored by Girish Parvate-Patil at Concordia Univeristy for his Master's degree, was to use a single solenoid for the valve operation [1]. Parvate's Design simplifies the design of the system, as it will only require one actuator per valve, and does not require a rocker arm mechanisms.

3.1.3 **Two Solenoids Versus One**

When only using one electric solenoid, the return motion is dependent on the spring. If the spring fails or wears, the solenoid will not function at the correct frequency and engine performance will decline. In order to avoid this problem, the use of two solenoids per valve may be considered. A solenoid by definition can only pull in one direction. The two solenoids need to be positioned back to back in order to create a push pull motion. Since both solenoid rods will be connected together, when one solenoid is activated and the other is deactivated, the rod will go in one direction. If the opposite becomes true, the rod will go in the other direction. Although there is more control with this method, since the two solenoids are touching, a proper heat sink must be established in order to keep the solenoids at the correct operating temperature. This could be obtained either by heat sink fins or forced convection. It must be noted that the cost of using two solenoids per valve is doubled.

3.1.4 Position Sensors

Two sensors are crucial to this project. These include angular and linear position sensors. They will be used to detect the angular position of the crank shaft and the lift of the valves, respectively. The sensors will be soldered onto separate circuit boards which will be mounted onto a frame perpendicular to the engine's shaft, in the case of the rotary sensor, and parallel to the face of our solenoid mounting system for the linear sensor. The circuit boards must be insulated from the frame electrically to assure that the circuits do not short-out. In companies such as QPS Photronics, plastic spacers are used to assure that the board is insulated from its base.

Another aspect that must be considered is the isolation between the circuit board mounting brackets and the engine due to the vibrational forces of the latter. If vibrations are too great, it may cause the sensor to misalign from the shaft, creating misinformed readings. Isolating sensors from their targets is a common practice. QPS Photronics who specialize in vibration detection isolate their vibration sensors from other sources of vibration while calibrating them to assure the readings are valid. The first approach in reducing these vibrations will be to reduce the overall vibration of the system. This can be done by adding dampers to the system, as it will be mentioned in section 4.1.4, a large metal block will be used to hold the whole. Along with full system vibration reduction, individual sections can be reduced as well. One aspect will be to secure both ends of whatever holders being used. Leaving one end unsecured provides an increase in movement created by vibrational forces. Another addition that can be made is using rubber pads in areas where the sensors will be secured.

3.1.5 Software

The options of using the Atmega2560 microcontroller, Arduino Mega, and BeagleBone Black were explored.

3.1.5.1 Atmega2560 Microcontroller

The Atmega2560 standalone microcontroller would require a Printed Circuit Board (PCB), and many electrical parts to set up the microcontroller for initial testing.

3.1.5.2 Arduino Mega Microcontroller

The Arduino Mega uses the Atmega2560 and is already integrated with a PCB; it only requires power from a USB to work. This solution was suggested to us from Gilles Huard, technician at Concordia University, as it is simple to set up and get started quickly.

3.1.5.3 BeagleBone Black Microcomputer

BeagleBone Black is a microcomputer that runs a Linux operating system that has various analog and digital inputs and outputs, and runs at a clock speed of 1 GHz. The BeagleBone Black is very fast when compared to the other two options. However, it is significantly more expensive.

3.2 Innovative Procedures

To test the rotary sensor and solenoids, innovative procedures were taken.

3.2.1 Rotary Sensor

CAD modeling to create a piece that fit a circular magnet and sensor to a bread board was completed. The magnet was inserted onto a small electric motor to test the sensor and microcontroller. It was therefore possible to simulate the engine's crankshaft motion. The microcontroller enabled one to output the motor's speed. Its accuracy was then verified with a stroboscope.

3.2.2 Solenoid Mount

A solenoid holder was also 3D printed to perform tests. The design process underwent iterations to obtain a suitable holder that simulated the valve timing. The test was done using a spring to observe the solenoid's performance. Maximum frequency, speed and stroke length was also monitored. A solenoid driver was used to suit the microcontroller's specifications. With Gilles Huard's help, the testing procedures were done safely and effectively.

3.3 Unique Solution

To ultimately make the decision of which system to use, Dr. Hong was surveyed to choose the correct choice for the Capstone Design Project. Dr. Hong recommended the EVA system. The solution chosen involves replicating the valve lift by implementing electrical actuators to a camless single cylinder four-stroke engine to control the intake and exhaust valve timings. To validate the project's primary objective, electronically controlled actuators will be mounted onto an engine head. The valve position will then be mapped with a position sensor. The valve lift will be measured using a linear magnetic position sensor which will output the valve lift as a digital signal to the microcontroller where it will be recorded. The valve events will be timed by a stimulated crankshaft position sensor, where DC motor will be used to represent the crankshaft's revolutions per minute to 1200 rpm. The data will be tracked by a microcontroller and later plotted to see if the data matches the valve lift, open duration, and offset of the original engine camshaft

3.4 Process of Selection for Solutions

To determine the required software and hardware components to be used, constraints must be established.

3.4.1 General Project Constraints

- Budget: $750
- Manufacturing completion date: January 12, 2015
- Manufacturing completion date: February 9, 2015
- Curved shape of camshaft cannot be replicated

3.4.2 Hardware

Dr. Henry Hong recommended the solenoid type design by adopting the EVA to design a VVA system. The actuator must function at a certain frequency, load, stroke length, size and temperature. After meeting with Dr. Hong, he suggested using electric solenoids as opposed to other types because of their cost and simplicity. Through thorough research, it was agreed that the pre-existing spring used to push the cam shaft was to be replaced by a weaker spring. This will reduce the weight and size of the solenoid. To calculate the new spring constant, Hooke's law and Newton's equation where used in conjunction with the solenoid datasheets.

To power the solenoid at the correct frequency, Gilles Huard suggested using a simple MOSFET switching circuit which will receive a signal from the microcontroller. The heat needed to remove will vary depending on the duty cycle. To dissipate such heat the datasheet recommends an aluminum square plate of four inches with a thickness of eight inches.

To track the position of the crankshaft, a 14-bit angular magnetic rotary encoder will be used. The AS5048 integrated circuit (IC) has a 360 degree angle positioning with an accuracy of 0.05 degrees (after linearization and averaging). The IC measures the absolute position of a magnet's rotation angle and consists of Hall sensors, analog digital converter, and digital signal processing. It features simple programmable zero position through SPI or I2C command with no external programmer needed.

To properly communicate with the magnetic rotary encoder an appropriate magnet is required, following the manufactures specifications it is possible to source a Neodymium Iron Boron magnet (NdFeB) with a specific size of 6 mm diameter and thickness of 2.5 mm. It is important to note that not just any magnet can be used, as the magnetic field needs to be vertical to the IC surface.

3.4.2.1 Solenoids

Solenoids

- Minimum actuation frequency/Engine speed: 14 Hz/840 RPM
- Stroke length: 0.22"

- Max package diameter: 1.24"
- Estimated lifetime of 25 million cycles [7]

3.4.2.2 Throttle Body

- Minimum Bore: 0.74”

3.4.2.3 Sensors

- Pressure sensor: The sensor will tolerate a pressure of 300 psi
- Rotary sensor: The sensor will have a 360 degree angle positioning with an accuracy of 0.05 degrees
- Linear sensor: Resolution of $<0.5\mu m$

3.4.3 Software

Today’s engines generally use an engine control unit (ECU) which is generally a proprietary computer or microcontroller. To therefore design a VVA system for an engine, one would need to monitor sensors and output performance parameters to an engine. The most affordable method of designing an ECU is to use a microcontroller. Concordia University technician, Gilles Huard, suggested to use an Arduino microcontroller since it uses Atmel microchips, and is quicker to set-up. Researching the options between the Atmega2560 and the Arduino Mega, it was found that the Arduino Mega, uses the same Microcontroller (Atmega2560) and it already comes on a development board that can be powered by USB. Whereas simply suing the Atmega2560, a development board would have to be designed, extra parts would have to be ordered such as Pins, connectors, oscillators, resistors, capacitors, and power supplies. Therefore the conclusion was made that it would be cheaper to purchase the Arduino Mega as opposed to designing a circuit and implementing an Atmega2560.

4 DETAILED DESIGN

4.1 Synthesizing Solutions

4.1.1 Hardware

4.1.1.1 Solenoid Driver

Refer to drawing 11521402 in Appendix E for the solenoid driver circuit

Circuit design for the operation of the solenoids was created by both the past knowledge of several group members and the consultation of Gilles Huard. Due to the initial availability of components through the department the initial driver circuit was quite complex. The use of a transistor driving circuit was needed to isolate the signal from the Arduino because the MOSFET was to be driven at a 10V bias (as opposed to the 5V signal from the Arduino). This circuit was not needed due to the Arduino supplies enough current to power the solenoid driver.

Having past experience in circuit design it was possible to research an appropriate MOSFET that had a 5V or less bias needed to be fully saturated. With the new MOSFET, the circuit would no longer require a transistor driving circuit for either solenoid. This circuit was tested with the previous MOSFET however because of the 5V driving voltage it is impossible to get full switching power for the solenoid.

Exporting this design to DIPTRACE it is possible to create a Printed Circuit Board (PCB). Some difficulties arise from the component placement, routing and pad customization for external heat sinks. With this PCB design it is now possible to get the board made and extra components ordered.

There is a lot of forethought and coordination in circuit design as all boards and external circuits must have interconnectivity and functionality, in this case the solenoids are externally mounted, there is a separate power supply for the solenoids and the driving signal from the Arduino.

4.1.1.2 Printed Circuit Board (PCB)

See Appendix E for circuit schematics and Appendix F for all PCB diagrams

The sensors used on this setup all have to be placed in specific areas. By making the circuit boards custom, it allowed them to be placed where necessary. The electronics of this project are composed of 6 sensors, an LCD display, 2 solenoid drivers, the Arduino processor board and a shield board for the Arduino with custom connector setup.

The creation of a shield for the Arduino, is seen to be a necessity, without it, all the sensors would be attached by jumper wires, which not only is messy and hard to follow when troubleshooting, it is also dangerous, as the wires can easily by placed in the wrong spot, or come out if not put in properly. With the shield on, the wires are manageable, and more secure. However with this comes some forward thinking. Some of the sensors that will be used are only for the second objective, so the circuits have not yet been tested. This means that some extra connectors need to be placed so that it is ready for these new circuits, without having to design a whole new board.

A large issue with the circuit portion of the project is that the designs rely on some factors other than the electronics itself. The size of the board is a function of the item it is being mounted on, the position if the sensor on the board is also a function of this, so if a small change is made to the mechanical portion of the project, this could mean a total redesign of the board. The programming must also be taken into account as it must know which pin goes where. This is why the groups within the team communicate constantly to make sure that certain constraints are not overlooked.

The rotary sensor used in this project is an AS5048 Hall effect sensor. Its high resolution and low cost was the main reason it was chosen. The constraints for this board lie on the amount of pins needed from the Arduino, and that it has mounting holes, so it can be secured close to the output shaft of the motor. This chip uses Serial Peripheral Interface (SPI), which makes it much simpler to program with Arduino.

The linear senor chosen is very similar except that it does not have SPI. To overcome this problem, a counter/encoder with SPI was purchased. This chip will then count the pulses on the linear sensor, and the program will then only have to read the count in the counter, allowing the program to be much more efficient. The following diagram shows the circuit flow.

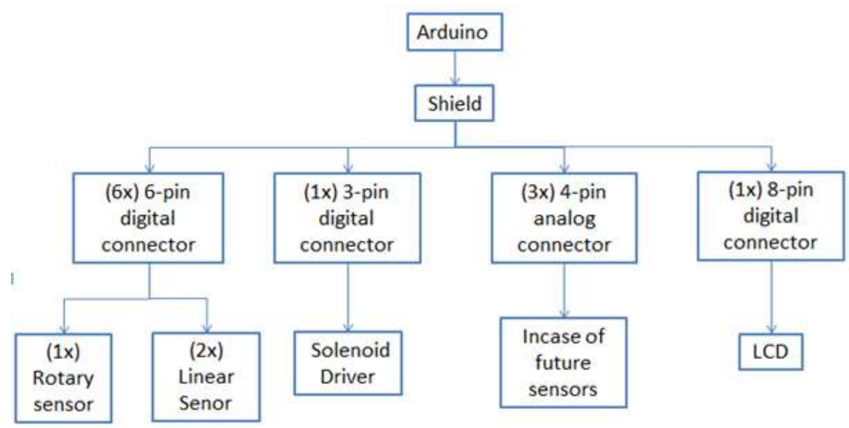


Figure 7: Connector Diagram

4.1.2 Software

Refer to Appendix K for software code

To determine the valve event, it was found to be very efficient to check the position of the crankshaft every iteration of code with and whether it is found in the pre-defined range where the valve is open. The valve will thus actuate open and remain open until it is out of the range. The next issue to overcome was to determine how to make the software interpret the results of the crankshaft position on a scale from 0 to 720 degrees instead of 0 to 360 degrees; as there are two rotations for every cycle. The sensor can only determine an absolute position of 360 degrees. To perform the operation, a change in crank position is detected. If it was found negative, it would change cycle from power stroke cycle to the intake and exhaust cycle, or vice versa, and continue counting until 720 at which point it will reset to 0 and loop over.

To validate the system, linear position sensors are used, software is designed to interpret the results from a counter and store the data on a microcontroller. The data is slowly read back on the serial port as not to disturb the system operation significantly during serial port communication.

For the secondary objective, the code was designed such that the input of the top right switch determines whether the system is in the start or run mode. In the start mode, the LCD screen writes the current crankshaft position as well as intake and exhaust valve lifts which were sensed by the system's linear sensors. This is to verify that the linear sensors are gathering correct data as well as ensuring that the code and the system's top dead center are in sync. Next, the run mode is mostly the code which was generated in the primary objective. The instantaneous and average rpm is written on the LCD screen. Additionally, intake and exhaust opening and closing times can be modified instantaneously through the use of the buttons located next to the LCD screen. These updates of valve timings are also indicated on the LCD so that the user is aware of the changes they have made.

4.1.3 Design and Manufacturing

Refer to Appendix B and C for all mechanical drawings

The original approach to the test cart was to allow it to have a single plate that would allow for the validation testing, running only the head, and for the engine to be mountable. However this proved to be the wrong approach. After a meeting with some of the lab technicians and Dominic Ng, it became apparent that much more was needed for the engine testing, whereas with the validation a simple set-up is needed. It was decided to split the testing set-up into a two parts.

4.1.3.1 Primary Objective

Refer to the cross-sectional view of the assembly in drawing 11521109 in Appendix B

4.1.3.1.1 Introduction

The first set-up for the validation will allow one to run the solenoids on the head and provide a view of the bottom so one can see the valves moving appropriately. The engine's head is to behave on its own with a valve actuation system. Modifications to the head include a finned housing for solenoids that are fastened to a plate mounted on the engine's head. Printed circuit boards for the linear sensor and solenoid driver were also added to the design and fastened to a plastic bracket. The rotary sensor's printed circuit board was fastened to a 3D printed mount for a small DC motor to simulate the engine's crankshaft position. With a microcontroller and computer displaying results for the motor's angular position which simulates the intended crankshaft position, one can be assured that the valves are doing as they are intended. The set-up is illustrated in Appendix A. To obtain a suitable solution to perform tests, many design iterations were performed for the solenoid mount's assembly. It is desired to obtain an assembly that will be simple to manufacture and cost-effective. This small test stand keeps everything together neatly so that trouble shooting is simpler to accomplish.

4.1.3.1.2 Solenoid Body and Threaded Plate

Refer to drawing 11521102 in Appendix B for the assembly and drawings 11521211 for the threaded plate in Appendix C

The 6.5 HP OHV engine was disassembled. A mount was modeled with respect to the engine head's dimensions. The solenoids purchased have a 0.75-16 UNF thread. They are fastened to a 1/2 inch plate to improve stability as they will be pushing the valves at various speeds. The plate was obtained from the EDML C.

4.1.3.1.3 Solenoid Tube Plate

Refer to drawing 11521118 for the solenoid tube plate assembly in Appendix B and drawings 11521208 and 11521209 for the left and right solenoid tubes, respectively, and drawing 11521210 for the slotted plate in which the tubes will be inserted in Appendix C

Two 1.5" outer diameter aluminum tubes with an inner diameter of 1" were obtained from the EDML C. It will be manufactured to fit design specifications. Fins will provide a necessary heat sink. The tubes are at a maximum package diameter of 1.24".

4.1.3.1.4 Solenoid Plunger and Magnet Cap

Refer to drawing 11521119 for the solenoid plunger and magnet cap assembly in Appendix C

A cap is glued to the top surface of the solenoid plunger. This cap is made of ABS due to its unique shape and extends to include a magnet strip for the linear sensor that determines the valves' positions at all times for any speed. The cap acts as a cantilever beam. Its thickness is therefore suitable to resist vibrations which would alter the output values from the sensor.

4.1.3.1.4.1 Printed Circuit Board Holder for the Linear Sensor and Solenoid Driver

Refer to drawing 11521403 in Appendix B

A bracket is added to the top plate which incorporates a circuit board that accommodates the linear sensors and solenoid driver. The bracket which is made of ABS and is fastened to the top surface of the solenoid finned tubes. The melting point of ABS and its non-magnetic properties were considered for all designs.

4.1.3.2 Secondary Objective

4.1.3.2.1 Introduction

From gathered data from the cam's profile and the piston's TDC position, one is able to determine the appropriate time for valve movement through push solenoids. The project's aim to mimic the valves' lift, open duration, and offset through a microcontroller that tracks and displays the angular position of the crankshaft and the valve lift from a linear sensor and magnet was set to a larger scale by implementing the project's first objective to an engine.

Components for the secondary objective were manufactured from mechanical drawings. Alterations were required on some components to fit the engine and to facilitate testing. The mechanical drawings are located in Appendix B and C. Moreover, modifications to OEM parts (LoveJoy coupler and a Delco-Remy alternator) were made.

Materials from the EDML were used to stay within the allotted budget under Michael Rembacz's supervision. *Refer to drawing 11521115 in Appendix A for the general assembly.*

4.1.3.2.2 Base Plate and Cart

Refer to drawing 11521115 for the secondary objective's general assembly in Appendix B and drawing 11521213 in Appendix C for the mount plate

To reduce vibration and to mount the engine, a large steel section of C-channel is used as a base. The base plate also includes a manifold air pressure (MAP) sensor, a guard, an alternator assembly, and a welded steel plate on which the rotary sensor stand assembly is fastened. The base plate is clamped to a cart for safety during testing.

4.1.3.2.3 Spacer

Refer to drawing 11521205 in Appendix C for the 5.5 mm spacer

An additional spacer was 3D printed to obtain a lift of 5.5 mm. This was to create a mechanical limit to the valve lift and to ensure there was no interference between the valves and piston. Because the spacer lowers the solenoid plunger within its case, it made it so the plunger bottoms out at its mechanical limit. Moreover, it provides a larger amount of force since it is located closer to the coils' core.

4.1.3.2.4 Alternator Assembly

Refer to drawing 11521105 and 11521106 for the alternator and bracket assembly, respectively, and drawing 11521116 for the coupler assembly in Appendix B

A steel stand was designed to support the alternator for these tests. The alternator is used to allow load based measurements of the engine’s performance. The alternator's pulley and cooling fan were removed and its rear bearing was modified to allow a customized fastener which contained the magnet that was used to determine the crankshaft's position. To connect the engine’s output shaft to the alternator a coupler set with a keyway was used. A coupler and the input shaft of the alternator was modified for fastening. For safety precautions, a guard was made to cover this portion of the assembly. The load was to be two 300 Watt resistors. Based on the resistance, either series or parallel, the appropriate connection would be selected. It was planned to measure current and voltage to calculate the power output of the engine at specific RPM’s. Due to complications with the coupler unfastening and a lack of time to readdress the issue, it was decided to remove the alternator and load banks from the design.

4.1.3.2.5 Rotary Sensor Bracket Assembly

Refer to drawing 11521111 in Appendix B for the assembly, and drawings 11521216 and 11521217 for the top and bottom components in Appendix C

Rotary sensor stand has two primary components: the base and upright. The base allows for movement in the Z- & X-axis while the upright enables displacement in the Y-direction. Since the stand only holds a circuit board, the printing process was deemed appropriate. The rotary sensor on the PCB was lined up with the magnet located on a customized bolt. The sensor and board need to be 0.5-3 mm away. Problems arose when it was determined that the alternator's coupler was unscrewing itself which resulted in an altered zero position, this caused for a significant source of error to the malfunctioning of the valve timing. The set-up was later simplified to a more direct path from the output shaft of the engine to the sensor directly.

4.1.3.2.6 Carburetor and Intake Assembly

Refer to drawing 11521214 and 11521219 for the MAP sensor part and servo throttle shaft in Appendix C

The engine used possesses a factory carburetor that has been modified to use an electric servo to control the throttle position. The air-fuel charge of the carburetor is based on intake vacuum created by the engine and respective throttle blade position. An intake spacer was made to mount a barb fitting in the intake before the intake valve, this effectively becomes a very small intake manifold for the project's purpose. The barb fitting was then connected to a MAP and is used to monitor intake manifold pressure which enables one to understand how the engine is performing.

4.1.3.2.7 Spark Plug and Pressure Sensor

The engine head was modified to accommodate a spark plug used from the thermodynamics laboratory that included a piezoelectric pressure sensor (PCB Model 112A05). The plug has previously been modified to include a pressure sensor that allows for the pressure in the combustion chamber to be determined mathematically and graphically, while testing, through an oscilloscope supplied by the laboratory technician Gilles Huard under Robert Oliver's supervision. Monitoring the pressure in the chamber is the most important piece of data that can be used for troubleshooting the operation of the engine. The oscilloscope also read data for the intake and exhaust valves, and the manifold air pressure. Wires were wrapped or fastened where possible to avoid tangling and misplacement.

4.1.4 Secondary Objective Testing Procedure

The original approach to the test cart was to allow it to have a single plate that would allow for the validation testing, running only the head, and for the engine to be mountable. However this proved to be the wrong approach. After a meeting with some of the lab technicians and Dominic Ng, it became apparent that much more was needed for the engine testing, whereas with the validation a simple set-up is needed. It was decided to split the testing set-up into a two parts.

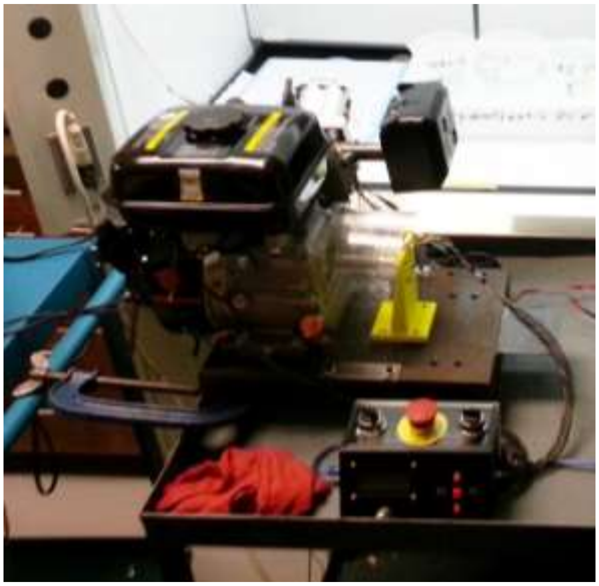


Figure 8: Secondary objective testing procedure

The first set-up for the validation will simply allow us to run the solenoids on the head and provide the team with a view of the bottom so that the valves could be monitored visually. With a computer and sensors connected to the solenoids, the user can be assured that the valves are doing as they are intended. The set-up is illustrated in Appendix B. This small test stand keeps everything together neatly so that trouble shooting is simpler to accomplish.

There are two main factors that make running the engine test somewhat difficult: the exhaust and vibrations. To run this engine, it must be placed either outdoors or a ventilated area. The stand that the engine is on must also be secured. The design of this test will be similar to that of the MECH 351 laboratory set-up. The motor will mount onto a heavy steel base which will then be secured to a cart. This base provides us with 2 crucial functions. One is the large mass acts as a mechanical vibrational damper, when the engine is run it will not vibrate as much while we are gathering data, making our values more precise due to a reduction in possible sources of error nor will the engine jerk as violently if our engine miss fires due to a miss calculation as well. The other function is that it provides us with a surface we can drill into to mount the other parts needed for the tests we need to run. This cart will then be rolled into a laboratory room with a ventilation hood where the fumes can be exhausted appropriately.

4.1.4.1.1 Arduino Box

Due to the great deal of electrical connections in this project, a central control console was created. This console houses the Arduino micro controller, an LCD screen to relay information in real time as well as an electronic throttle control and safety/mode switches. This box allowed the project to be cleaner and easier to maintain. The console acted as the central control unit while running tests, and allowed for a much more efficient execution of said tests.

The unit operates in 2 modes. Mode 1 is for “system check” this mode allows the user to fire the solenoids individually to check functionality. In this mode the LCD also provides readout of current crankshaft angle, as well as valve lift. With this mode active the user can also check the location of top dead center, verify valve lift, and solenoid functionality. With all these items checked and verified, the unit can be switched to mode 2.

In mode 2 the microcontroller is now in run mode. In this mode the code with all the valve timing information is engaged, and the engine is ready to be turned over. Before it can run though, the big red stop button must be disengaged and the switch to the left must be on. These two switches are safety measures implemented to assure safe testing. If the engine revs too high and becomes unstable the red button can be pressed, this will kill the spark to the engine as well as the 120 Volts being fed to the 12 Volt power supply. The switch on the left is to cut the 12 volts that is supplied the solenoids.

The box itself was designed with an incline front to make reading the LCD screen easier. It was 3D printed in two parts (walls and front panel) to allow for quicker and less wasteful print. The fact that it was 3D printed also allowed for easy customizability in the positioning of items. The top and rear portion of the box are laser cut Plexiglas. This allows one to see inside the box, which can sometimes aid in trouble shooting cases. Another aspect to aid in trouble shooting is the fact that the rear is held on by magnets which makes for a quick opening of the box, if an adjustment is needed.

4.2 Substantiating Through Testing

Refer to Appendix G for test status and procedures

4.2.1 Hardware

The original spring located in the engine head had an approximate spring stiffness of 30lb/inch. In order to keep the cam rotating at a certain speed this spring stiffness was needed. Initial research was made to find a solenoid which was powerful enough to push the spring while keeping a low profile. Unfortunately the cost for such an item was quite high due to the needed force. In order to fix this issue it was decided to remove the original spring and replace it with a much smaller one. Once a tubular solenoid was found, the spring constant for the new spring was calculated using kinematic equations. A small amount of preload was also taken in consideration to keep the spring in constant compression which found the ideal spring rate to be 1.76 lb/inch.

In order to test the tubular solenoid speed with the new spring a small test stand needed to be fabricated. The least expensive material solution was plastic. The design needed was not standard so an alternative fabrication method was required. A 3D model was designed and modeled to fit specifications.

Figure 9: Solenoid holder design for testing

It was then 3D printed using the team’s personal resources. There were flaws which lead to the first design ultimately cracking. A small magnetic strip also needed to be attached to the moving rod to observe stroke length. As seen below a second model was created with modifications and improvements using Concordia University's printer. After being tested, this new model proved to be well constructed and performed as expected. The magnetic strip will be attached to a spring holder which connects the spring to the solenoid.

4.2.2 Software

Timing is the greatest concern when it comes to programming. Every action will have an effect on the system's performance. The timing issue was first realized when the rotary angle sensor that was used for the crank shaft position was only reading a position every 70 ms, or less than one read for every rotation. By changing Serial port communication speed, the loop iteration speed was decreased to 10 ms. Finally as this was still not a realistic loop iteration speed, a method was developed to store the data and stop displaying information every iteration, as the display was taking the longest part of the cycle which brought the iteration loop time below 1 ms. It was timed by a digital pulse for every iteration and measured with an oscilloscope to ensure accuracy.

It is important to know rpm because the valve timings are not only in relation to the current crankshaft position but also in relation to rpm. In other words, since there is a delay between sending a signal to activate the solenoids and the actual actuation of the solenoids, an offset in relation to rpm for signaling valve timing is to be incorporated into the code. To know the amount of offset required, gathering an abundance of data is required. Since serial printing requires too much processing time, a dynamic array was created to store a larger amount of the following data for greater precision: time, crankshaft position, commanded intake valve position, commanded exhaust valve position, actual intake valve position, and actual intake valve. Note that the actual valve positions were measured with the linear sensor of our system. Through analysis of valve timing and offset factor was determined based on RPM. It is required to be based around RPM because all system timing is done in reference to crankshaft degrees; therefore the offset gives an offset in degrees (equation 1), and advances the turn on or off position of the solenoid.

$$offset = \frac{(INSTANT_{RPM} * 360.0 * 16.0)}{60000.0;} \qquad (1)$$

$with\ RPM\ in\ deg/ms$

4.2.3 Secondary Objective

In order to fulfill the secondary objective, Lotus Engine Simulation has been used to compare the simulated engine performance of the camshaft profile to the camless profile. Simulations are required to observe and understand the system before experimental tests are to be done. The specifications of the engine used in this design project were incorporated into the simulation. The original valve angles which were obtained experimentally were also incorporated into the simulation yielding following pressure-volume diagram.

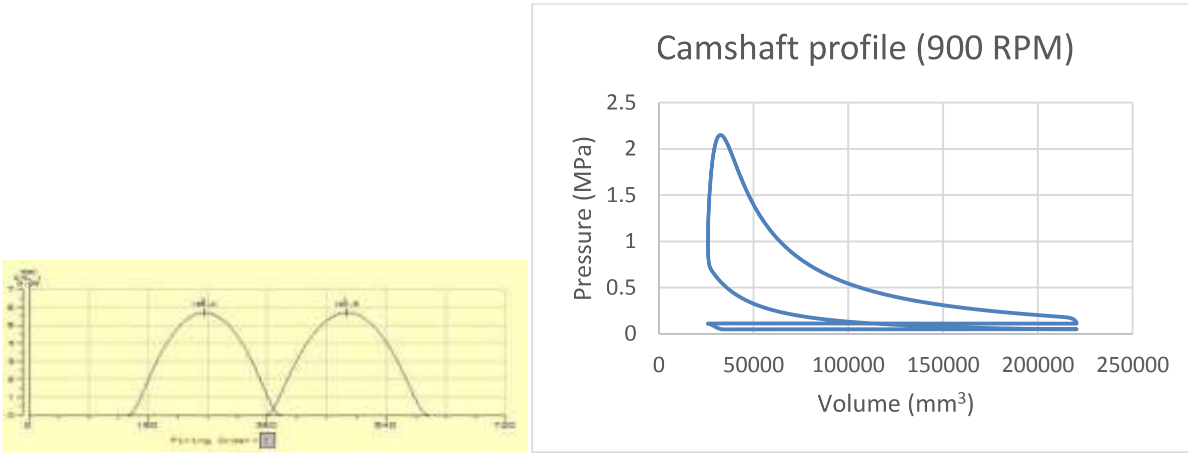


Figure 10: Camshaft Engine a) Exhaust and intake valve profiles for angles EVO = 31, EVC = 21, IVO = 3, IVC = 64, b) Pressure-Volume diagram

Using the same valve timings are the camshaft simulation and changing the intake and exhaust valve profiles to illustrate the solenoid behavior of the camless engine yields the following simulation.

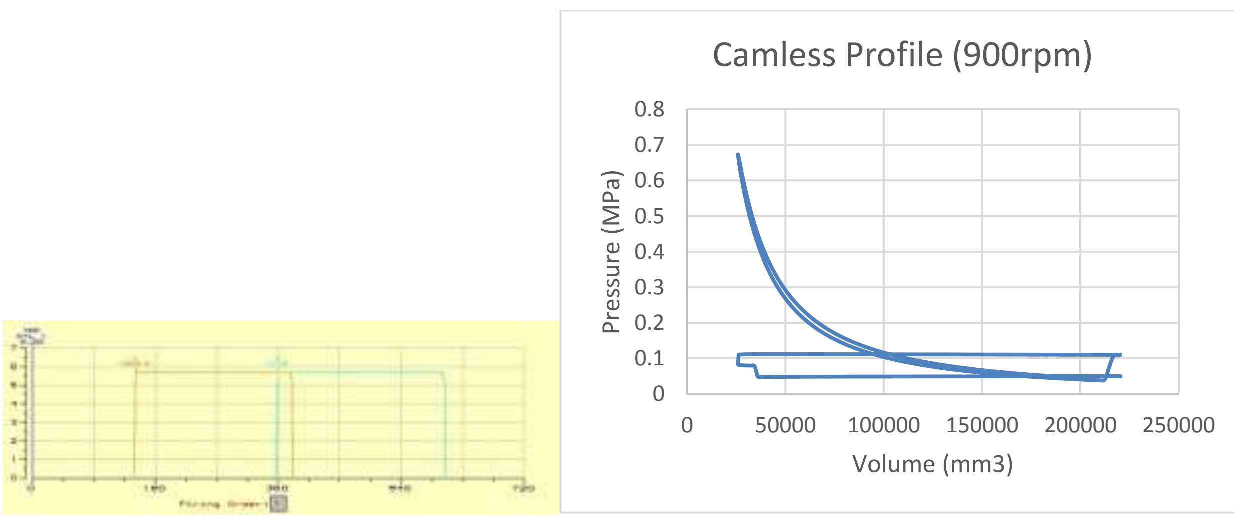


Figure 11: Camless Engine a) Exhaust and intake valve profiles of camless engine for angles EVO = 31, EVC = 21, IVO = 3, IVC = 64, b) Pressure-Volume diagram

This pressure-volume diagram is noticeably worse when compared to that of the original camshaft. The reasons for this include a larger amount of overlap between the intake and exhaust valve timings. Research suggests that less overlap is required for low RPMs in order to “increase vacuum, decrease fuel seepage into the exhaust, and less reversion of exhaust gases into the intake port” [27]. To reduce the overlay, it is favorable to reduce the exhaust valve closing time as oppose to change the intake valve opening time. Additionally, the intake valve closing time is to be manipulated to increase the maximum torque [30]. By making alternations to the valve timings as so: EVO = 31, EVC = 5, IVO = 0, IVC = 40; the following figure demonstrates a simulated camless engine with a similar performance to that of the original camshaft engine performance.

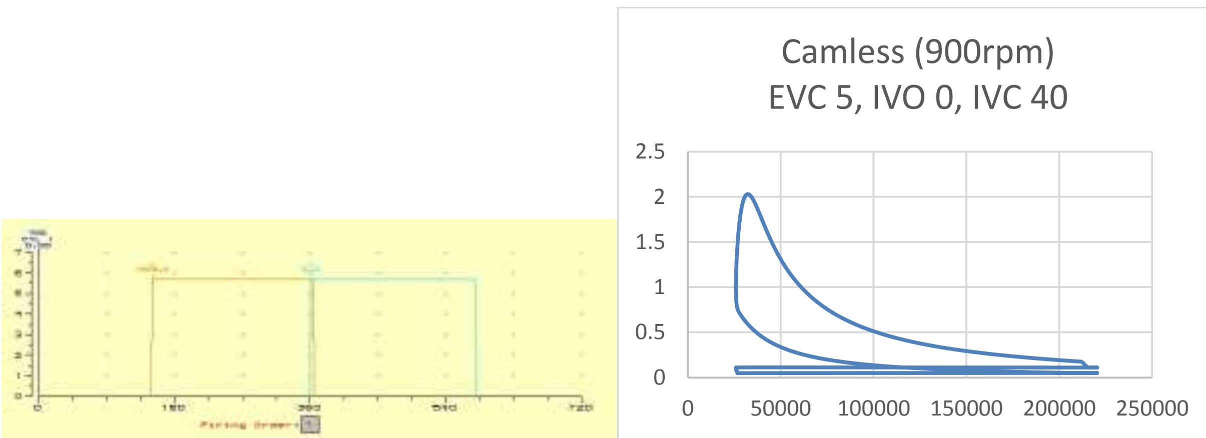


Figure 12: Camless Engine a) Exhaust and intake valve profiles of camless engine for angles EVO = 31, EVC = 5, IVO = 0, IVC = 40, b) Pressure-Volume diagram

4.2.4 Solenoid Actuated Intake Valve (SAIV)

To gain valuable knowledge of solenoid actuation, and deliver a working prototype of a solenoid actuation system for an internal combustion engine, it was decided a small change in system operation would be needed. Therefore the valve train was changed to allow the camshaft to continue working on the exhaust valve, and the intake valve would be solenoid operated.

The theory behind the SAIV system is that under all situations tested the exhaust valve always had a significant amount of residual pressure on it when it needs to be open, its purpose is to relieve this internal pressure caused by combustion. Because the residual pressure in our experiments exceeded the strength of the actuators for the exhaust stroke only it was decided to modify the head assembly to house the original valve train and camshaft components for the exhaust only. This would guarantee the exhaust valve would open when it is supposed to regardless of the exhaust pressure.

To re-install the exhaust valve train, modifications needed to be made to the current assembly. A 7/8" slot with a DOC of .030" was need on the exhaust valve side of the threaded solenoid plate for rocker arm clearance. Also two through holes were needed for rocker arm stud clearance on both the threaded solenoid plate and solenoid tube plate (refer to drawings 11521117 and 11521118 in Appendix B), although the exhaust was the only portion using the camshaft to facilitate installation the intake rocker arm stud was installed to located the pushrod guide plate (re-using the OEM system).

4.3 Future Improvements

4.3.1 Fuel injection

Initial plans for the project included fuel injection, but due to an already complex goal the use of a carburetor was used instead. For future improvements the use of fuel injection could be incorporated as fuel can be supplied exactly when needed and a in a precise amount. This would allow for tuning of the engine if an oxygen sensor was incorporated into the exhaust stream as well. Using an injector duty cycle of 45-75% (typical range), for a single cylinder 6.5 hp engine with a brake specific fuel consumption of 0.5 (naturally aspirated) a 7.2 lb/hr or 4.3 lb/hr injector would need to be used respectively. This would aid in maintaining a proper air-fuel mixture for all ranges of engine RPM once tuned and calibrated correctly.

To properly tune and calibrate a base fuel table would have to be created based on engine speed, manifold pressure and air temperature. Depending on the mentioned parameters calculations can be made to assume a safe value for fuel injected, then through self analysis of the oxygen sensor the code can richen or lean the fuel mixture over time to self tune itself to reach the desired air-fuel mixture.

4.3.2 O_2 Sensor

Further modeling or alterations to components could be done to incorporate an O_2 sensor to obtain more data from the combustion process. Data would be gathered from the engine running with its camshaft. It would then be compared to the project's secondary objective set-up. This additional information would enable one to tune an engine that lowers emissions by adjusting the valve's offset.

5 VALIDATION

5.1 Primary objective

Refer to APPENDIX A drawing 11521001 for a system diagram of the validation setup.

Various tests on the engine head containing the EVAS (is this defined somewhere) and then compared to previous measurements of the camshaft (indicate where). Using Visual Basic for Applications (VBA) with software written by the team, that allows for file conversion, data manipulation and plotting. Then tests on the engine head are able to be run and then plotted in near real time, which reduces time between testing, as tests can be validated quickly, and reran if there is an issue with data.

5.1.1 Results

Compiling test results for 300 RPM to 1750 RPM of valve position and time, it was found that on average there is a 1.84ms lead on opening, and 1.63ms lag on closing, when compared to the camshaft system, according to the tests conducted. Figure 13 below is an example of the data gathered to verify solenoid operation.

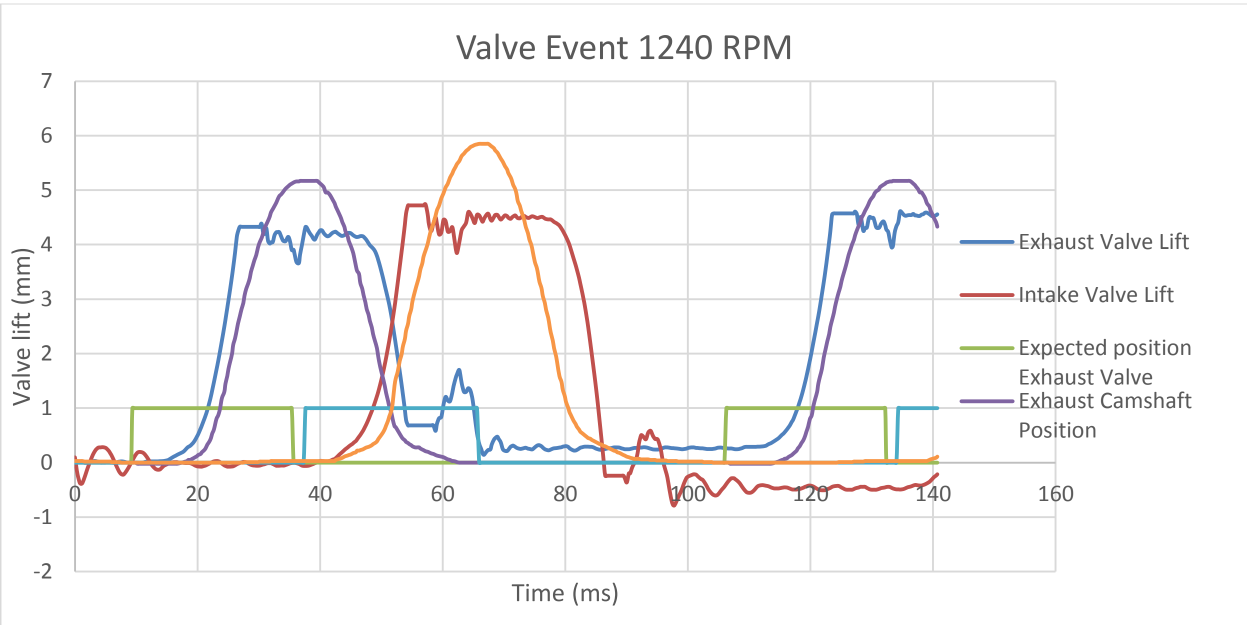


Figure 13: data gathered to verify solenoid operation for valve event at 1240 RPM

To validate that the valve would be acceptable for air flow into an engine – the valve lift data was integrated with respect to time and compared to the same lift that the camshaft could expect. Figure 15 and Figure 14, are a representation of the valve lift multiplied by time, to compare the area with that of the camshaft

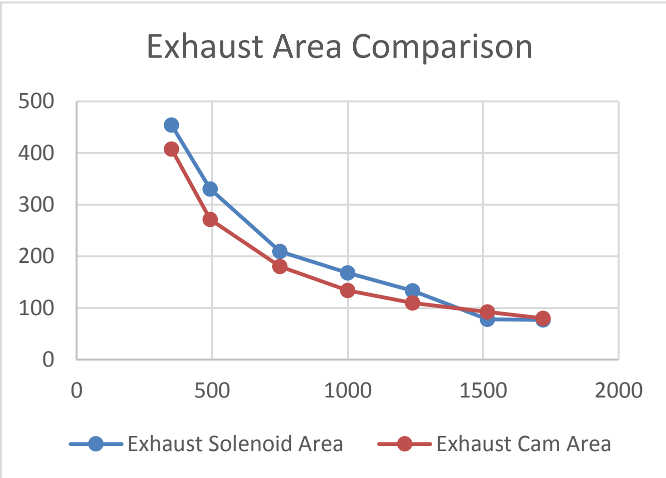


Figure 15: Exhaust area comparison

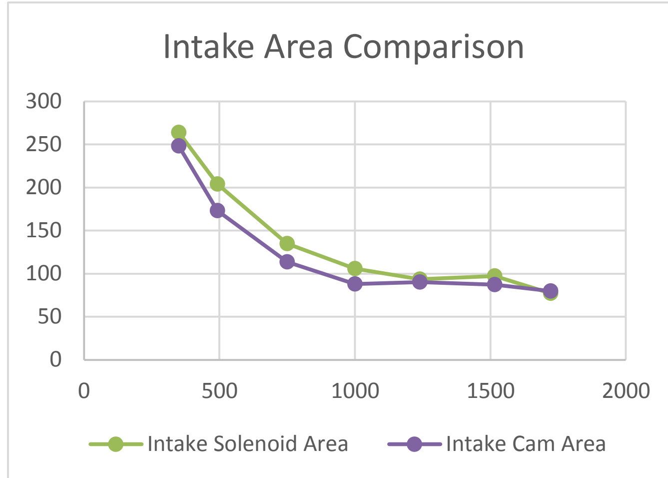


Figure 14: Intake area comparison

The results yielded on average 11% difference between the tests ran. Looking at the plots, it can be seen that before 1200 RPM the area of the camless system is greater – which is to be expected due to the actuation speed is much greater when compared to the crankshaft speed at low RPM. At the design objective of 1200 RPM the system performs as expected and gives a comparable area to that of the camshaft. After 1200 RPM the camless system does start to give less area, as the solenoids cannot keep up with the speed of the engine. The solenoids are a limiting factor in this system, as the actuation time of the solenoid is approximately 16ms for actuation and return, and as seen in Figure 16 after 2600 RPM the system can no longer compensate for actuations delays, and keep valves open for the required amount of degrees. Faster solenoids and stronger return springs would help in increases the maximum speed of the engine, as the system delay would be less. Considering the results obtained, from these tests, it was deemed that the system could operate an engine at speed up to 2500 RPM

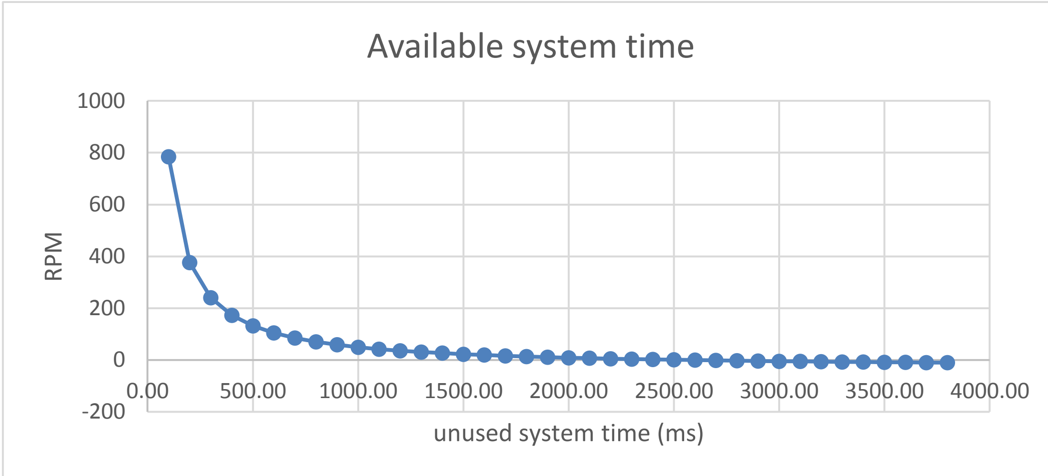


Figure 16: system cannot compensate for actuations delays

5.2 Secondary objective

Refer to APPENDIX A drawing 11521000 for a system diagram of the validation setup.

A pressure sensor and MAP sensor were added to the system in order to obtain their measurements of the engine via an oscilloscope. Firstly, these measurements were taken while running the original camshaft on the engine, as seen in Figure 17 below.

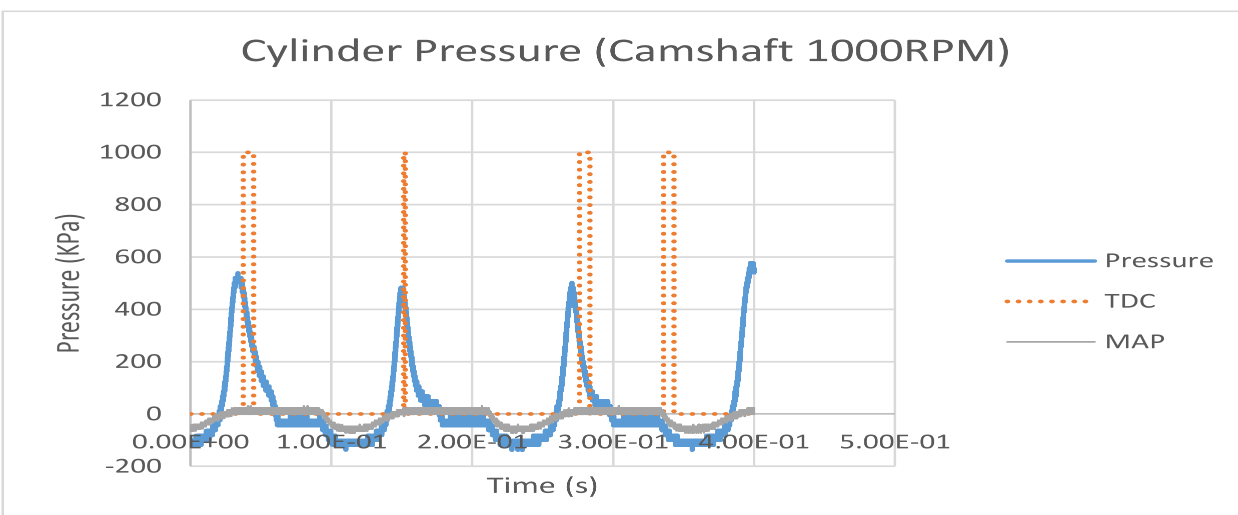


Figure 17: Camshaft Engine Pressure, MAP measurements with respect to TDC

The data obtained was expected. The data obtained will be explained using Figure *18* which illustrates the four stroke engine cycle. As seen in Figure *17*, pressure spikes occurred at top dead center which is to be expected to occur between the compression stroke and working stroke of the four stroke engine. This is shortly followed by a slight positive pressure representing the exhaust stroke whereby exhaust gases exit the chamber. Lastly, right before the next pressure spike, a slight negative pressure occurs representing the intake stroke whereby fresh air and fuel are sucked into the chamber.

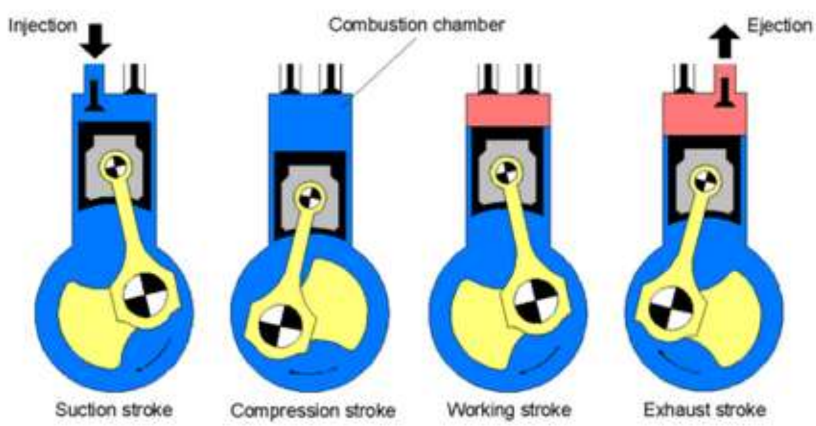


Figure 18: Four-Stroke Engine Cycle [8]

Having obtained the camshaft engine's data to be used a baseline, the designed and manufactured camless head has been mounted onto the engine. The camless engine was able to run for a maximum of 8 seconds long before stopping. Figure 19 illustrates the data obtained from this run.

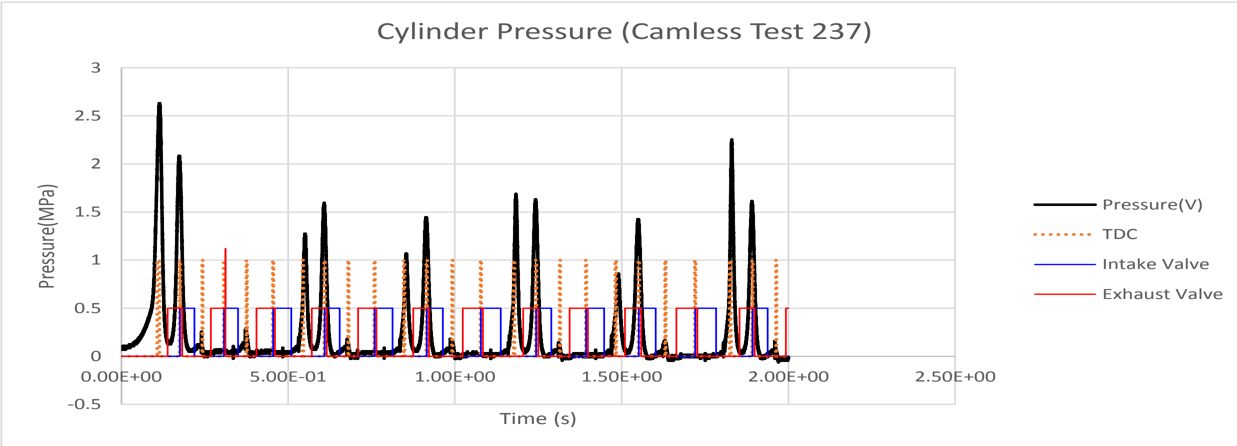


Figure 19: Camless Engine Pressure, MAP measurements with respect to TDC

When compared to the Figure 18, the data obtained for the camless engine is significantly different for a few reasons. Firstly, two pressure spikes occur during every other cycle of the four stroke engine. In other words, the initial pressure spike which occurs between the compression stroke and the working stroke is to be expected. However, there is an additional pressure spike between the exhaust and intake stroke of the engine cycle which is not supposed to occur. Furthermore, the cycle that follows this unanticipated behavior exhibits no pressure spike whatsoever until the following cycle. The reason behind this behavior is due to the strength of the actuators. The actuator is too weak to open when the chamber experiences 7 psi and higher—as tested with a pressure gauge and compressed air. This is an undesirable limitation since engine chambers undergo such pressures. The double spike can therefore be explained. Since the exhaust valve is unable to open due to the pressure found inside the chamber, and thus not allowing the exhaust gas to exit the chamber, the engine experiences a second pressure spike. Then on the following power stroke, enough energy in the form of pressure has decreased enough for the exhaust valve to open on the second power stroke. Due to this phenomenon the system was able to run intermittently, but unable to sustain full combustion.

5.3 Solenoid Actuated Intake Valve

To provide a meaningful experiment, and obtain research data, the intake valve solenoid was actuated and then the valve event timing was varied through a potentiometer on the control box. By delaying intake opening it is possible to change the amount of residual exhaust gasses in the combustion chamber, while testing it was possible to increase engine rpm by delaying intake opening until after the exhaust valve had closed, this works by having a stronger concentration of air-fuel mixture. By delaying intake closing it is possible to pass the point where the engine goes further than BDC with the intake valve open and begins pushing air-fuel mixture out the intake; at this extreme it has no practical use because it is possible to expel all air-fuel charge, or enough to not limit combustion. In theory this could be used to increase air-fuel volume to the engine as long as the valve is closed before BDC. With further testing and development it would be possible to simulate various valve timing events using the solenoid actuated system rather than changing camshafts as current vehicles do.

After experimenting with the system it is possible that the project could have used a smaller actuator on the intake valve and a larger actuator on the exhaust valve, this could possibly solve the residual pressure issue while also maintaining the current overall package size. The solenoids were initially chosen with the primary objective in mind while having the additional requirements to attempt the secondary objective, they are appropriately sized and offer occasional complete combustion, with fuel injection it is possible to lower the cylinder pressure by injecting less fuel and controlling the intake valve for airflow.

6 BUDGET

See Appendix I for mini PDM

The total cost of the project without samples, donations or free components from the EDML is just under $1200, by using all the resources available for the project it was possible to complete it for a grand total of just under $620. Resources used include but are not limited to; early submission of drawings for manufacturing in EDML A, raw materials from EDML B, components from EDML C and samples from electrical manufacturers. By making large orders and combining shipping from manufacturers it was possible to also conserve funds, the key to making the most out of any budget is proper planning and organization when ordering or shipping components.

7 GRADUATE ATTRIBUTES

7.1 Cooperation

The group dynamic is near ideal. All group members attend at least ninety percent of the team meetings. During these meetings, all team members actively participate by asking questions and explaining what needs to be done such that all members are on the same page. There are several means by which the group communicates: Facebook, Asana, Google Drive, and meetings. Since all members of the group have similar communication styles, this enables the project to progress smoothly. The team members have been especially cooperative during testing and drawing sessions such that everyone gave their inputs on the process, assisted one another when needed as well as give explanations when certain aspects may have seemed unclear for some members.

7.2 Practical Contribution

As previously mentioned, all team members work well together. This can be illustrated in the division of work concerning the reports. Each member either volunteers or is assigned a section of the report. The report is then compiled by one team member and proof read by another two to three members in order to provide constructive feedback what has been written. In order to ensure that the report includes the required content, team meetings are scheduled the week before the report is due. This demonstrates the organizational skills of our team leader, Alex Boehler, as well as the contribution and cooperation of all members. The team leader is the main team member that ensures the organization of various aspects of the project. All team members facilitate the organization of the project by responding efficiently to the team leader's needs. For example, the team leader keeps the capstone binder updated by asking each member to print specific documents on their work.

7.3 Conceptual Contribution

The camless engine project is separated into three main categories: design, hardware, and software. Each category requires its own set of research, some more than others. The design aspect of the project includes all the 3DCAD drawings, such as that of the test stand and the mount head. This requires more brainstorming in terms of design as oppose to research. While CADing the different components of the project, these members keep the others up to date with their design process, ask others for a second opinion, and ensure that their design has been agreed upon. The hardware aspect of the project required some research on which electronics to purchase as well as test and calibrate these electronics. As with the design team, the hardware team members discussed their findings with the entire group in order to have agreed upon decisions. Due to everyone's contribution and curiosity, most team members were present during both the hardware and software testing sessions. The software aspect of the project requires research in order write proper code in the microcontroller to control the solenoids such that it yields the desired response. Since most team members have taken programming courses, these members volunteer to help debug the code when needed. Overall, ideas are generated by various team members for all categories of the project to ensure the quality of work.

The main difficulty that the group has come across is to accommodate their schedules to that of Gilles Huard since the electrical equipment in room H-1057 is needed for testing and calibration of the purchased electronics. A solution to this is to email both Gilles Huard and Dominic Ng to rent the necessary equipment from room H-1057 to be used in room EDML-C.

An additional problem includes delay occurrences in shipment. This delays the testing and advancement of certain components of the project. The solution to this is to anticipate for these kinds of delays and work on other courses in order to be available once the parts arrive.

7.4 Work Ethic

The main reason why this group demonstrates excellent work ethic is because they had selected its own members based on previous experience. It is in the personality of each team member to be respectful towards others and their ideas as well as being engaged in projects. If there were to be any conflict, it tends to be quickly resolved through means of effective communication. This is an important aspect of team work since team-based projects progress the best when the team works together efficiently. For example, since every member takes their work seriously, deadlines both set by MECH 490 and by the group are met.

8 CONCLUSION

This report summarizes the techniques used in determining the ideal solution to design and fabricate a variable valve actuation (VVA) system. Researching past methods of designing VVA, three main methods were found: electromagnet, pneumatic, and hydraulic designs. Careful consideration and discussions with professors determined electromagnetic would be the most feasible.

The main objective of this capstone project was to design a VVA system which will replicate the lift and phase separation of a single cylinder internal combustion engine. To realize this goal, the team is split into three groups: Software, Hardware, and Design and Manufacturing teams. The software team designed the code for the microcontroller to actuate the solenoids based on engine parameters. The hardware team selected all sensors, solenoids, and electronics involved in the VVA system. Lastly, the modeling and manufacturing team designed the hardware and its interface with an internal combustion engine.

Running the validations test for the camless engine system yielded satisfactory results with only a average 11% error graphical integrated area when compared to the camshaft profile – that would indicate the engine will be able to flow sufficient air to run. The test results also indicated that the system had a slight error in actuation time, with an approximate lead before opening of 1.5 ms and 1.6 ms lag after closing. The discrepancy after system delay compensation seems like it is caused by data acquisition limitations, and has no effect on the operation of the valve actuation timing.

Running the camless engine system on the carbureted engine initially yielded positive results, as tests proved the system was capable of running; with a max run time of eight seconds. Although the engine ran, the combustion was generally intermittent and found not to combust every cycle.

The problem was tracked to their being too much exhaust residual, and with a cylinder chamber pressure greater than 7 PSI, the exhaust valve would not actuate.

To deliver a working engine prototype using an electromagnetic valve actuation system, the design was changed to a camshaft being used to control the exhaust valve, as it would guarantee valve actuation, and the solenoid actuator would remain on the intake valve – to allow for controllable intake valve. From experimentation with the intake valve open and closing timing, it was found that reducing the overlap between the intake and exhaust valve increased engine RPM, while increasing the intake closing time past BDC, would cause the engine to slow down and eventually shut down.

To further advance the project and be able to have a completely working camless engine some equipment would need to be improved. First, the solenoid chosen to actuate the valves must be capable of at least being able to actuate under 10 PSI of cylinder pressure, therefore stronger solenoids area must to continue this project. The solenoids should also preferably have a linear force rating, such that a PID controller could be developed to control the solenoid in a full feedback loop with linear position sensors. Greater combustion control could be achieved by using a fuel injection system, as it would all for variability over air fuel mixtures, and exhaust gas sensing, from and oxygen sensor. Lastly microcontroller or processor would need to be significantly quicker or multiple microcontrollers should be used, as to allow for less delay from commands, and quicker data acquisition.

REFERENCES

[1] H. H. a. B. G. G. B. Parvate-Patil, "An assessment of Intake and Exhaust Philosophies for Variable Valve Timing," in 2003 SAE/JSAE Small Engine Technology, Madison, 2003.

[2] L. L. Peters, D. A. Vittorio, A. S. Ghuman and H. G. Weber.

[4] J. S. Brader, "Development of a Piezoelectric Controlled Hydraulic Actuator for a Camless Engine," University of South Carolina , South Carolina , 2001.

[7] "Long Life Solenoid Selection," Ledex, Johnson Electric, 2014. [Online]. Available: http://www.ledex.com/long-life-solenoids/long-life-solenoids.html

[9] Y. Wang, A. Stefanopoulu and R. Smith, "Inherent limitations and control design," INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL, pp. 1023-1042, 2001.

[10] J.-M. Kang and J. W. Grizzle, "Dynamic control of a SI engine," INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL, pp. 399-420, 2003.

[11] G. B. Parvate-Patil, H. Hong and B. Gordon, "Analysis of Variable Valve Timing Event and Their Effects on Single Cylinder Diesel Engine," SAE International, pp. 1510-1520, 2004.

[12] I. Haskara, V. V. Kototovic and L. A. Mianzo, "Control of an electro-mechanical valve actuator," INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL, pp. 561-579, 2004.

[13] J. Donaldson.United States of America Patent 7,204,212, 2007.

[14] M. Hammoud, I. H. Makki and J. M. Kerns.Europe Patent 1,170,489, 2001.

[15] J. L. Liedtke and T. E. Wenzel.United States of America Patent 6,739,294, 2004.

[16] G. R. Babbit and C. S. Balton.United States of America Patent US 2004/0261736 A1, 2004.

[17] G. Parvate-Patil, "Solenoid Operated Variable Valve Timing for Internal Combustion Engines," Concordia, Montreal, 2005.

[23] Jason Meyer, "Engine Modeling of an Internal Combustion Engine With Twin Independent Cam Phasing," Ohio State University, 2007.

[24] Perihan Sekmen and Yakup Sekmen, "Mathematical Modeling of a SI Engine Cycle With Actual Air-Fuel Cycle Analyses," Karabuk University, Turkey, 2007.

[25] Marcin Slezak and co., "Mathematical Model of Four-Stroke Combustion Engine Working Process," Journal of KONES Powertrain and Transport, Vol. 14, No.3, 2007.

[26] Sitthichok Sitthiracha, "An Analytical Model of Spark Ignition Engine For Performance Prediction," King Mongkut's Institue of Technology North Bangkok, 2006.

[27] D. J. Podnar, "Development of an Electromangetic Valve Actuation System on a Kohler Engine," U.S. Army TARDEC Fuels and Lubricants Research Facility, San Antonio, 1998.

WEB REFERENCES

[3] Cargine, "Cargine," [Online]. Available: http://www.cargine.com/technology/improved-performance/. [Accessed 10 August 2014].

[5] X. Lan and Y. Liu, "Life Cycle Assessment of Lawnmowers: Two Mowers Case Studies," Department of Energy and Environment, Chalmers University of Technology, Sweden, 2010. [Online]. Available: http://publications.lib.chalmers.se/records/fulltext/141490.

[6] American Chemical Society, "One Hour of Grass Cutting Equals 100 Miles Worth of Auto Pollution," ScienceDaily, 2001. [Online]. Available: www.sciencedaily.com/releases/2001/05/010529234907.html

[8] D.-I.-D. Ngo, " TU Berlin", [Online]. Available: http://www-mat.ee.tu-berlin.de/research/sic_sens/sic_sen3.htm.

[18] MAN B&W, "ME Engines - the New Generation of Diesel Engines," [Online]. Available: http://www.mandieselturbo.com/1002810/Press/Publications/Technical-Papers/Marine-Power/Low-Speed/Low-Speed-Archive/ME-Engines---the-New-Generation-of-Diesel-Engines.html. [Accessed 08 August 2014].

[19] "Understanding Cam Profile Terms," Lunati, [Online]. Available: https://www.lunatipower.com/Tech/Cams/CamProfileTerms.aspx.

[20] "Choosing The Right Cam," Hot Rod Magazine, 07 07 2004. [Online]. Available: http://www.hotrod.com/techarticles/113_9604_choosing_the_right_camshaft/viewall.html.

[21] "Lobe Separation Angle," Pantera Forum, 11 10 2012. [Online]. Available: http://pantera.infopop.cc/eve/forums/a/tpc/f/5650045562/m/3481028246/p/2.

[22] J. Smith, "Cam Basics," Super Chevy Magazine, 14 05 2002. [Online]. Available: http://www.superchevy.com/how-to/53439-cam-basics/.

[28] Marlan Davis, "Variable Valve Timing - the next phase," Hot Rod, 2009. [Online] Available: http://www.hotrod.com/how-to/engine/hrdp-0908-variable-valve-timing/

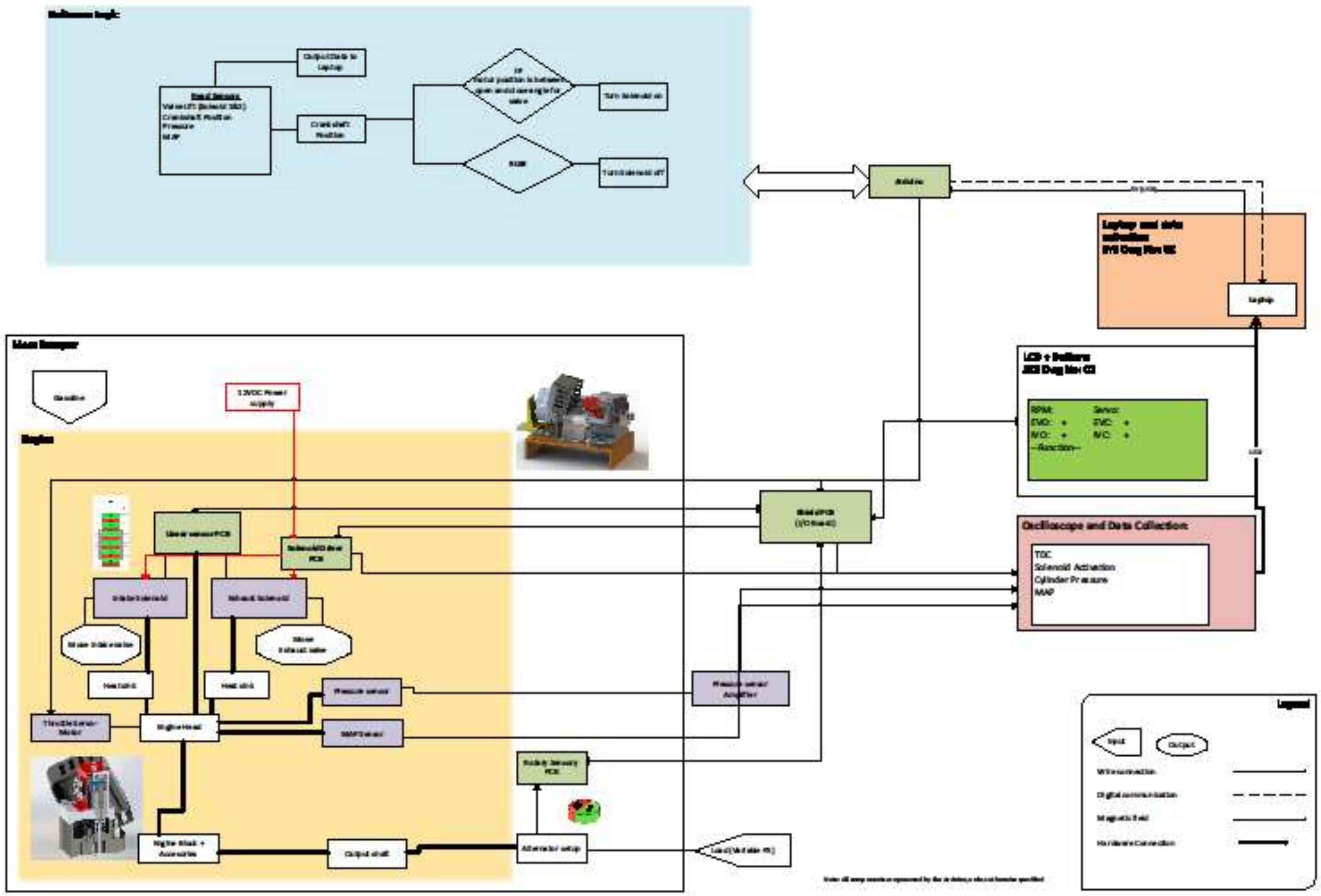
[29] "Variable Valve Timing," CDX Online eTextbook. [Online] Available: http://www.cdxetextbook.com/engines/comp/vlves/variablevalvetiming.html

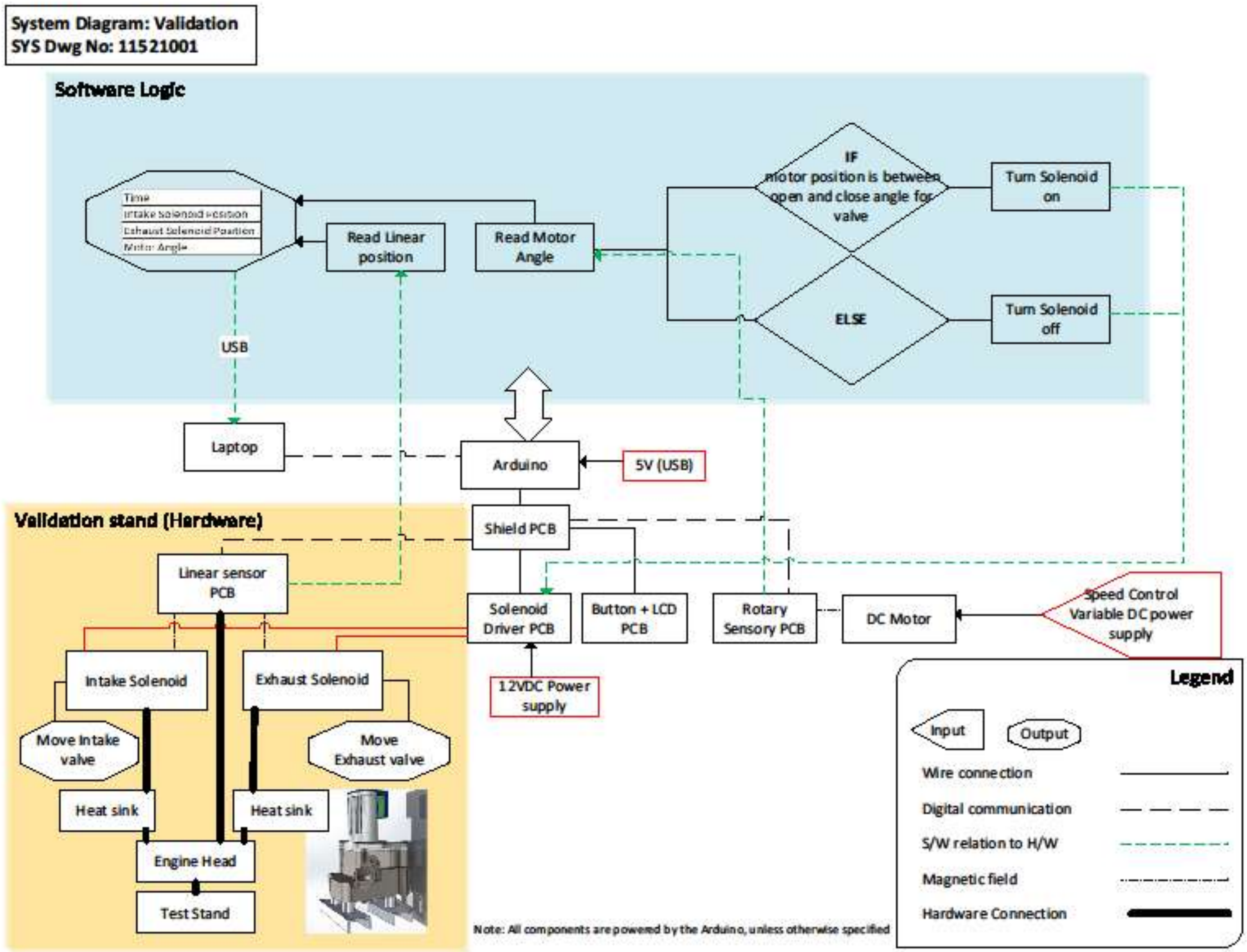
[30] "VVT: Variable Valve Timing," AASA. [Online] Available: http://www.knowyourparts.com/technical-articles/vvt-variable-valve-timing/

[31] "The Impact of Valve Events Upon Engine Performance and Emissions," Mechadyne International, 2006. [Online] Available: http://www.mechadyne-int.com/vva-reference/

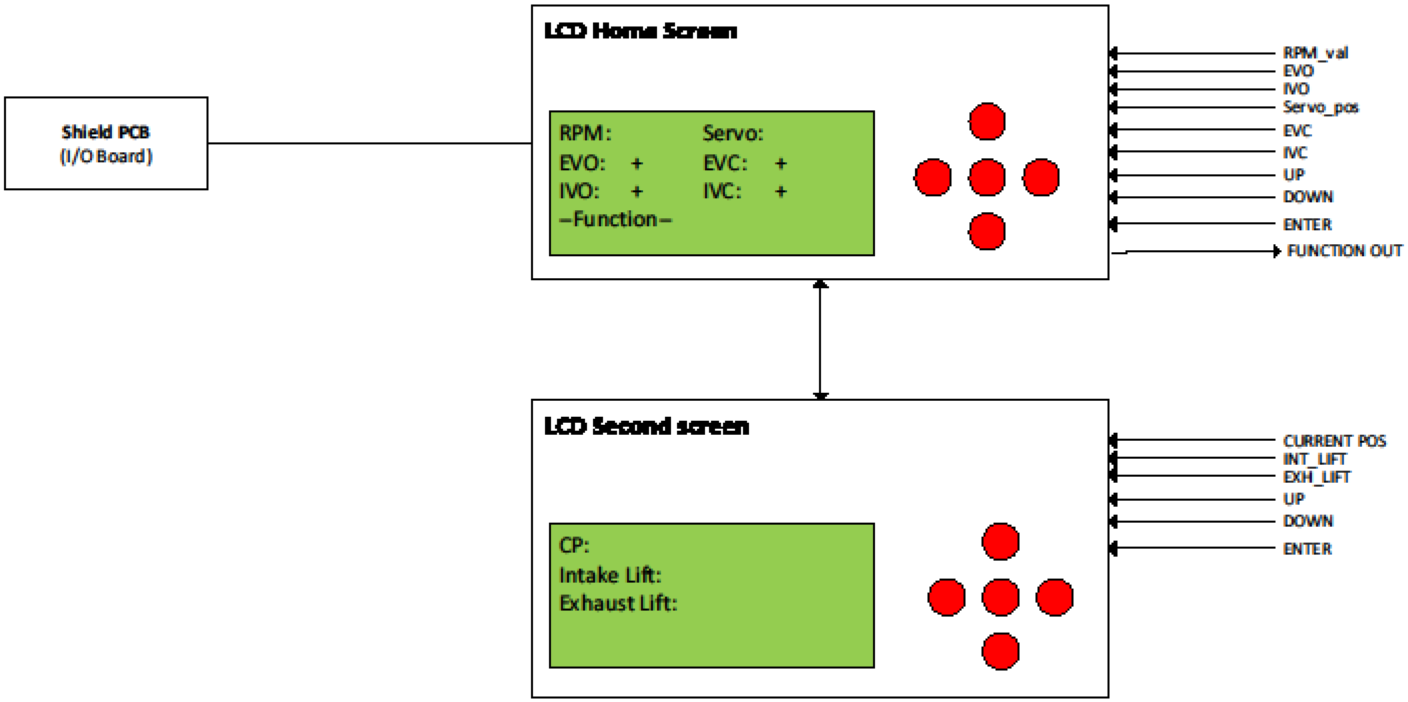
APPENDIX A: SYSTEM DIAGRAMS

System Diagram: Secondary Objective
Overview
SYS Dwg No: 11521000

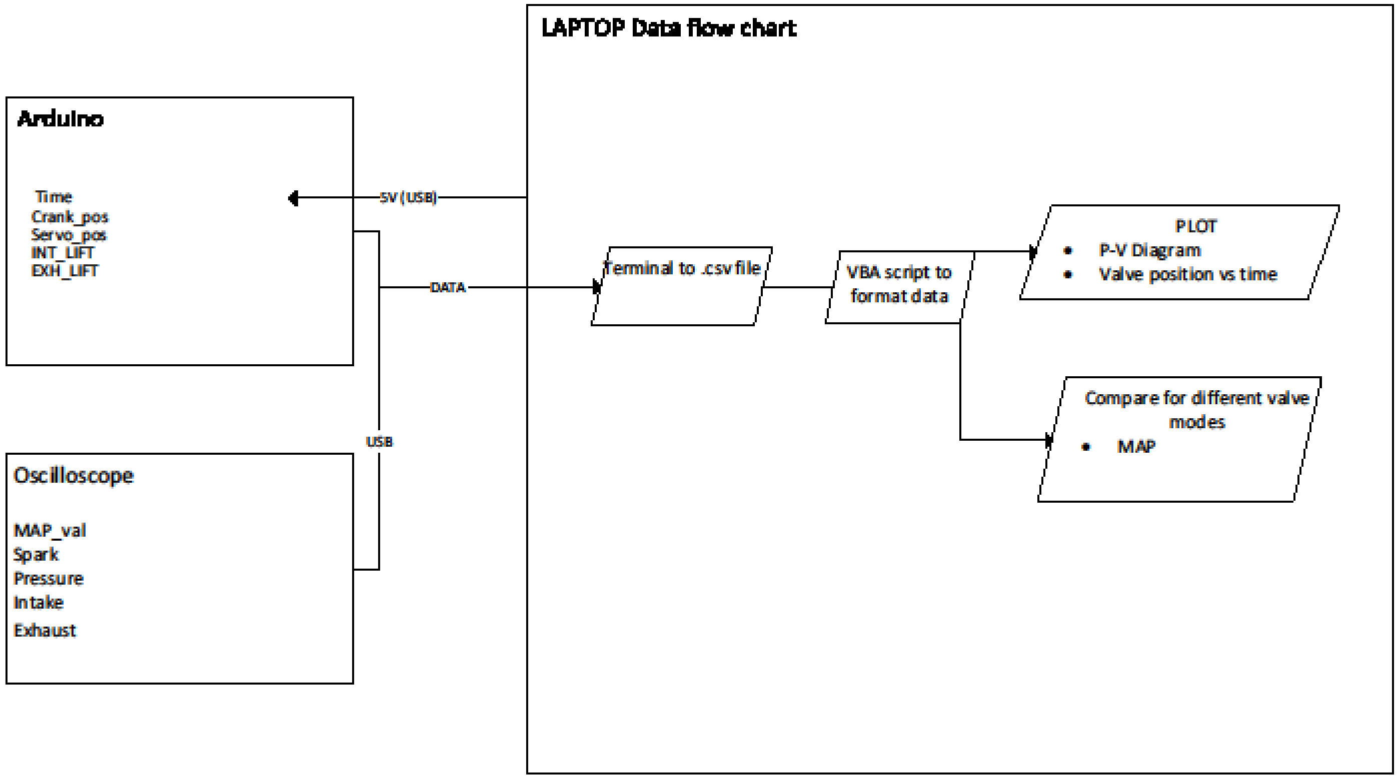


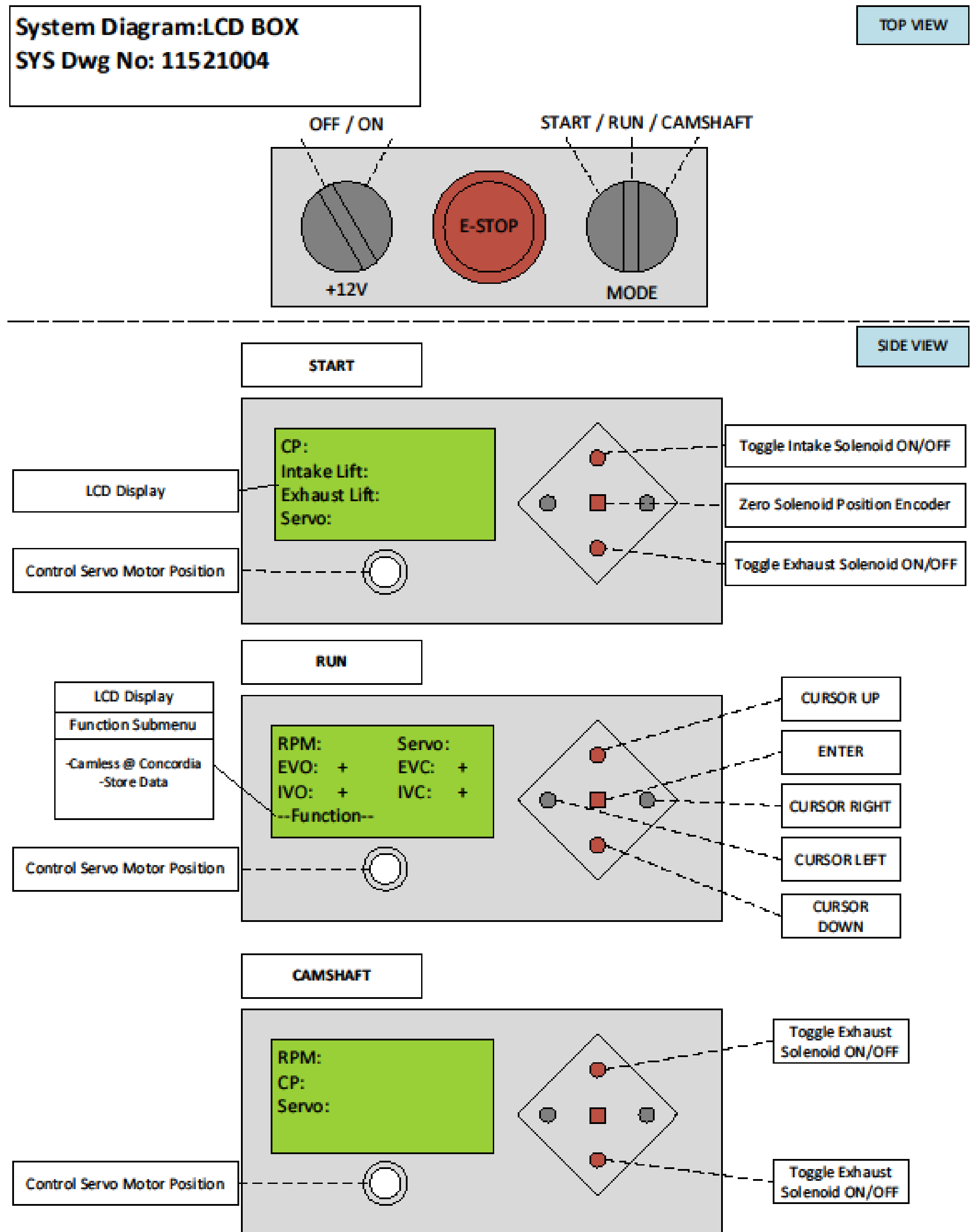

System Diagram: Validation
SYS Dwg No: 11521001
Software Logic
Time
Intake Solenoid Position
Exhaust Solenoid Position
Motor Angle
Read Linear position
Read Motor Angle
IF motor position is between open and close angle for valve
Turn Solenoid on
ELSE
Turn Solenoid off
USB
Laptop
Arduino
5V (USB)
Validation stand (Hardware)
Shield PCB
Linear sensor PCB
Solenoid Driver PCB
Button + LCD PCB
Rotary Sensory PCB
DC Motor
Speed Control Variable DC power supply
Intake Solenoid
Exhaust Solenoid
12VDC Power supply
Move Intake valve
Move Exhaust valve
Heat sink
Heat sink
Engine Head
Test Stand
Note: All components are powered by the Arduino, unless otherwise specified
Legend
Input
Output
Wire connection
Digital communication
S/W relation to H/W
Magnetic field
Hardware Connection

System Diagram:LCD
SYS Dwg No: 11521002



System Diagram: Laptop
SYS Dwg No: 11521003

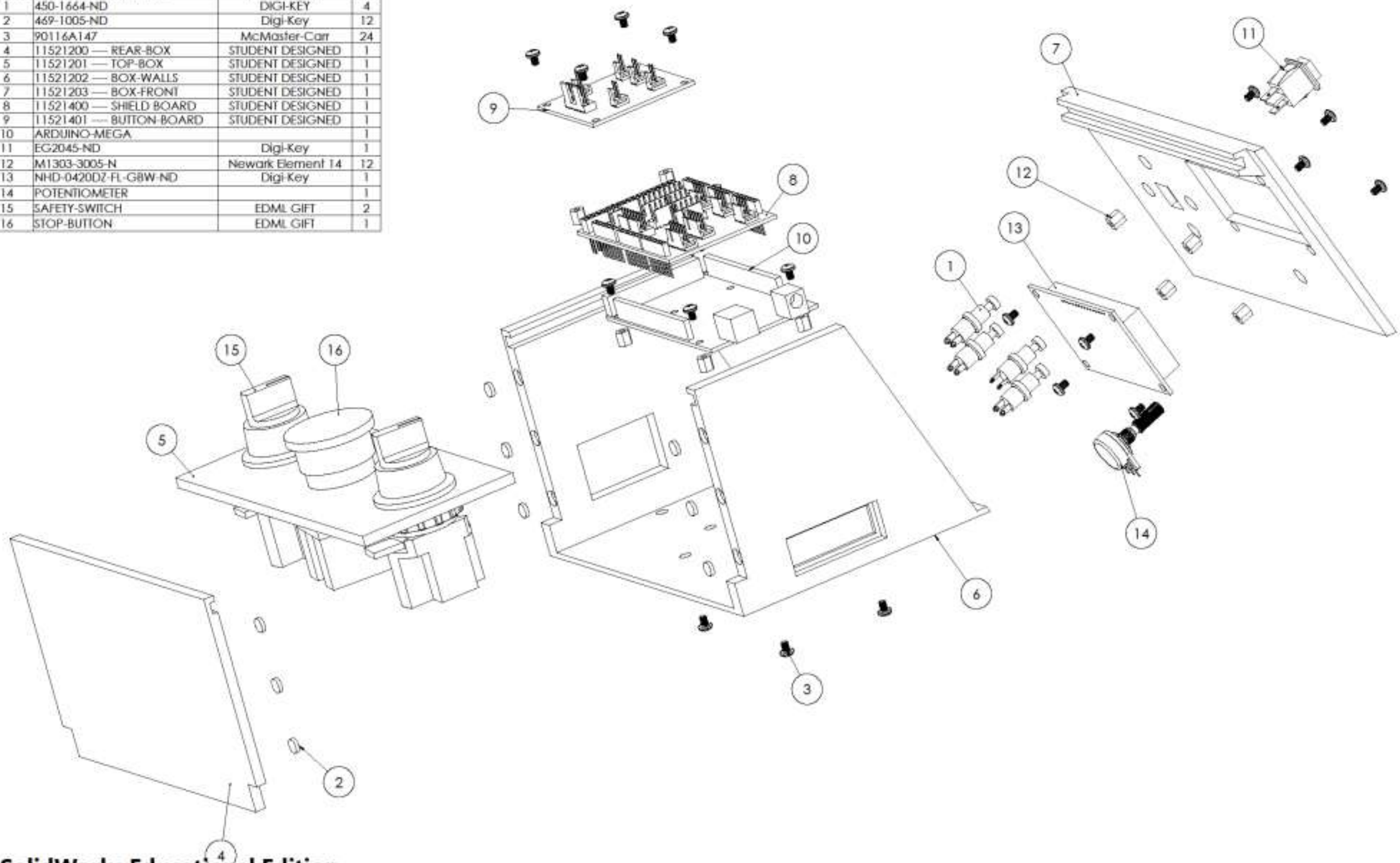



System Diagram:LCD BOX
SYS Dwg No: 11521004
TOP VIEW
OFF / ON
+12V
E-STOP
START / RUN / CAMSHAFT
MODE
SIDE VIEW
START
LCD Display
CP:
Intake Lift:
Exhaust Lift:
Servo:
Control Servo Motor Position
Toggle Intake Solenoid ON/OFF
Zero Solenoid Position Encoder
Toggle Exhaust Solenoid ON/OFF
RUN
LCD Display
Function Submenu
-Camless @ Concordia
-Store Data
RPM:
Servo:
EVO: +
EVC: +
IVO: +
IVC: +
--Function--
Control Servo Motor Position
CURSOR UP
ENTER
CURSOR RIGHT
CURSOR LEFT
CURSOR DOWN
CAMSHAFT
RPM:
CP:
Servo:
Control Servo Motor Position
Toggle Exhaust Solenoid ON/OFF
Toggle Exhaust Solenoid ON/OFF

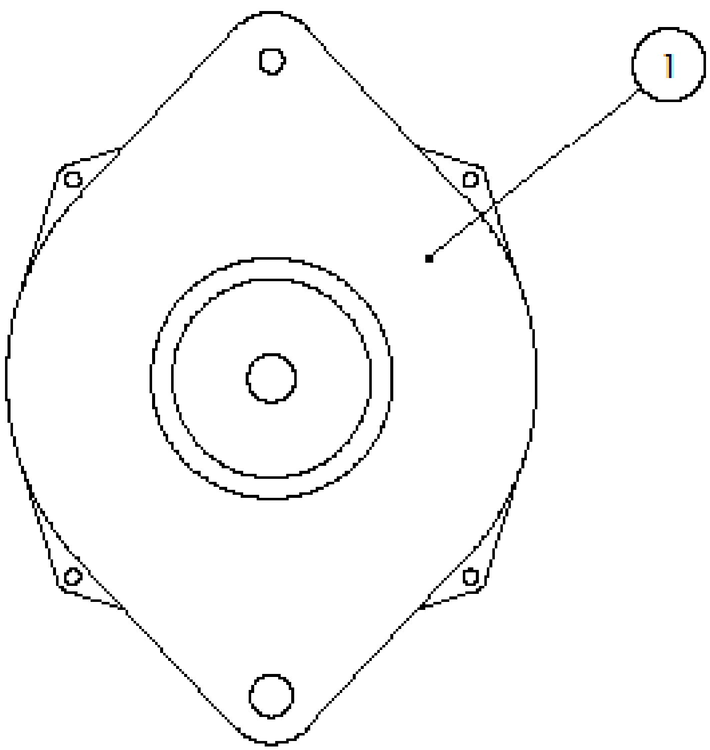
APPENDIX B: ASSEMBLY DRAWINGS

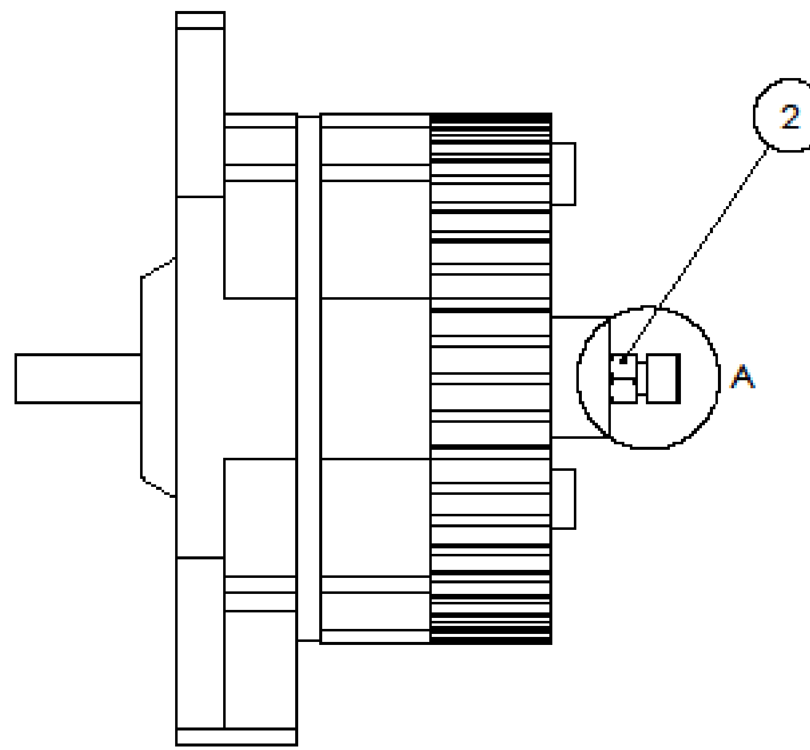
11521100 CONTROL BOX
11521105 ALTERNATOR ASSEMBLY
11521106 ALTERNATOR BRACKET ASSEMBLY
11521107 ENGINE ASSEMBLY
11521108 MUFFLER
11521109 HEAD ASSEMBLY
11521111 ROTARY BRACKET ASSEMBLY
11521112 ROTARY MAGNET ASSEMBLY
11521114 VALVE AND SPRING ASSEMBLY
11521115 GENERAL ASSEMBLY
11521116 GUARD ASSEMBLY
11521117 THREADED PLATE AND SOLENOIDS
11521118 SOLENOID TUBE PLATE AND TUBES
11521119 PLUNGER AND MAGNET CAP
11521403 LINEAR SENSOR

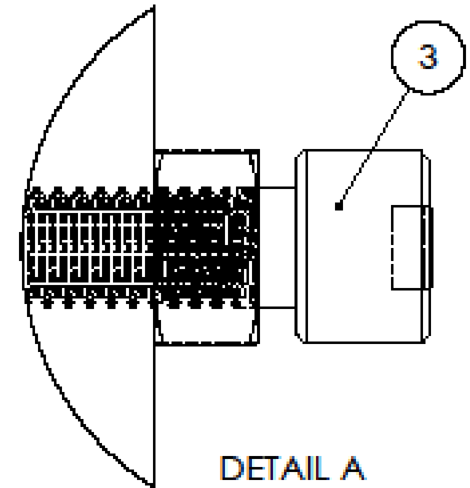
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	450-1664-ND	DIGI-KEY	4
2	469-1005-ND	Digi-Key	12
3	90116A147	McMaster-Carr	24
4	11521200 — REAR-BOX	STUDENT DESIGNED	1
5	11521201 — TOP-BOX	STUDENT DESIGNED	1
6	11521202 — BOX-WALLS	STUDENT DESIGNED	1
7	11521203 — BOX-FRONT	STUDENT DESIGNED	1
8	11521400 — SHIELD BOARD	STUDENT DESIGNED	1
9	11521401 — BUTTON-BOARD	STUDENT DESIGNED	1
10	ARDUINO-MEGA		1
11	EG2045-ND	Digi-Key	1
12	M1303-3005-N	Newark Element 14	12
13	NHD-0420DZ-FL-GBW-ND	Digi-Key	1
14	POTENTIOMETER		1
15	SAFETY-SWITCH	EDML GIFT	2
16	STOP-BUTTON	EDML GIFT	1



No. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	Delco-Remy Alternator		1
2	94804A030	HEX NUT, 5/16"-18	1
3	11521112	ROTARY MAGNET ASSEMBLY	1



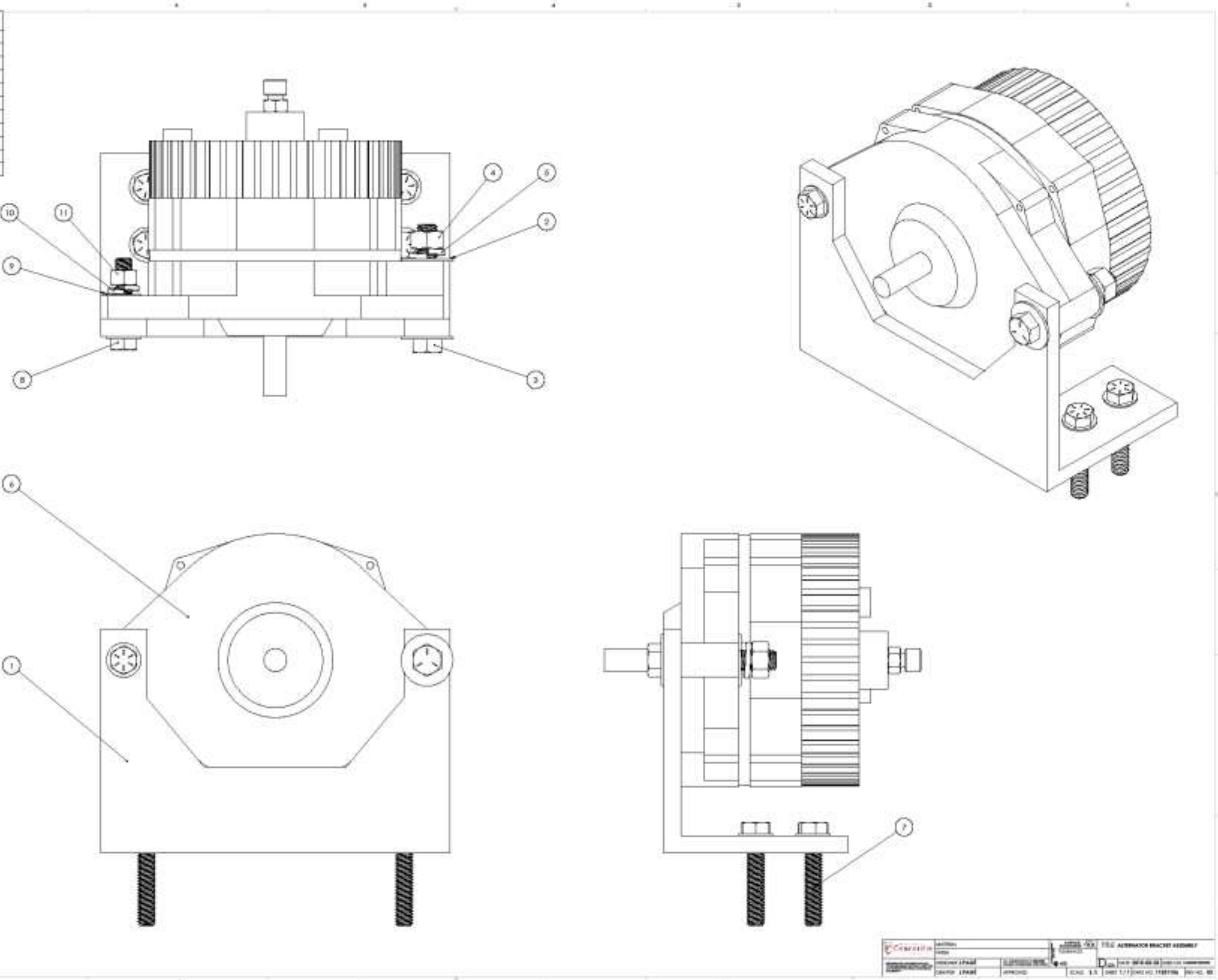




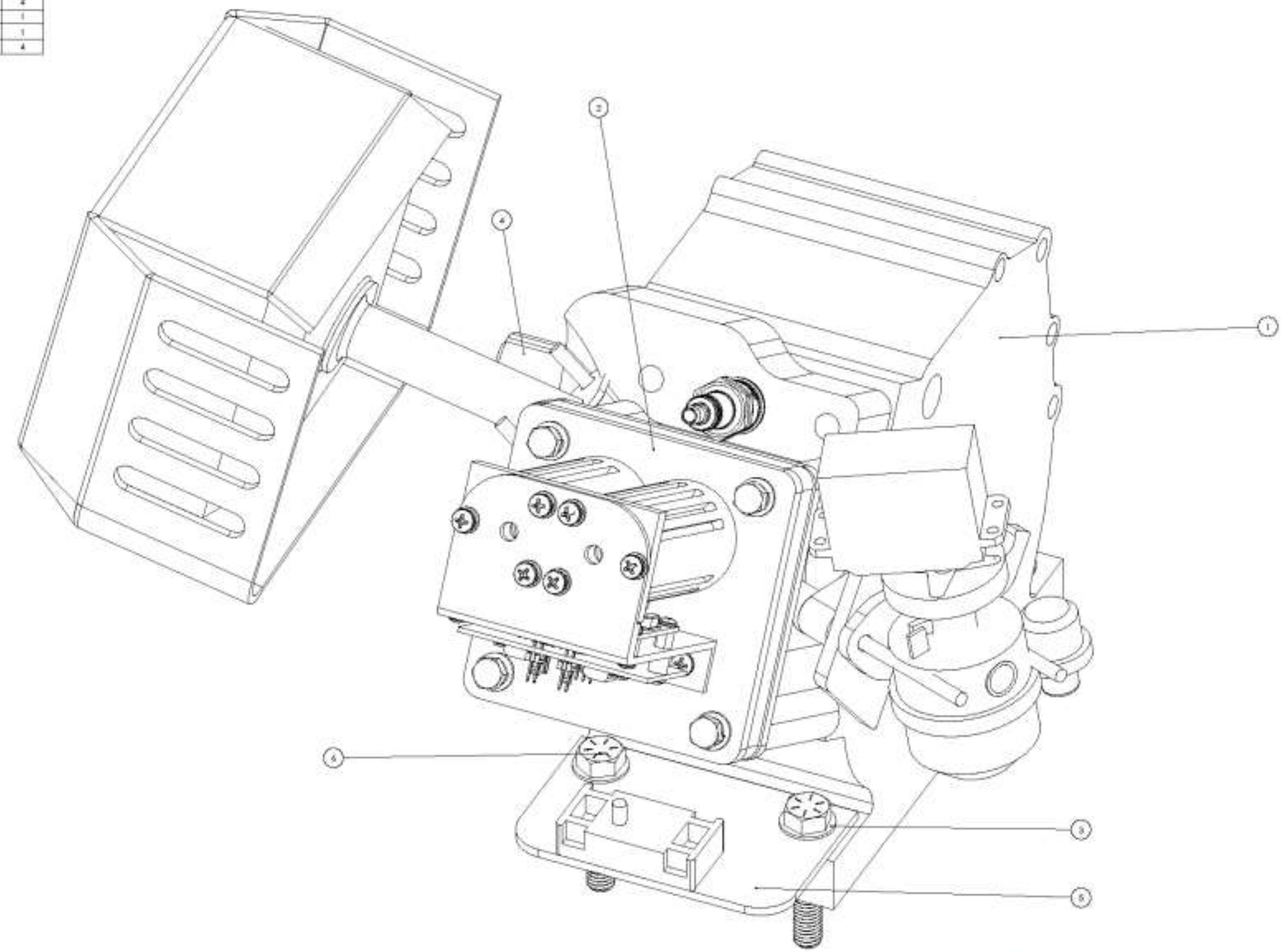
DETAIL A
SCALE 2 : 1

Concordia University	MATERIAL		SURFACE ROUGHNESS √63	TITLE ALTERNATOR ASSEMBLY
	FINISH		TOLERANCES	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE · DATE 2015-03-22 · USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED	SCALE 1:2	SHEET 1 / 1 · DWG NO. 11521105 · REV NO. 02

No. ARTICLE	NUMERO DE PIECS	DESCRIPTION	QTE
1	11021222	ALTERNATOR BRACKET	1
2	90107A032	FLAT WASHER, 7/16"	2
3	92865A677	CAP SCREW, 7/16"-14 2-1/2"LONG	1
4	94804A330	HEX NUT, 7/16"-14	1
5	92147A032	SPLIT LOCK WASHER, 7/16"	1
6	11021106	ALTERNATOR ASSEMBLY	1
7	92620A632	CAP SCREW, 3/8"-16 2"LONG	4
8	92620A630	CAP SCREW, 3/8"-16 1-3/4"LONG	1
9	90107A127	FLAT WASHER, 3/8"	6
10	92147A031	SPLIT LOCK WASHER, 3/8"	1
11	94804A320	HEX NUT, 3/8"-16	1

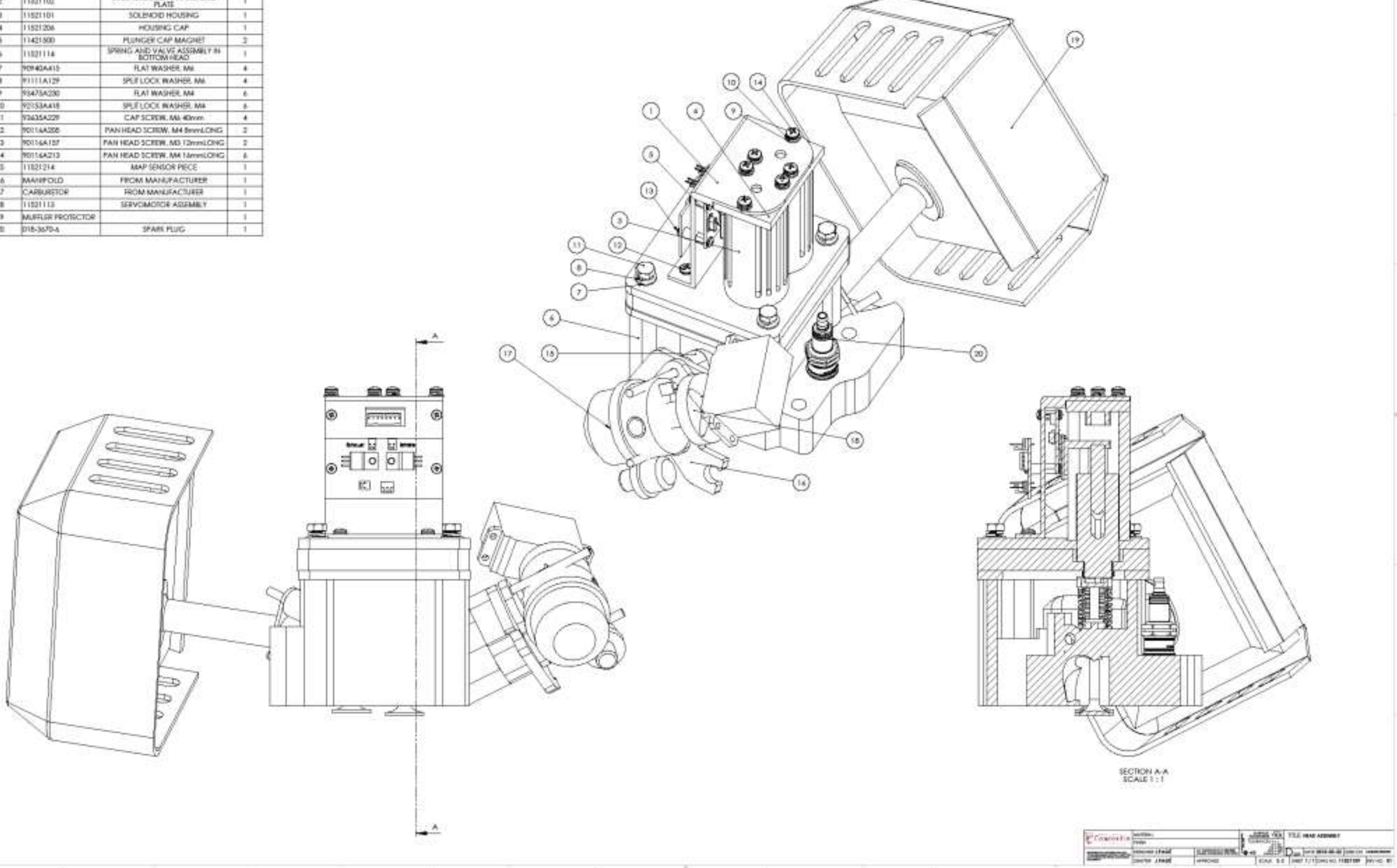


No. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	ENGINE BLOCK	FROM MANUFACTURER	1
2	11021109	HEAD ASSEMBLY	1
3	90107A127	FLAT WASHER, 3/8"	4
4	ENGINE OUTPUT SHAFT	FROM MANUFACTURER	1
5	16254539	MAP SENSOR	1
6	92620A632	CAP SCREW, 3/8"-16 2"LONG	4

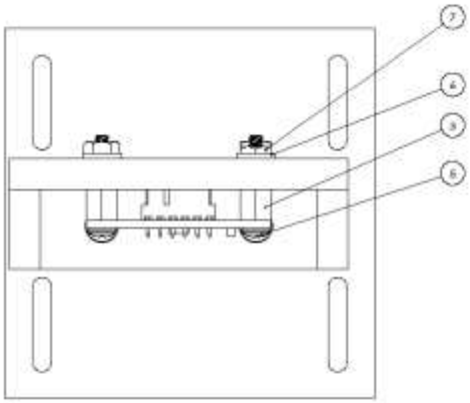


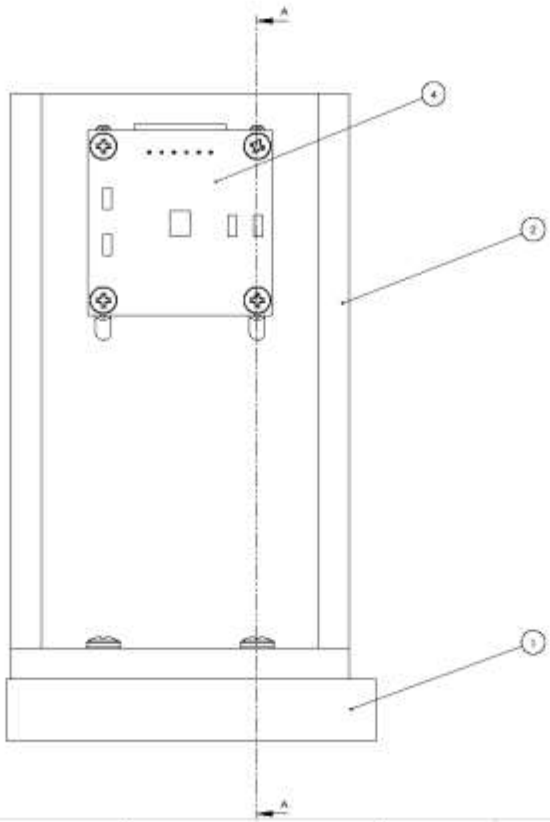
TITLE: ENGINE ASSEMBLY

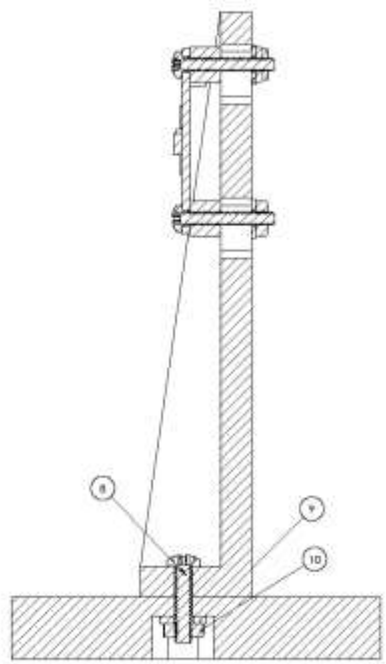
NO. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	11521403	LINEAR SENSOR WITH HOLDER	1
2	11521102	SOLENOID BODY IN THREADED PLATE	1
3	11521101	SOLENOID HOUSING	1
4	11521206	HOUSING CAP	1
5	11421500	PLUNGER CAP MAGNET	2
6	11521114	SPRING AND VALVE ASSEMBLY IN BOTTOM HEAD	1
7	90940A415	FLAT WASHER, M6	4
8	91111A129	SPLIT LOCK WASHER, M6	4
9	93475A230	FLAT WASHER, M4	6
10	92153A418	SPLIT LOCK WASHER, M4	6
11	93635A229	CAP SCREW, M6 40mm	4
12	90116A208	PAN HEAD SCREW, M4 8mmLONG	2
13	90116A157	PAN HEAD SCREW, M3 12mmLONG	2
14	90116A213	PAN HEAD SCREW, M4 16mmLONG	6
15	11521214	MAP SENSOR PIECE	1
16	MANIFOLD	FROM MANUFACTURER	1
17	CARBURETOR	FROM MANUFACTURER	1
18	11521113	SERVOMOTOR ASSEMBLY	1
19	MUFFLER PROTECTOR		1
20	018-3670-A	SPARK PLUG	1



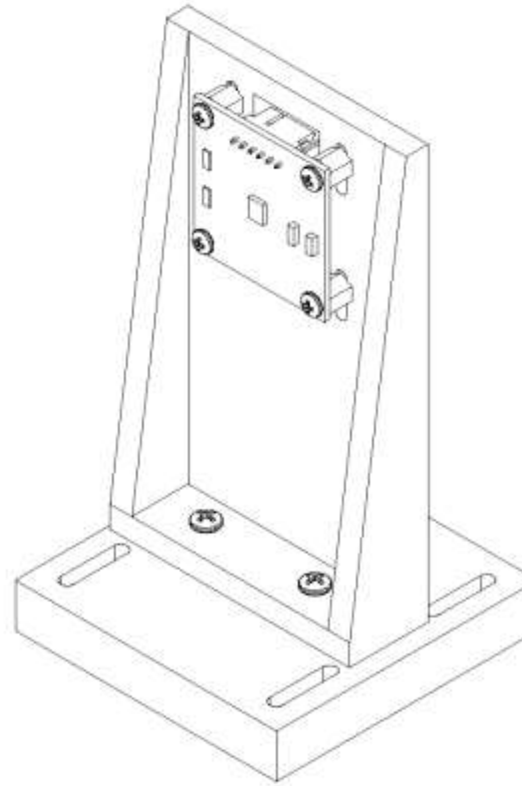
NO. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	11521216	ROTARY SENSOR BRACKET BASE	1
2	11521217	ROTARY SENSOR BRACKET TOP	1
3	ROTARY SENSOR STANDOFF	STANDOFF, 4-40	4
4	11521404	ROTARY SENSOR	1
5	91772A113	PAN HEAD SCREW, 4-40 3/4"LONG	4
6	94659A101	FLAT WASHER, NUMBER 4	4
7	90480A005	HEX NUT, 4-40	4
8	91772A150	PAN HEAD SCREW, 6-32 5/8"LONG	2
9	94659A102	FLAT WASHER, NUMBER 6	2
10	90480A007	HEX NUT, 6-32	2





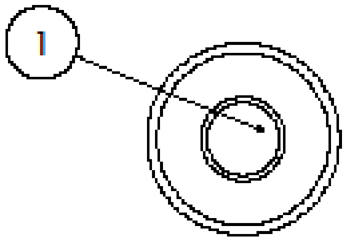


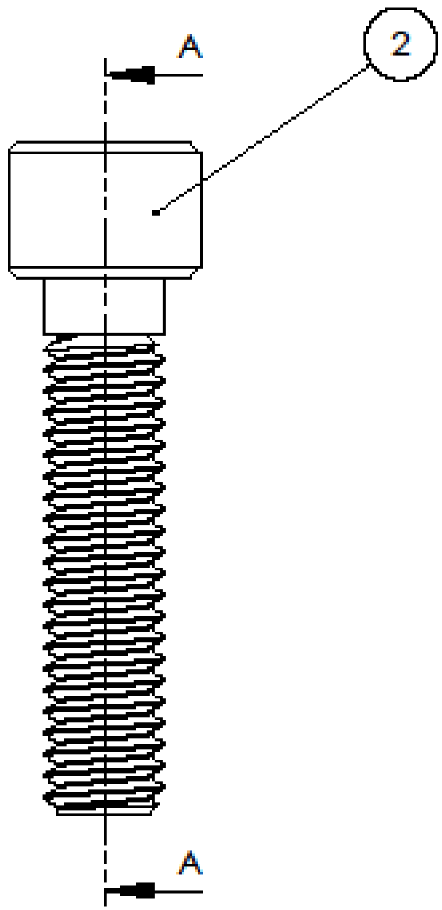
SECTION A-A
SCALE 2 : 1

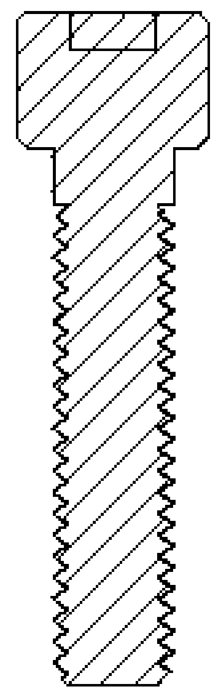


APPROVED | SCALE 2:1 | SHEET 1/1

No. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	AS5000-MD6H	ROTARY MAGNET	1
2	98750A037	ROTARY SENSOR MAGNET HOLDER	1



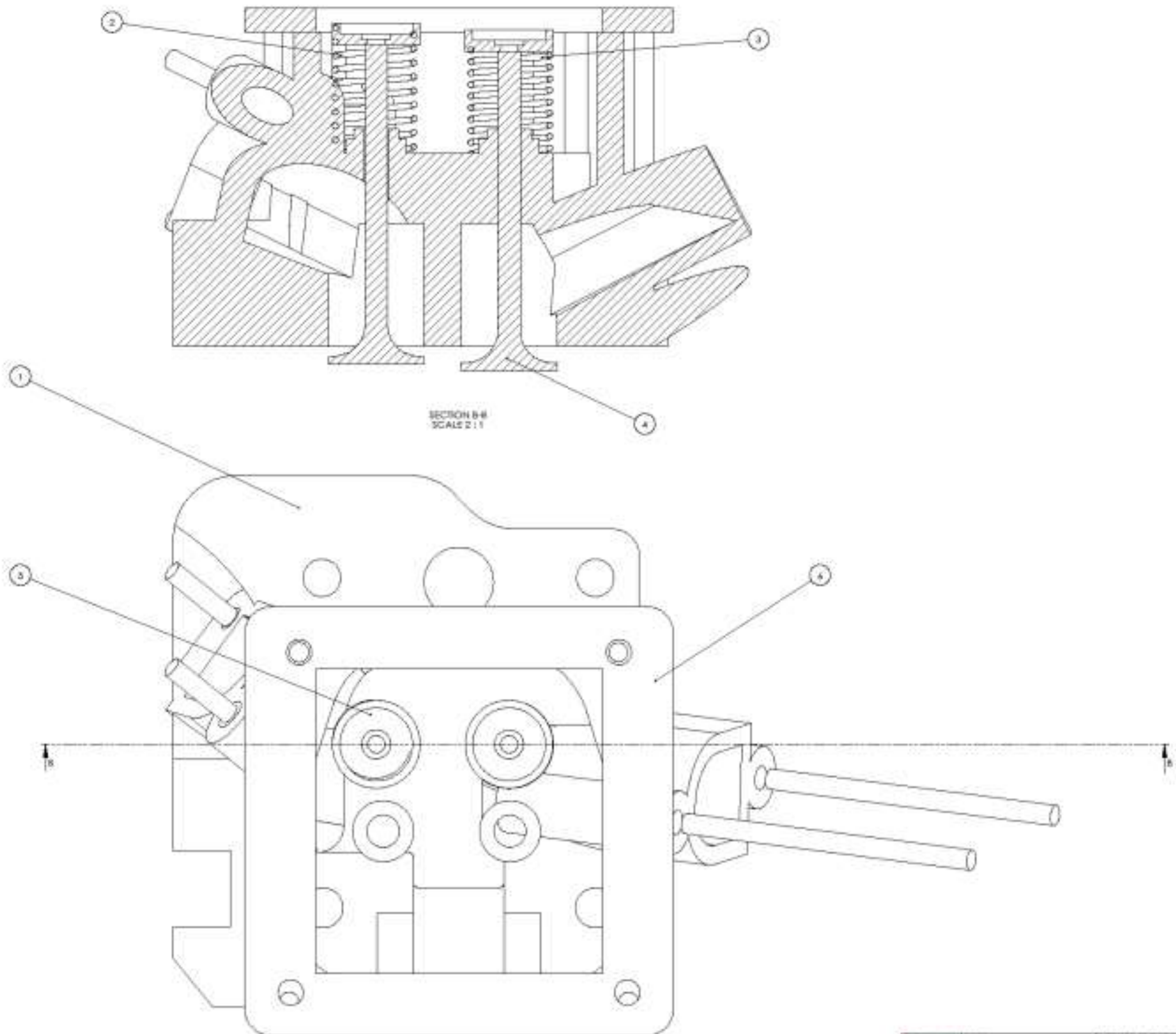




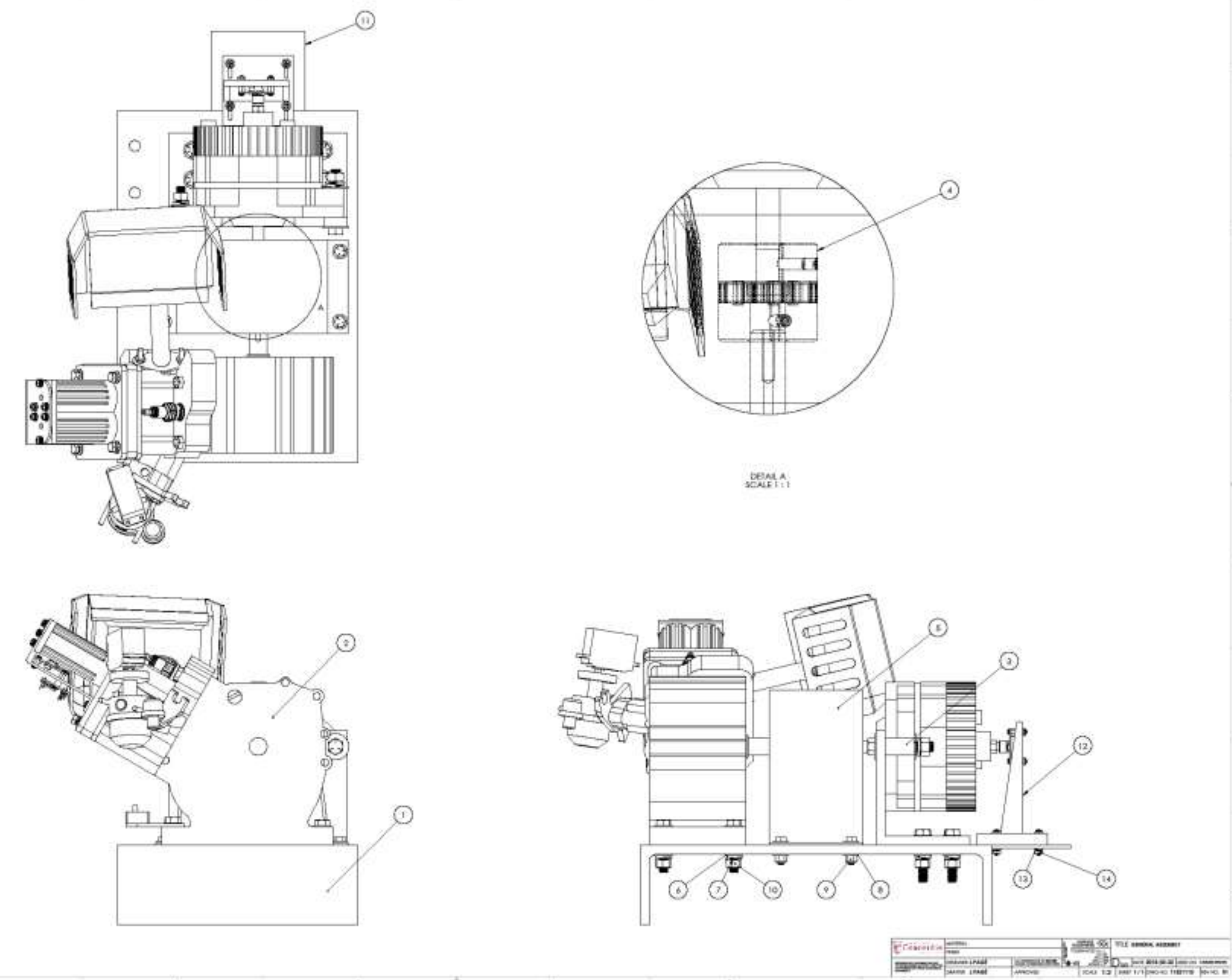
SECTION A-A

CONCORDIA UNIVERSITY	MATERIAL		SURFACE ROUGHNESS √XX	TITLE ROTARY MAGNET ASSEMBLY			
	FINISH		TOLERANCES .X ± .1 .XX ± .05 .XXX ± .005 .XXXX ± .000 ANGLE ± .5°				
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.BRACKEN	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE	DATE 2015-03-22	USED ON CAMLESS ENGINE	
	DRAFTER J.PAGÉ	APPROVED	SCALE 2:1	SHEET 1 / 1	DWG NO. 11521112		REV NO. 02

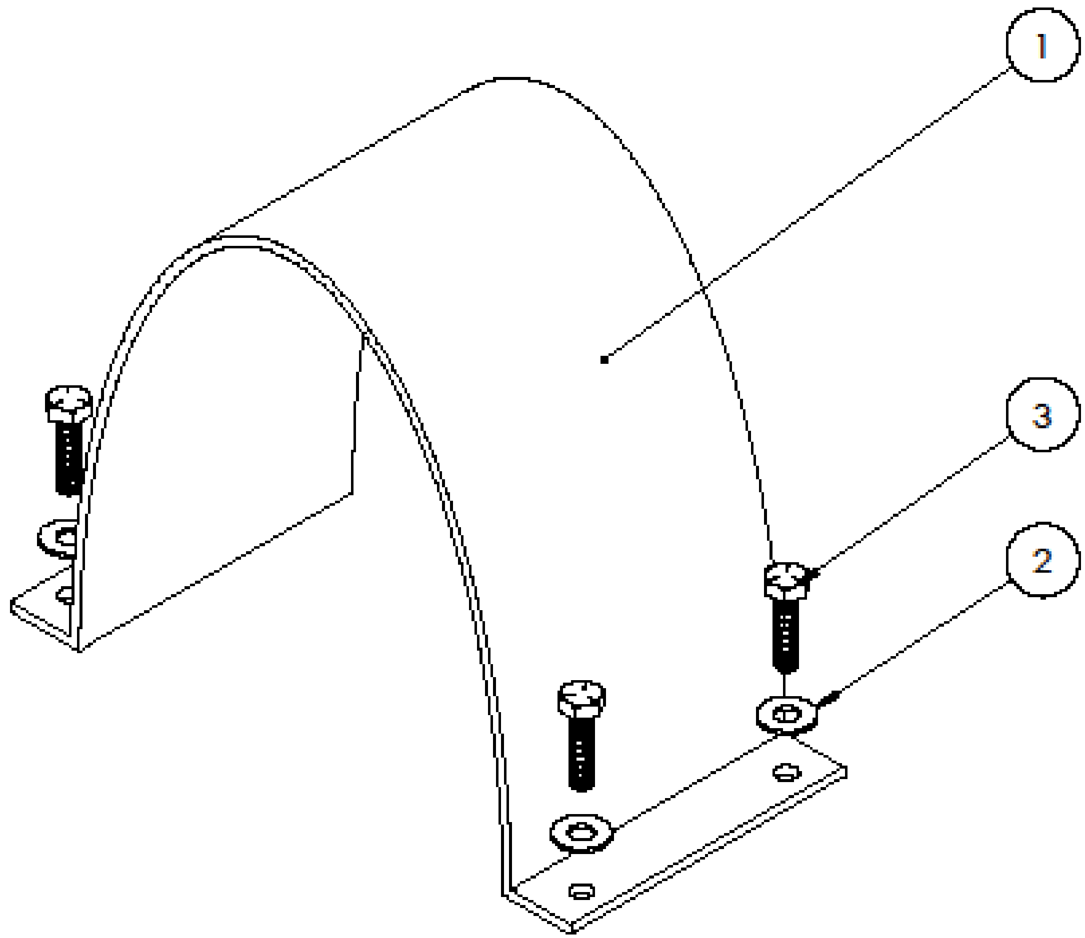
No. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	HEAD	FROM MANUFACTURER	1
2	LEFT SPRING		1
3	RIGHT SPRING		1
4	VALVE	FROM MANUFACTURER	2
5	SPRING CAP	FROM MANUFACTURER	2
6	11021200	5.5mm PRINTED SPACER	1



NO. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	11521213	MOUNT PLATE	1
2	11521107	ENGINE ASSEMBLY	1
3	11521106	ALTERNATOR BRACKET ASSEMBLY	1
4	COUPLER ASSEMBLY	LOVEJOY COUPLER	1
5	11521116	GUARD	1
6	90107A127	FLAT WASHER, 3/8"	8
7	92147A031	SPLIT LOCK WASHER, 3/8"	8
8	90107A029	FLAT WASHER, 1/4"	4
9	90715A125	NYLON-INSERT LOCKNUT, 1/4"-20	4
10	94804A320	HEX NUT, 3/8"-16	8
11	11521220	ROTARY SENSOR PLATE	1
12	11521111	ROTARY SENSOR BRACKET ASSEMBLY	1
13	96659A102	FLAT WASHER, NUMBER 6	8
14	90480A007	HEX NUT, 6-32	4



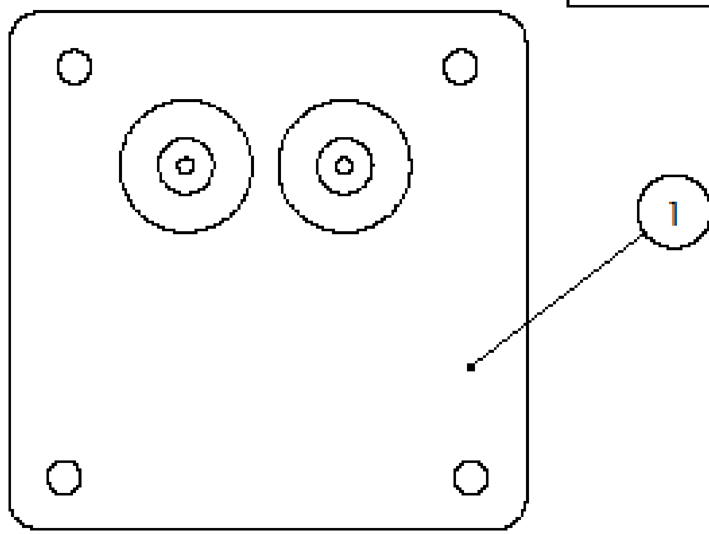
No. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	GUARD	1/8" THK BENT LEXAN SHEET	1
2	90107A029	FLAT WASHER, 1/4"	4
3	92865A542	CAP SCREW, 1/4"-20 1"LONG	4

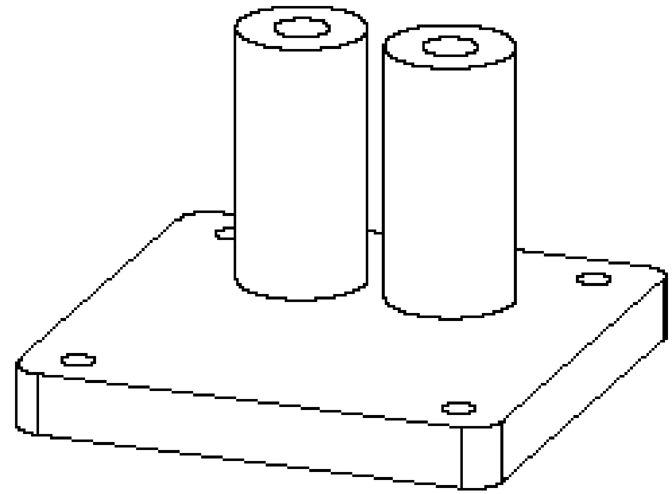


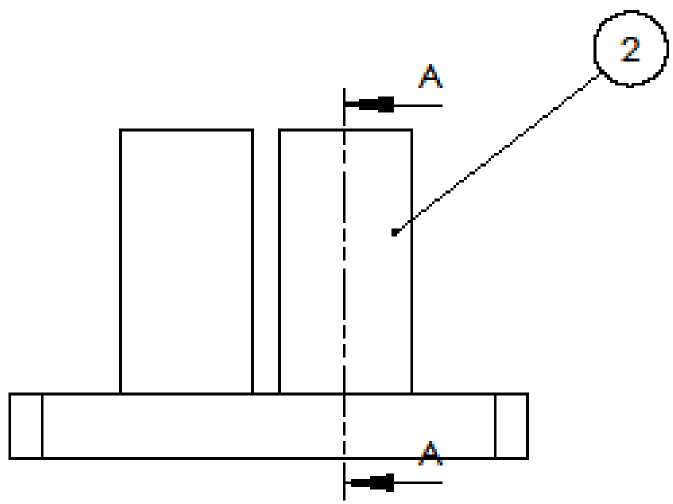
NOTE: BENT TO LEAVE SUFFICIENT CLEARANCE BETWEEN THE LEXAN SHIELD AND COUPLER

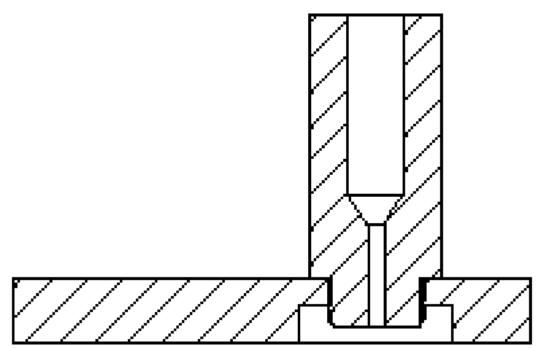
Concordia University	MATERIAL FINISH	SURFACE ROUGHNESS √63 TOLERANCES	TITLE GUARD ASSEMBLY
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ DRAFTER J.PAGÉ	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED APPROVED	A SIZE DATE 2015-03-22 USED ON CAMLESS ENGINE SCALE 1:2 SHEET 1 / 1 DWG NO. 11521216 REV NO. 01

No. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	11521211	THREADED SOLENOID PLATE	1
2	195207-224-A	SOLENOID BODY	2





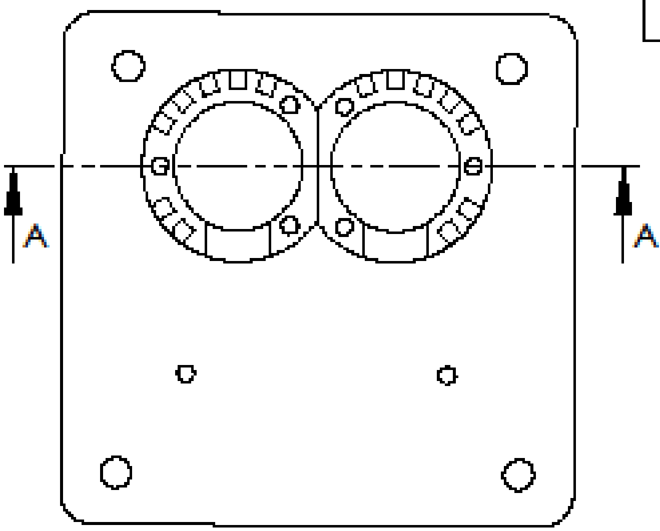


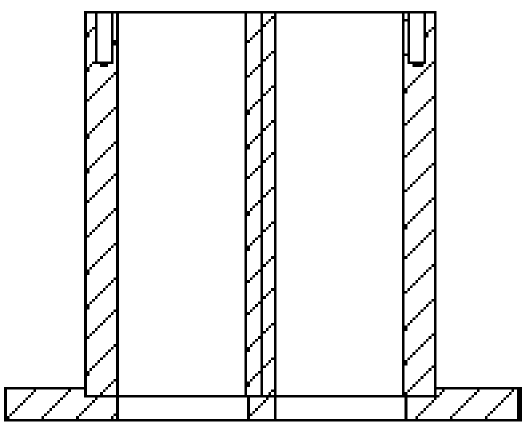


SECTION A-A
SCALE 2 : 3

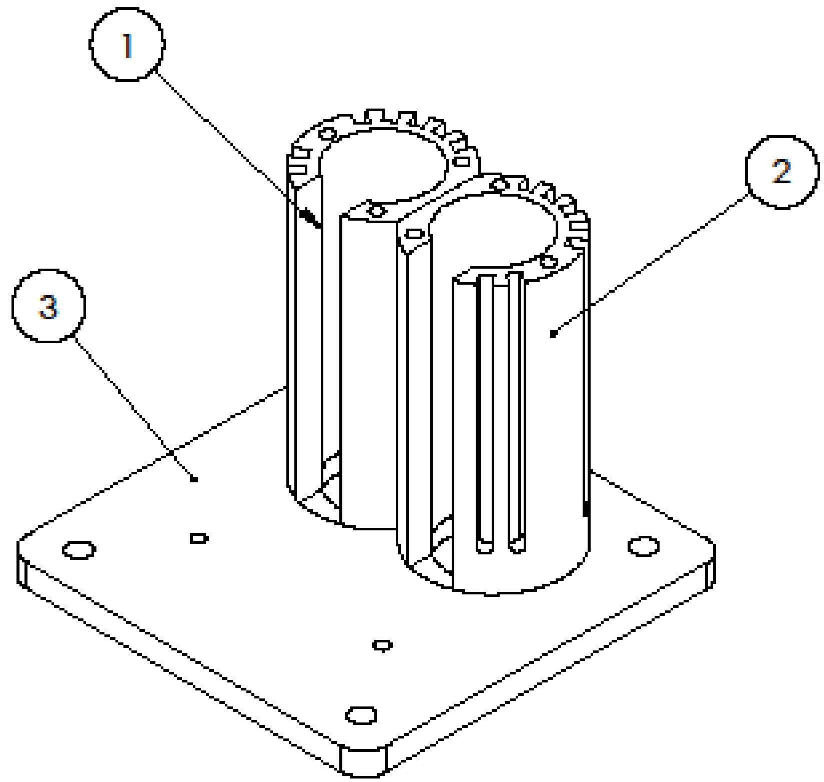
UNIVERSITÉ Concordia UNIVERSITY	MATERIAL	SURFACE ROUGHNESS √63	TITLE THREADED PLATE AND SOLENOIDS
	FINISH	TOLERANCES X ±.1, .X ±.05, .XX ±.005, .XXX ±.000, ANGLE ±.5°	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ / ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE / DATE 2015-03-22 / USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ / APPROVED	SCALE 2:3	SHEET 1 / 1 / DWG NO. 11521117 / REV NO. 01

No. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	11521208	LEFT SOLENOID TUBE	1
2	11521209	RIGHT SOLENOID TUBE	1
3	11521210	SOLENOID TUBE PLATE	1



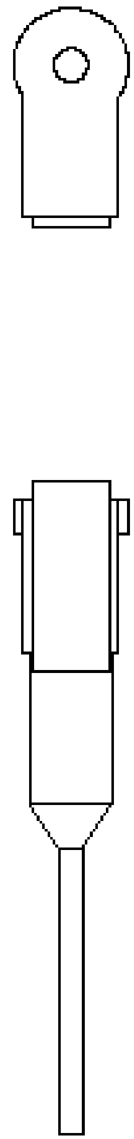


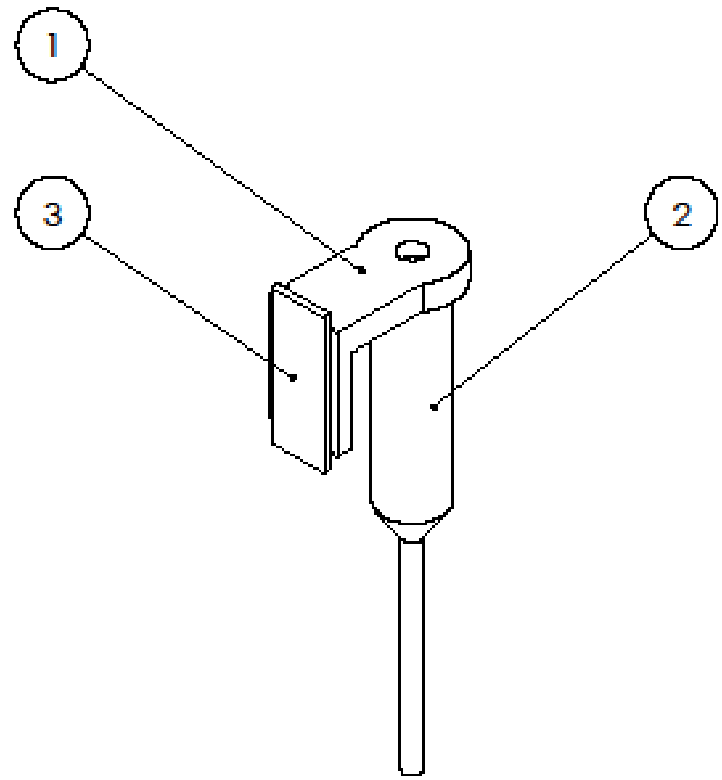
SECTION A-A
SCALE 2 : 3



Concordia University	MATERIAL		SURFACE ROUGHNESS √63	TITLE SOLENOID TUBE PLATE AND TUBES
	FINISH		TOLERANCES X ± .1 .X ± .05 .XX ± .005 .XXX ± .000 ANGLE ± .5°	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE · DATE 2015-03-22 · USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED	SCALE 2:3	SHEET 1 / 1 · DWG NO. 11521118 · REV NO. 01

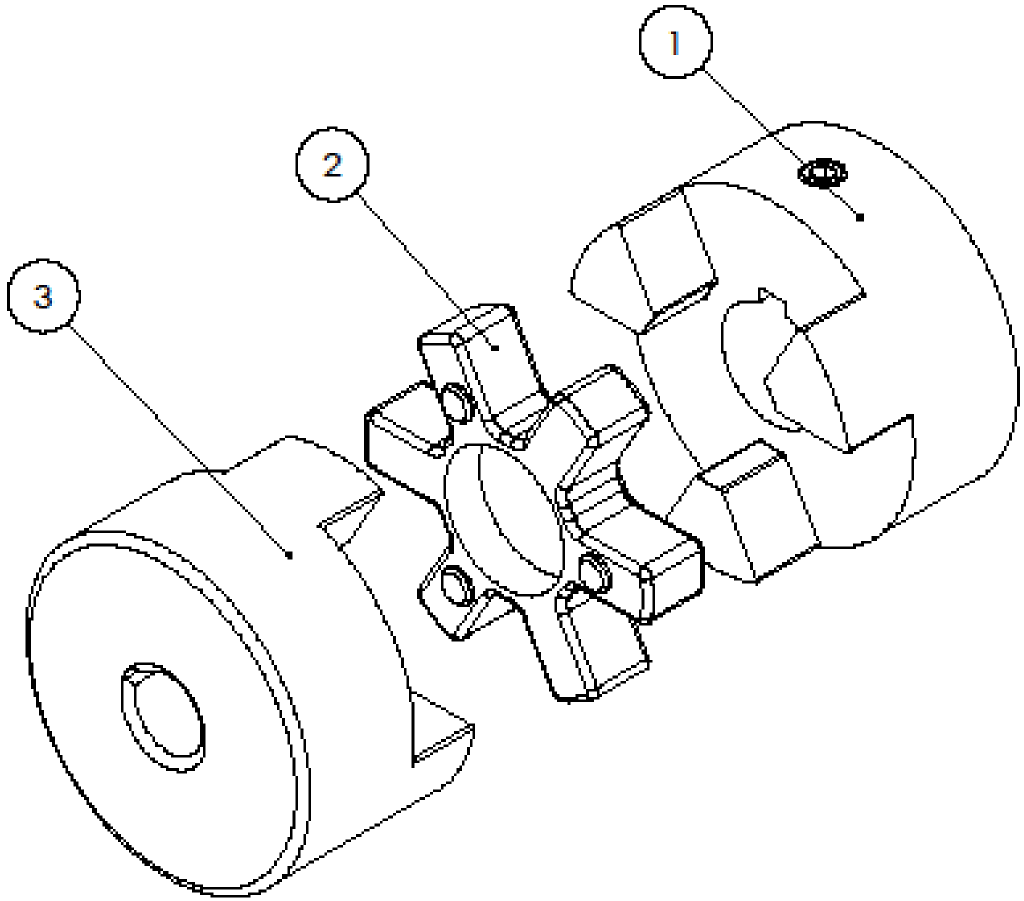
No. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	11521212	PLUNGER CAP	1
2	195207-224-B	SOLENOID PLUNGER WITH TAP	1
3	AS500-MS10-300	LINEAR MAGNET	1





Concordia University	MATERIAL	SURFACE ROUGHNESS √63	TITLE PLUNGER AND MAGNET CAP
	FINISH	TOLERANCES X ± .1, .X ± .05, .XX ± .005, .XXX ± .002, ANGLE ± .5°	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED	A SIZE — DATE 2015-03-22 — USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED — SCALE 1:1	SHEET 1 / 1 — DWG NO. 11521119 — REV NO. 01

No. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	68514476291	LOVEJOY L-090 .75 3/16 KEYWAY	1
2	RUBBER COUPLER	LOVEJOY L-090 RUBBER COUPLER	1
3	68514410769	LOVEJOY L-090 .5	1



Concordia University

PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY

MATERIAL

FINISH

DESIGNER J.PAGÉ

DRAFTER J.PAGÉ

ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED

APPROVED

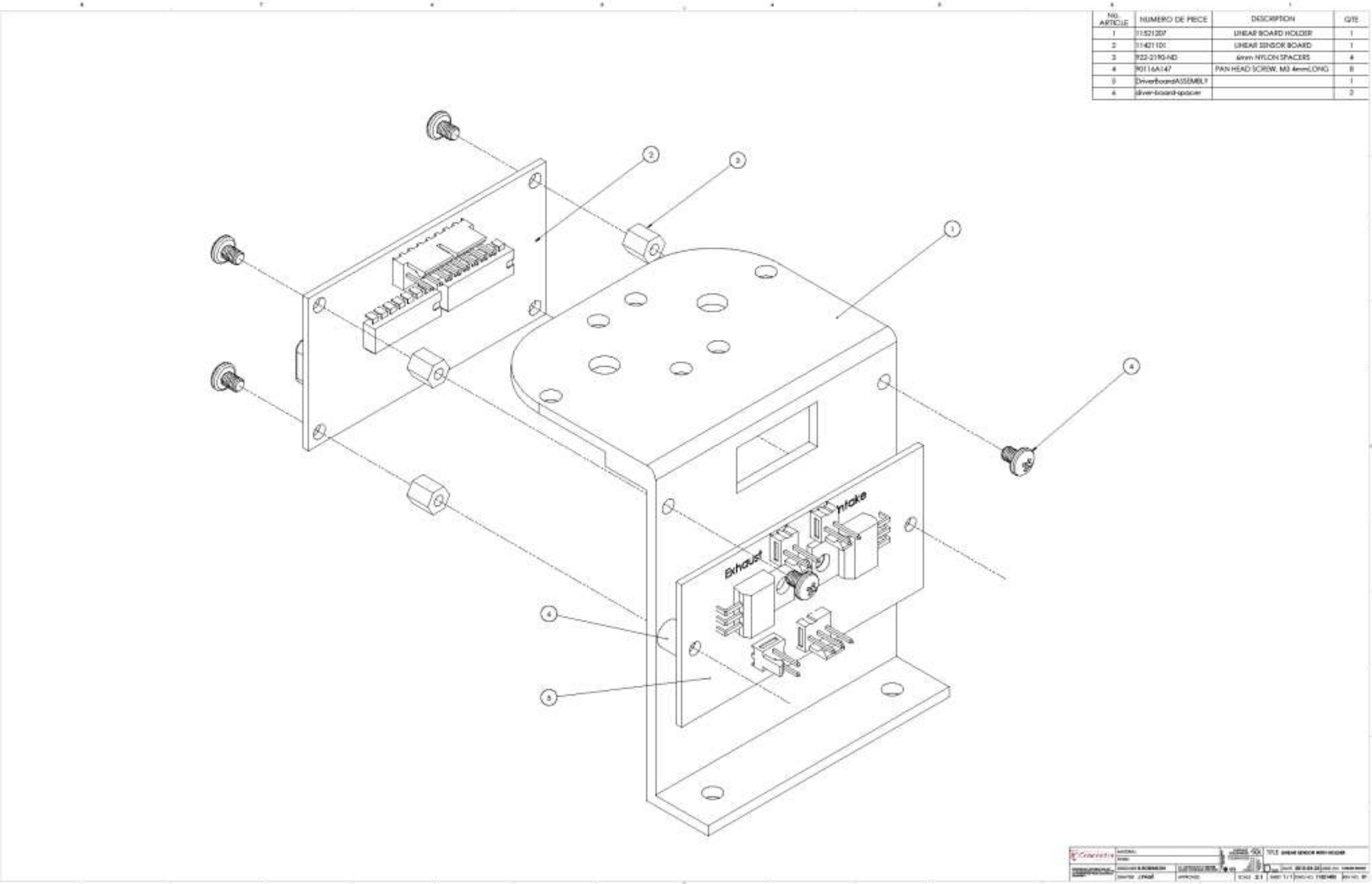
SURFACE ROUGHNESS √XX

TOLERANCES X ± .1, .X ± .05, .XX ± .005, .XXX ± .003, ANGLE ± .5°

TITLE COUPLER ASSEMBLY

A SIZE | DATE 2015-03-22 | USED ON CAMLESS ENGINE

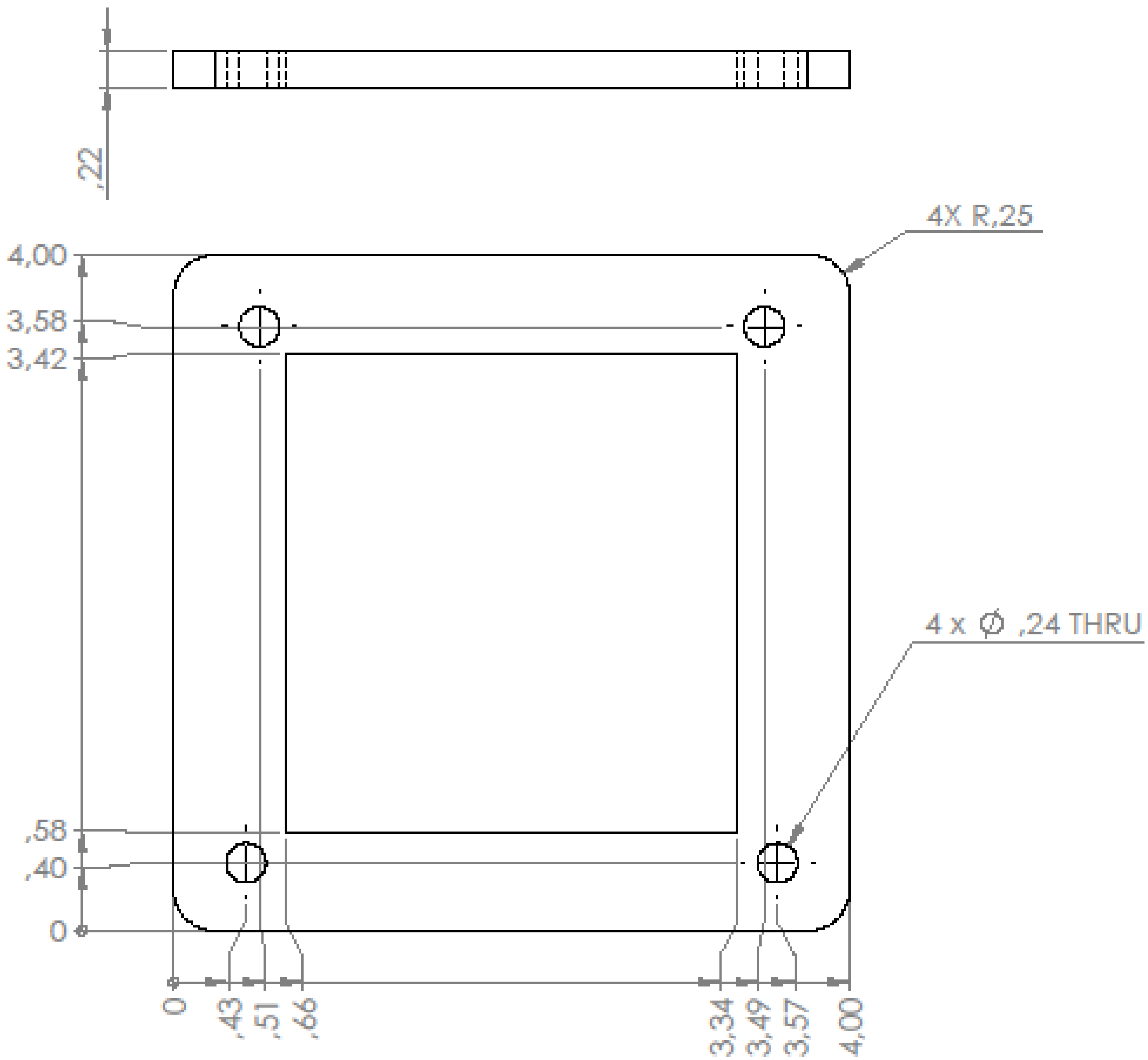
SCALE 1:1 | SHEET 1 / 1 | DWG NO. 11521116 | REV NO. 01



No. ARTICLE	NUMERO DE PIECE	DESCRIPTION	QTE
1	11521207	LINEAR BOARD HOLDER	1
2	11421101	LINEAR SENSOR BOARD	1
3	922-2190-ND	6mm NYLON SPACERS	4
4	90116A147	PAN HEAD SCREW, M3 4mmLONG	8
5	DriverBoardASSEMBLY		1
6	driver-board-spacer		2

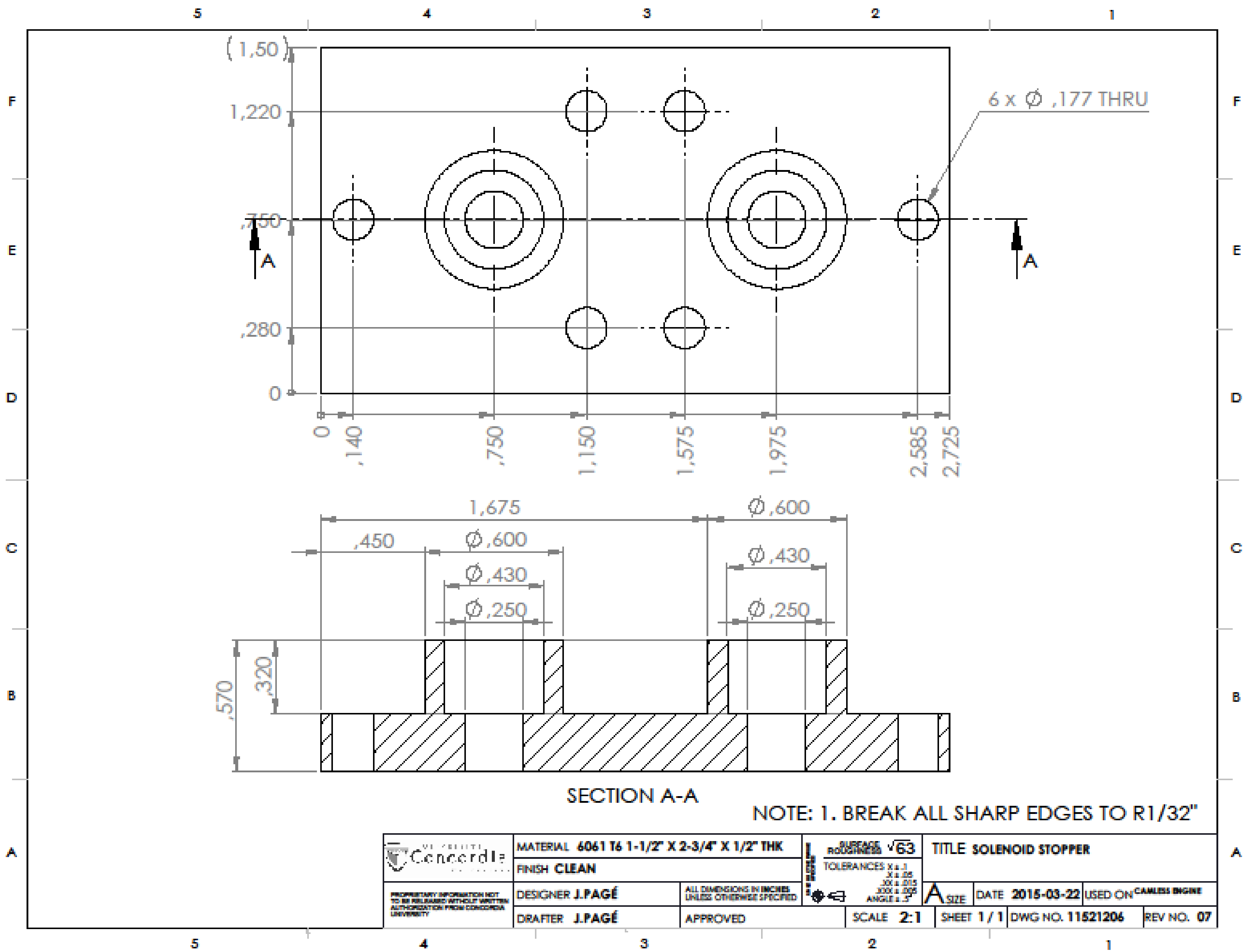
APPENDIX C : COMPONENT DRAWINGS

11521205	5.5mm PRINTED SPACER
11521206	SOLENOID STOPPER
11521207	PCB HOLDER
11521208	LEFT SOLENOID TUBE
11521209	RIGHT SOLENOID TUBE
11521210	SOLENOID TUBE PLATE
11521211	THREADED SOLENOID PLATE
11521212	PRINTED SOLENOID MAGNET CAP
11521213	MOUNT PLATE
11521214	MAP SENSOR PART
11521216	ROTARY SENSOR BRACKET BASE
11521217	ROTARY SENSOR BRACKET TOP
11521218	ROTARY SENSOR MAGNET HOLDER
11521219	SERVOMOTOR THROTTLE SHAFT
11521220	ROTARY SENSOR PLATE
11521222	ALTERNATOR BRACKET

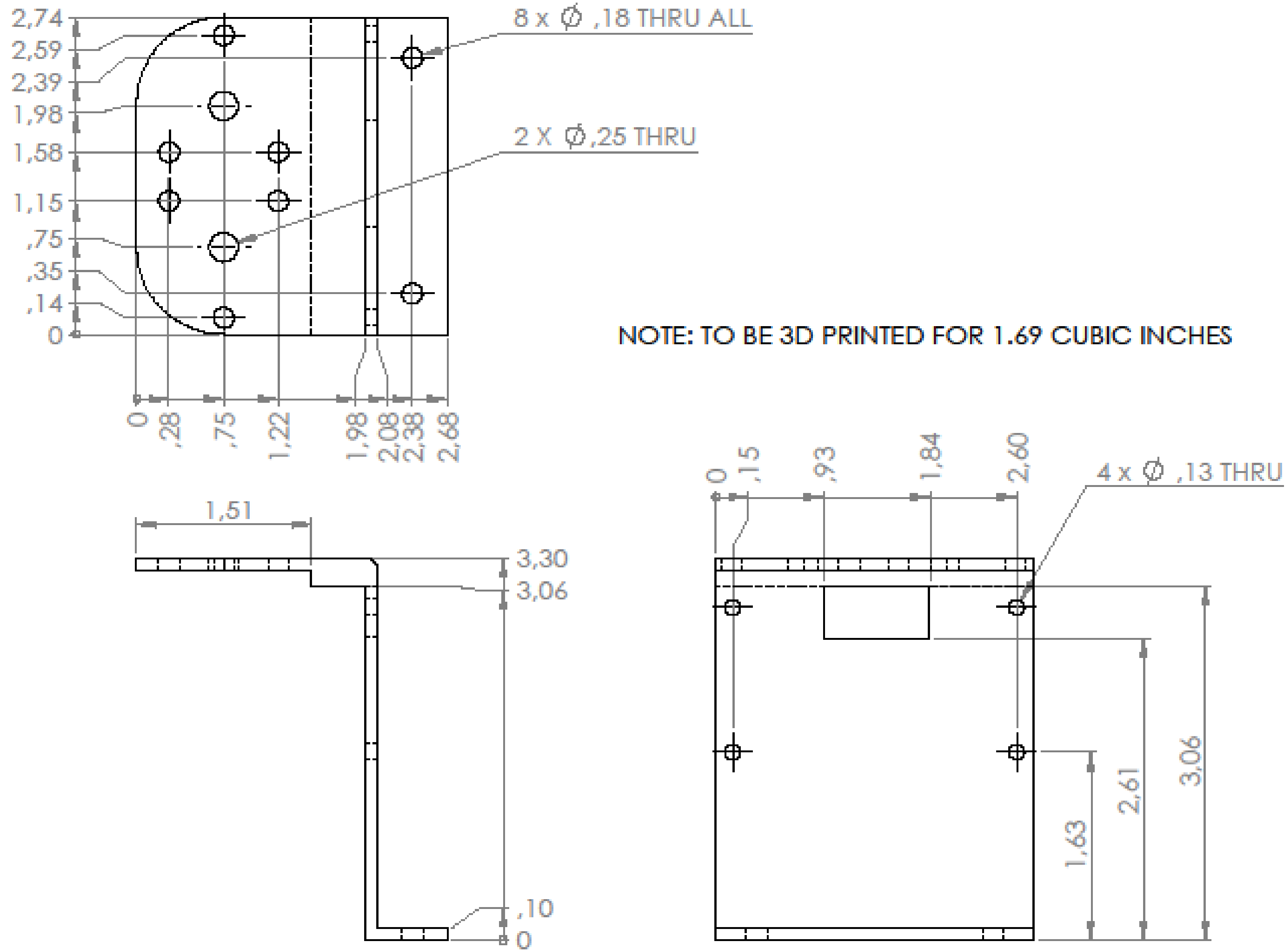


NOTE: 1. TO BE 3D PRINTED BY STUDENT

Université Concordia	MATERIAL ABS/PLASTIC	SURFACE ROUGHNESS √250	TITLE 5.5mm PRINTED SPACER
	FINISH	TOLERANCES ALL ± .01	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ — ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE — DATE 2015-03-22 — USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ — APPROVED	SCALE 1:1	SHEET 1 / 1 — DWG NO. 11521205 — REV NO. 01

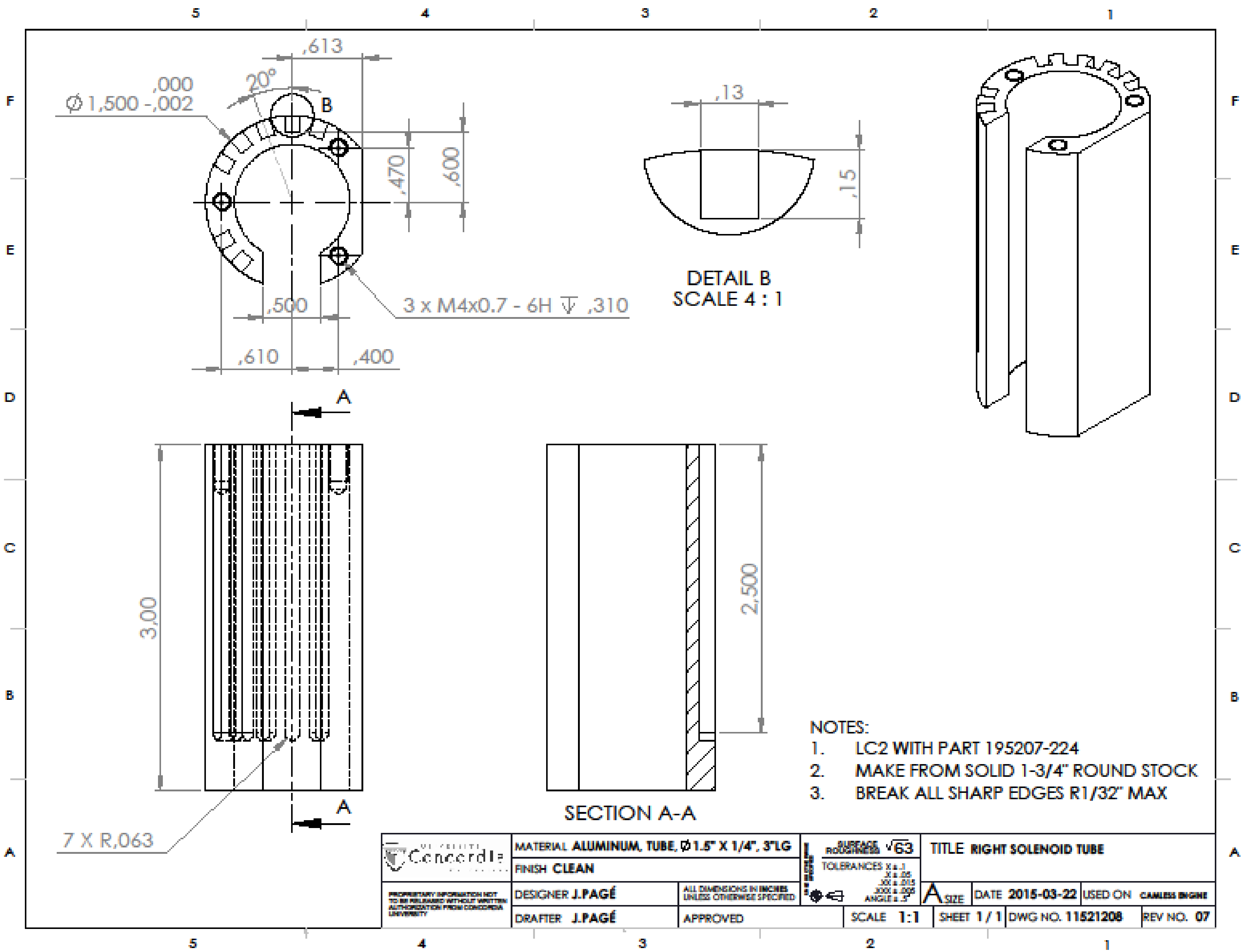


(1,50)
1,220
,750
,280
0
6 x Ø ,177 THRU
A
A
0
,140
,750
1,150
1,575
1,975
2,585
2,725
1,675
Ø ,600
,450
Ø ,600
Ø ,430
Ø ,430
Ø ,250
Ø ,250
,570
,320
SECTION A-A
NOTE: 1. BREAK ALL SHARP EDGES TO R1/32"
MATERIAL 6061 T6 1-1/2" X 2-3/4" X 1/2" THK
FINISH CLEAN
DESIGNER J.PAGÉ
DRAFTER J.PAGÉ
ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED
APPROVED
SURFACE ROUGHNESS √63
TOLERANCES
SCALE 2:1
TITLE SOLENOID STOPPER
A SIZE
DATE 2015-03-22
USED ON CAMLESS ENGINE
SHEET 1 / 1
DWG NO. 11521206
REV NO. 07

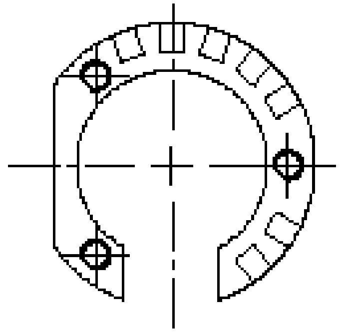


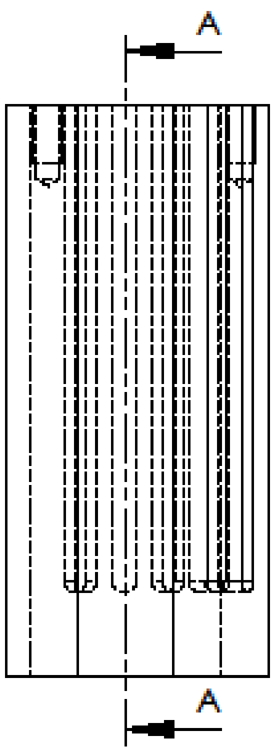
NOTE: TO BE 3D PRINTED FOR 1.69 CUBIC INCHES

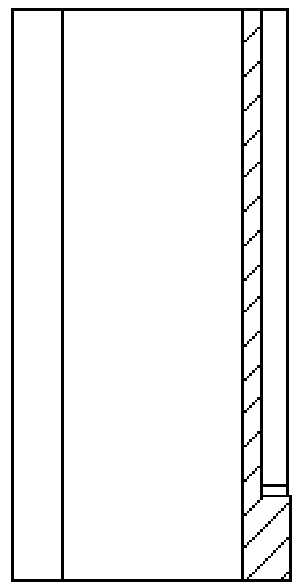
Concordia University	MATERIAL ABS/PLASTIC		SURFACE ROUGHNESS √250	TITLE PCB HOLDER
	FINISH		TOLERANCES ALL ± .01	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER B.ROBINSON	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE · DATE 2015-03-22 · USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED	SCALE 4:5	SHEET 1 / 1 · DWG NO. 11521207 · REV NO. 02



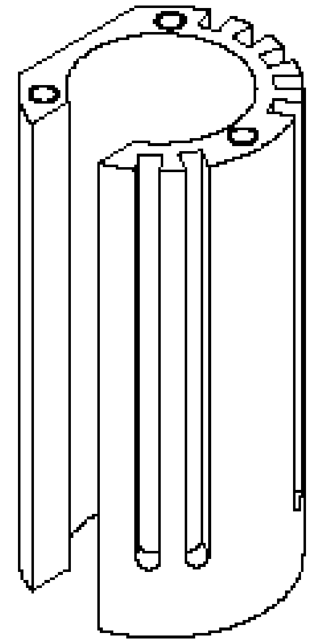
,613
20°
,000
Ø 1,500 -,002
B
,470
,600
,500
3 x M4x0.7 - 6H ↧ ,310
,610
,400
,13
,15
DETAIL B
SCALE 4 : 1
A
3,00
2,500
7 X R,063
SECTION A-A
NOTES:
1. LC2 WITH PART 195207-224
2. MAKE FROM SOLID 1-3/4" ROUND STOCK
3. BREAK ALL SHARP EDGES R1/32" MAX
MATERIAL ALUMINUM, TUBE, Ø 1.5" X 1/4", 3"LG
FINISH CLEAN
SURFACE ROUGHNESS √63
TITLE RIGHT SOLENOID TUBE
TOLERANCES
DESIGNER J.PAGÉ
ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED
A SIZE
DATE 2015-03-22
USED ON CAMLESS ENGINE
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY
DRAFTER J.PAGÉ
APPROVED
SCALE 1:1
SHEET 1 / 1
DWG NO. 11521208
REV NO. 07





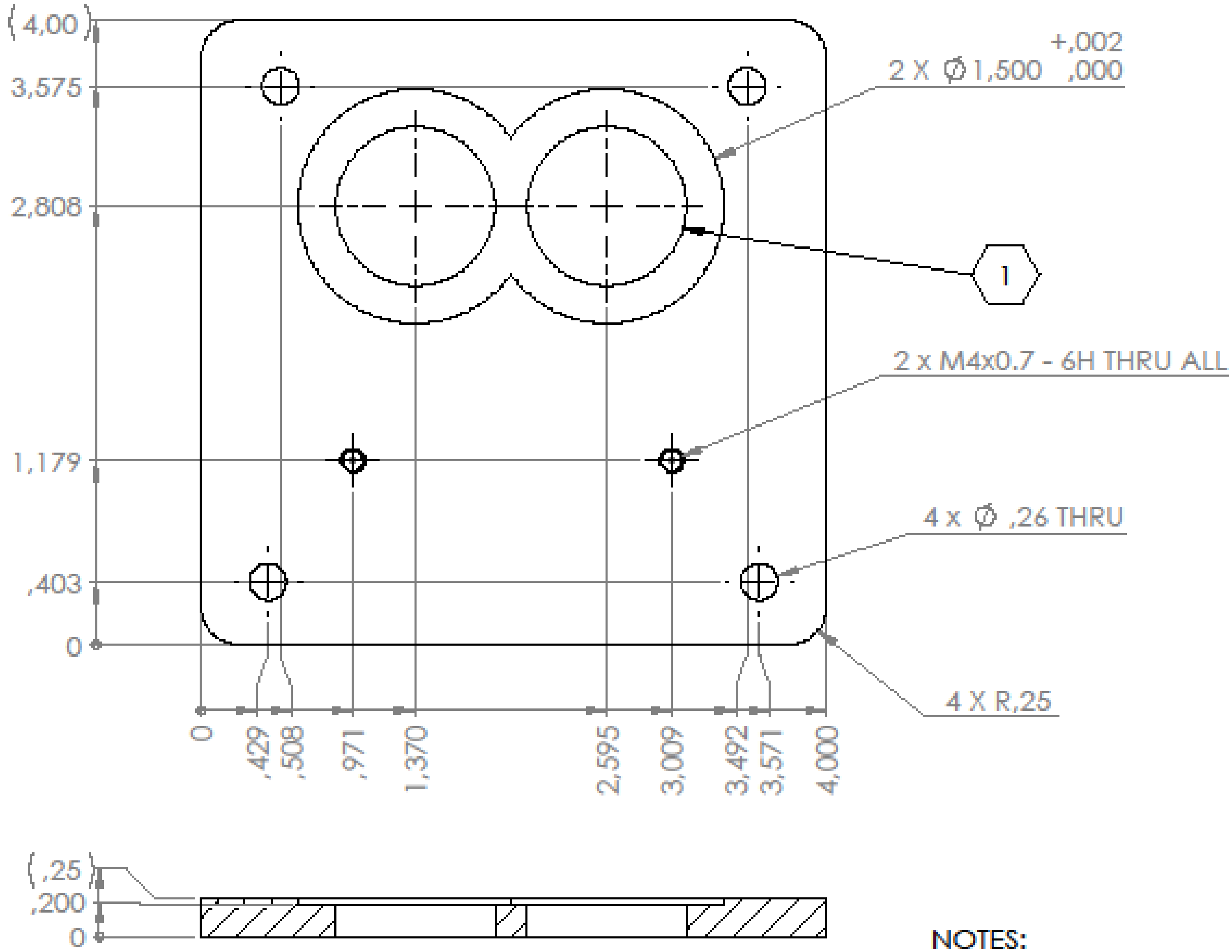


SECTION A-A



NOTE: ALL DIMENSIONS TO MIRROR THOSE OF 11521208

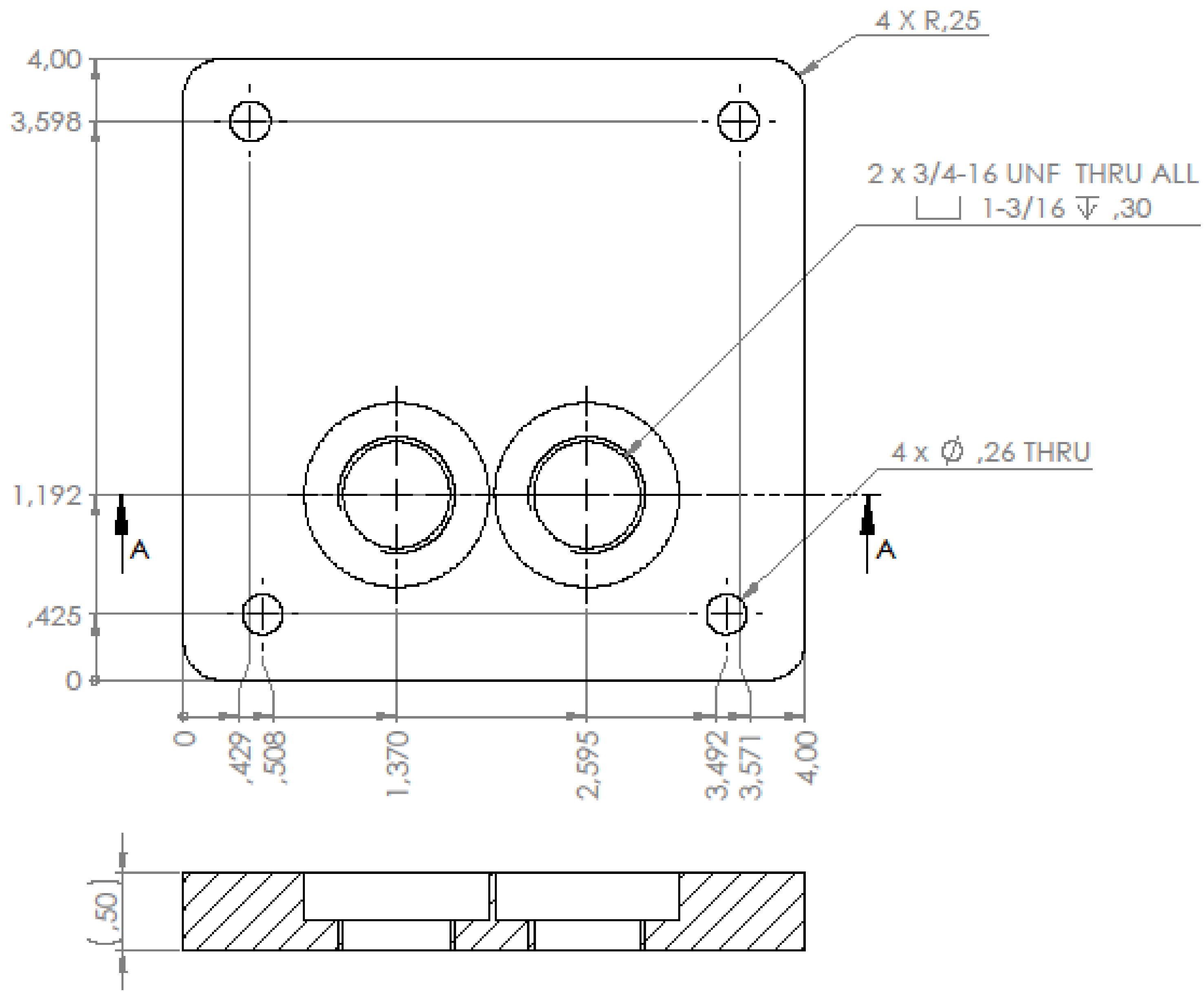
UNIVERSITE Concordia UNIVERSITY	MATERIAL **ALUMINUM, TUBE, Ø 1.5" X 1/4", 3"LG**		SURFACE ROUGHNESS √63	TITLE **LEFT SOLENOID TUBE**			
	FINISH **CLEAN**		TOLERANCES X ± .1 .X ± .05 .XX ± .015 .XXX ± .005 ANGLE ± .5°				
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER **J.PAGÉ**	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE	DATE **2015-03-22**	USED ON **CAMLESS ENGINE**	
	DRAFTER **J.PAGÉ**	APPROVED	SCALE **1:1**	SHEET **1 / 1**	DWG NO. **11521209**		REV NO. **07**



SECTION A-A
SCALE 1 : 1

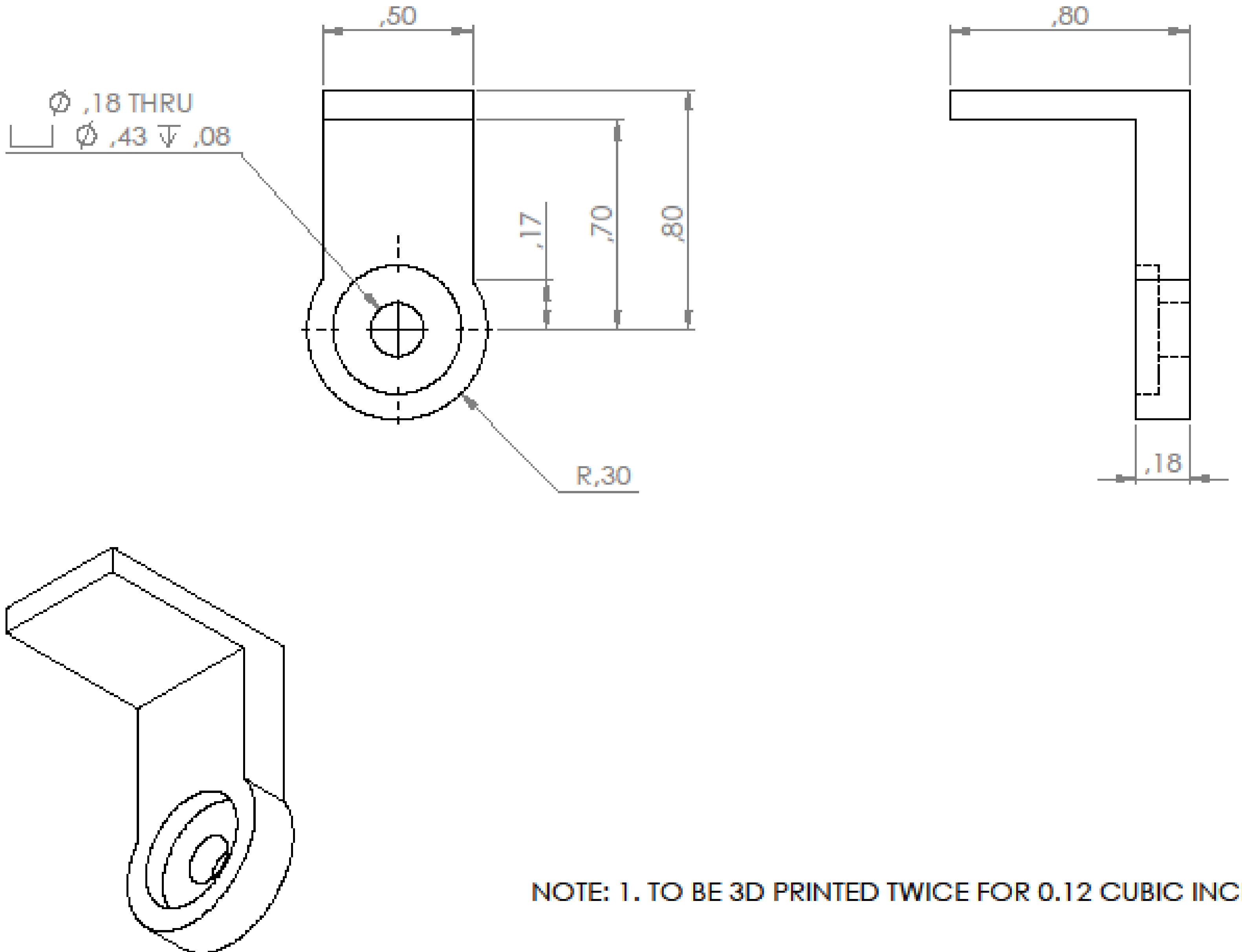
NOTES:
1. LC2 WITH PART 195207-224
2. BREAK ALL SHARP EDGES R1/32" MAX

Concordia University	MATERIAL 6061 T6 - 4" X 4" X 1/4" THK		SURFACE ROUGHNESS √63	TITLE SOLENOID TUBE PLATE
	FINISH CLEAN		TOLERANCES X ±.1 .X ±.05 .XX ±.015 .XXX ±.005 ANGLE ±.5°	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE · DATE 2015-03-22 · USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED F.LASTNAME	SCALE 1:1	SHEET 1 / 1 · DWG NO. 11521210 · REV NO. 07



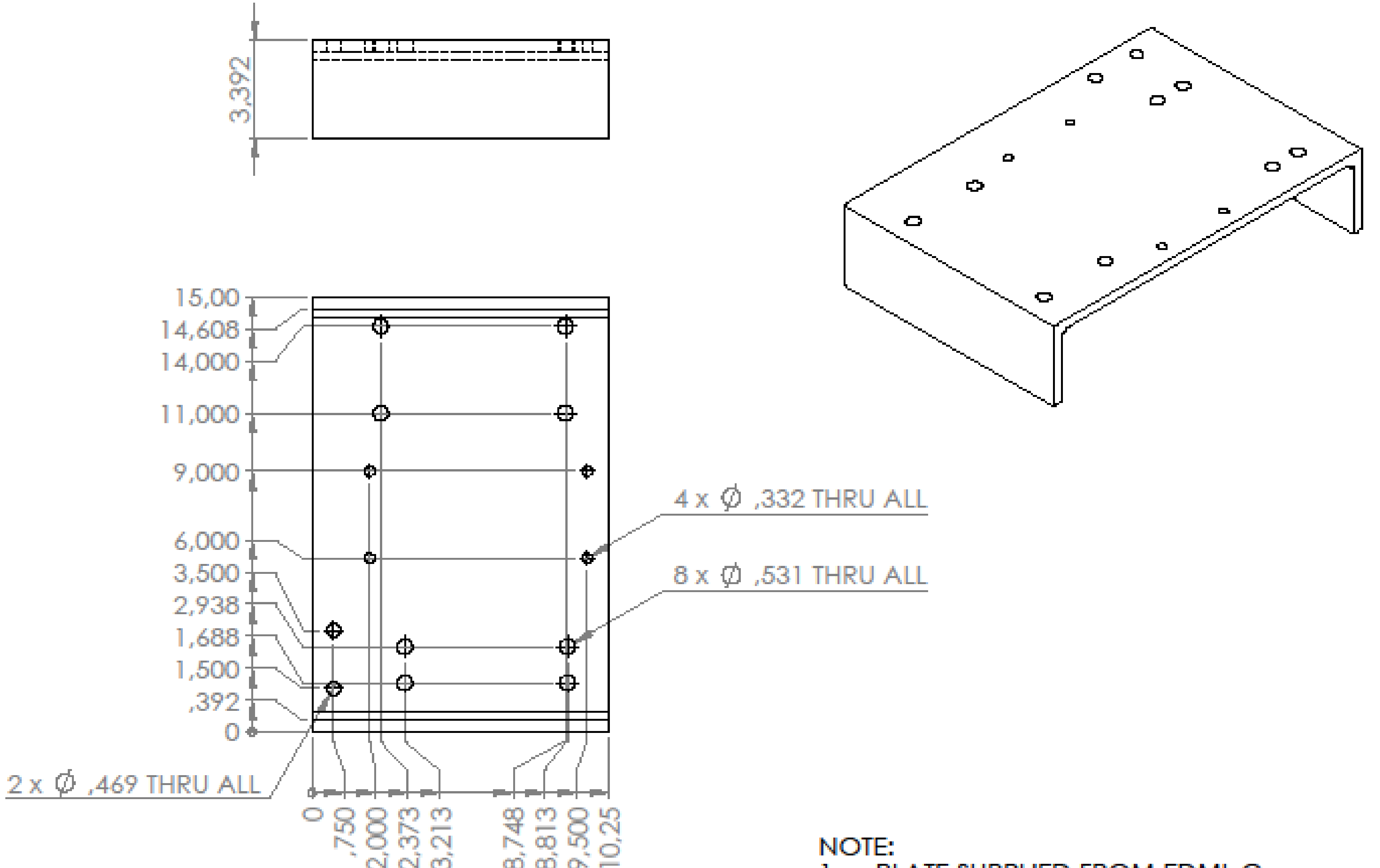
SECTION A-A

Concordia University	MATERIAL 6061 T6 - 4" X 4" X 1/2" THK		SURFACE ROUGHNESS √63	TITLE THREADED SOLENOID PLATE
	FINISH CLEAN		TOLERANCES X ± .1, .X ± .05, .XX ± .015, .XXX ± .005, ANGLE ± .5°	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED	A SIZE	DATE 2015-03-22 · USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED	SCALE 1:1	SHEET 1 / 1 · DWG NO. 11521211 · REV NO. 07



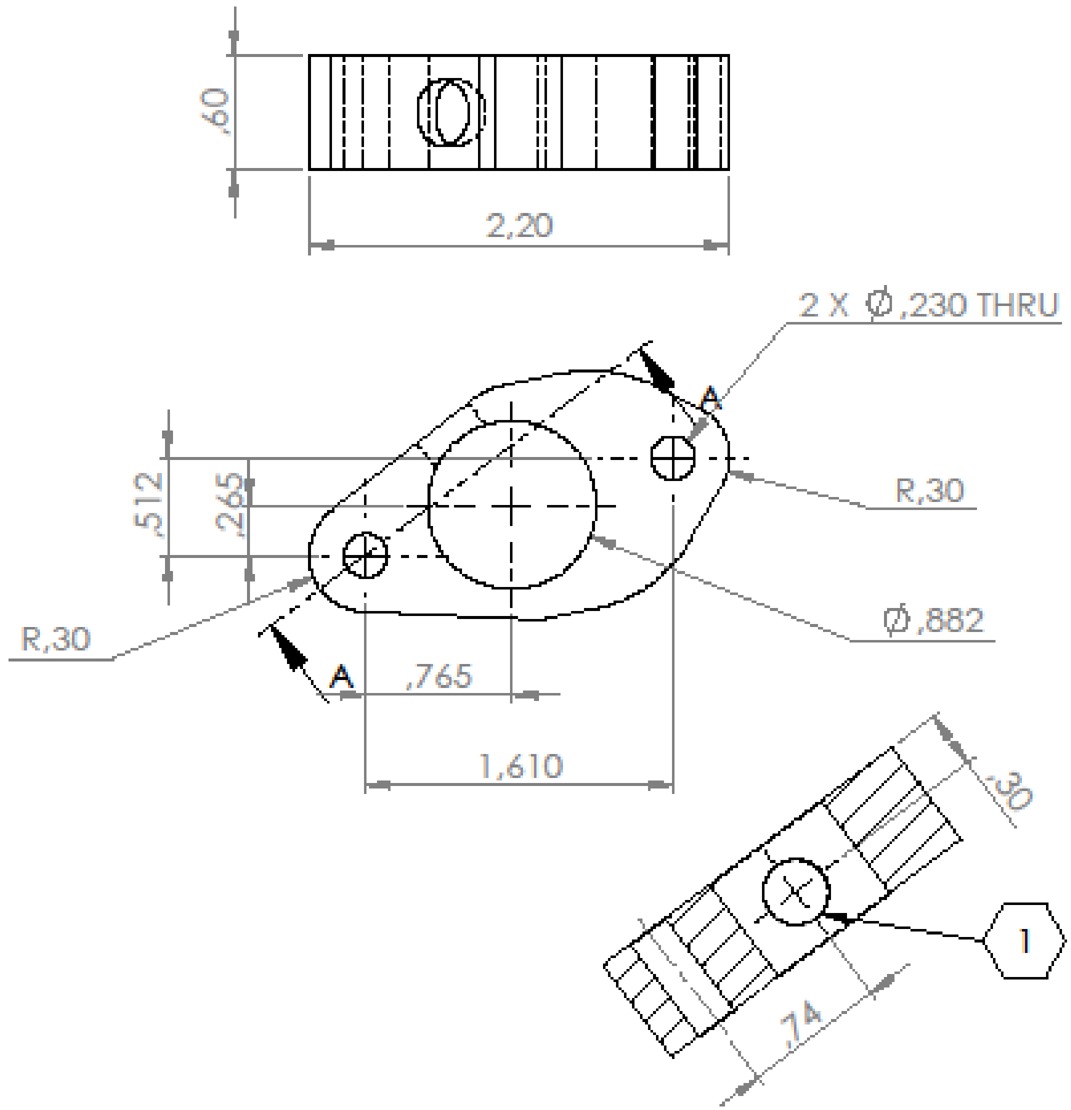
NOTE: 1. TO BE 3D PRINTED TWICE FOR 0.12 CUBIC INCHES PER PART

Concordia	MATERIAL PLASTIC/ABS		SURFACE ROUGHNESS √250 / TOLERANCES ALL ± .01	TITLE PRINTED SOLENOID MAGNET CAP
	FINISH			
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE / DATE 2015-03-22 / USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED	SCALE 2:1	SHEET 1 / 1 / DWG NO. 11521212 / REV NO. 01

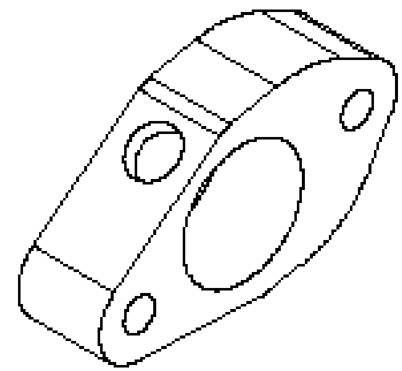


NOTE:
1. PLATE SUPPLIED FROM EDML C
2. BREAK ALL SHARP EDGES TO R1/32" MAX

Concordia	MATERIAL STEEL C-CHANNEL PLATE 10-1/4" X 15", 3" LEGS, 3/8" THK	SURFACE ROUGHNESS √63	TITLE MOUNT PLATE
	FINISH CLEAN	TOLERANCES X ± .1 .X ± .05 .XX ± .015 .XXX ± .005 ANGLE ± .5°	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ / ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE / DATE 2015-03-22 / USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ / APPROVED	SCALE 1:5	SHEET 1 / 1 / DWG NO. 11521213 / REV NO. 03

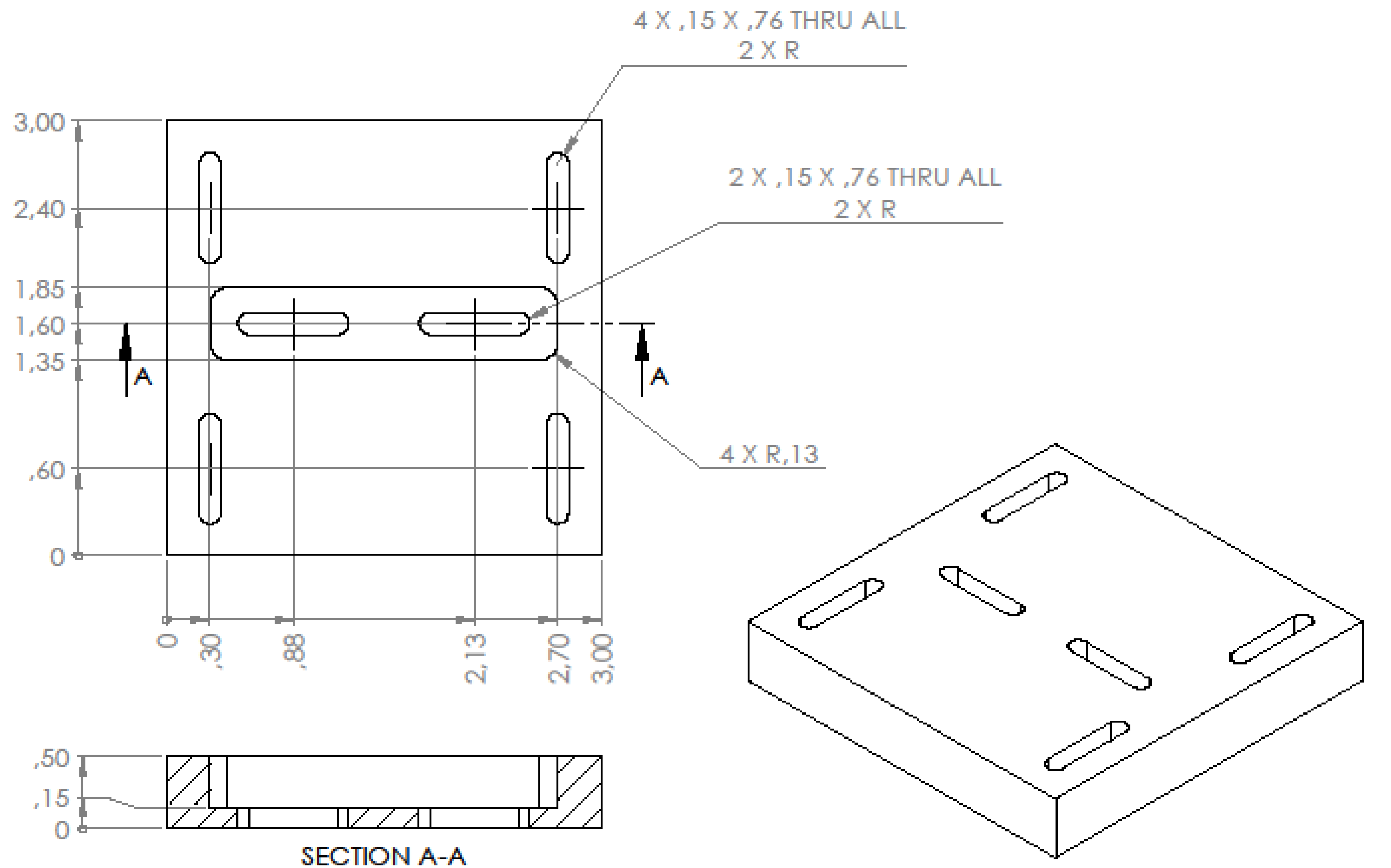


SECTION A-A



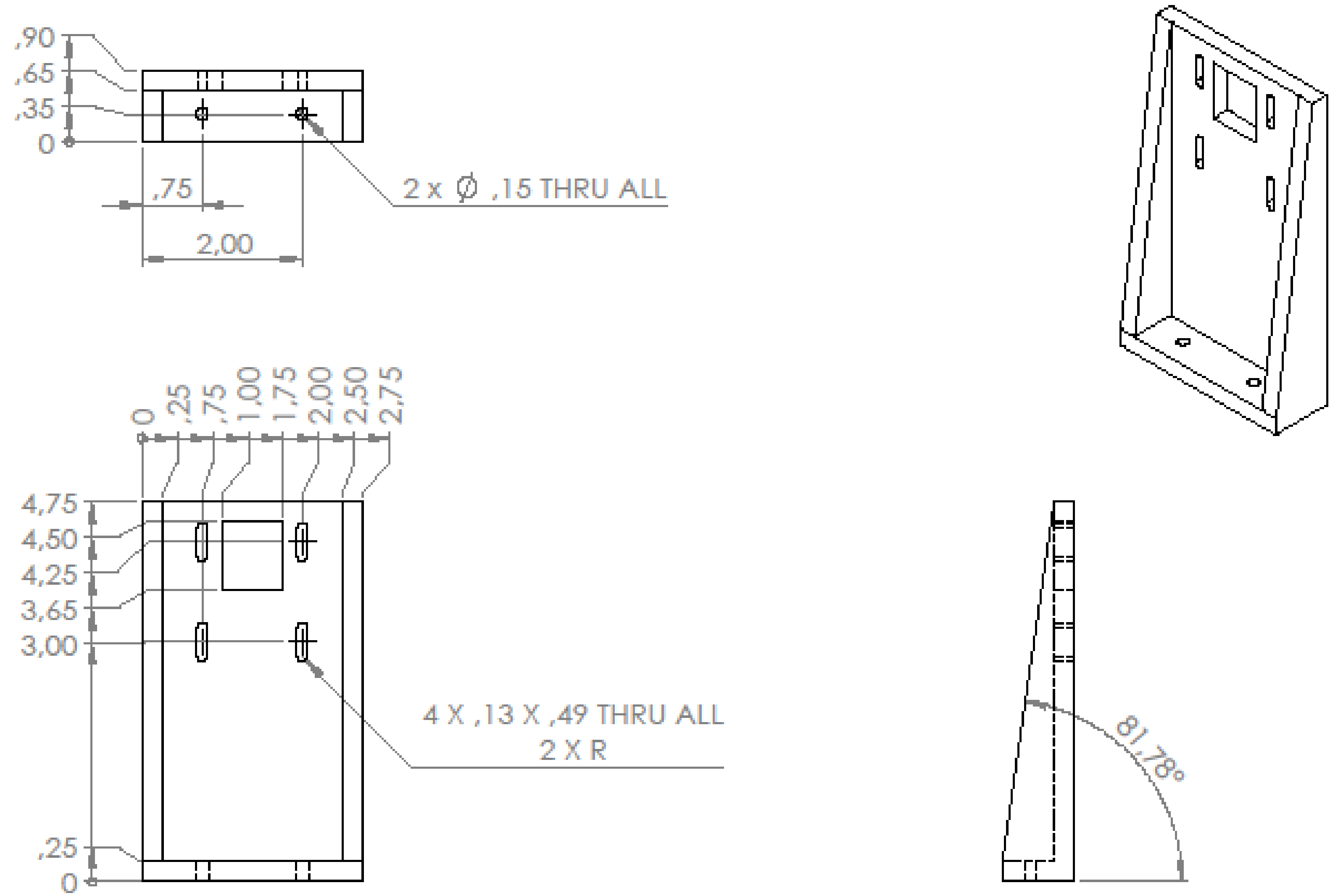
NOTES:
1. USE R DRILL SIZE FOR 1/8"-27 NPT
2. BANDSAW TO PROFILE
3. ROUND EDGES
4. SAND SMOOTH

Concordia University	MATERIAL 6061 T6		SURFACE ROUGHNESS √63	TITLE MAP SENSOR PART
	FINISH CLEAN		TOLERANCES X ± .1, .X ± .05, .XX ± .015, .XXX ± .005, ANGLE ± .5°	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.BRACKEN	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED	A SIZE	DATE 2015-03-22 · USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED	SCALE 1:1	SHEET 1 / 1 · DWG NO. 11521214 · REV NO. 01



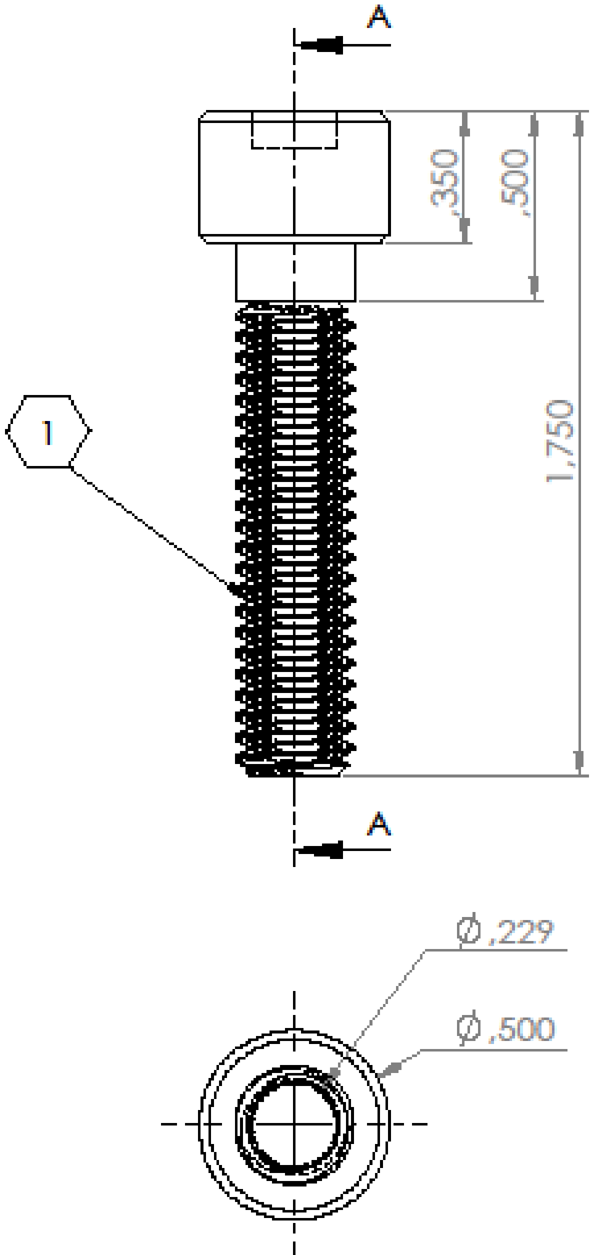
NOTE: TO BE 3D PRINTED FOR 3.83 CUBIC INCHES

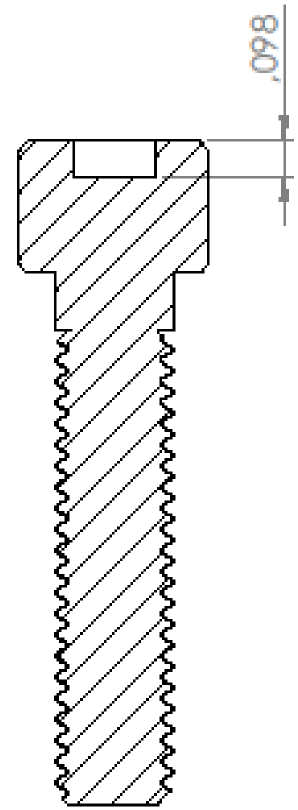
Concordia	MATERIAL PLASTIC/ABS	SURFACE ROUGHNESS √250	TITLE ROTARY SENSOR BRACKET BASE
	FINISH CLEAN	TOLERANCES ALL ± .01	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED	A SIZE — DATE 2015-03-22 — USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED — SCALE 1:1	SHEET 1 / 1 — DWG NO. 11521216 — REV NO. 04



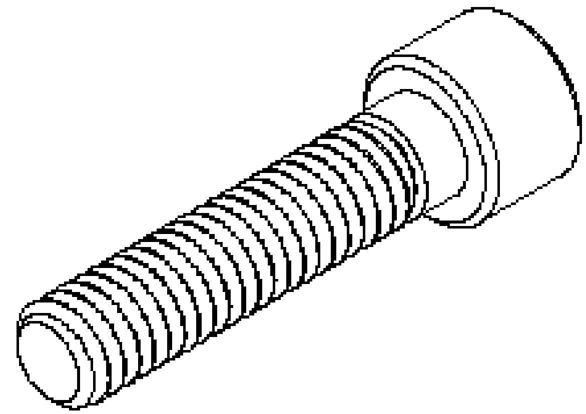
NOTE: 1. TO BE 3D PRINTED FOR 4.22 CUBIC INCHES

Concordia University	MATERIAL PLASTIC/ABS	SURFACE ROUGHNESS √250	TITLE ROTARY SENSOR BRACKET TOP
	FINISH CLEAN	TOLERANCES ALL ± .01	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ / ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED	A SIZE	DATE 2015-03-22 / USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ / APPROVED	SCALE 1:2	SHEET 1 / 1 / DWG NO. 11521217 / REV NO. 04





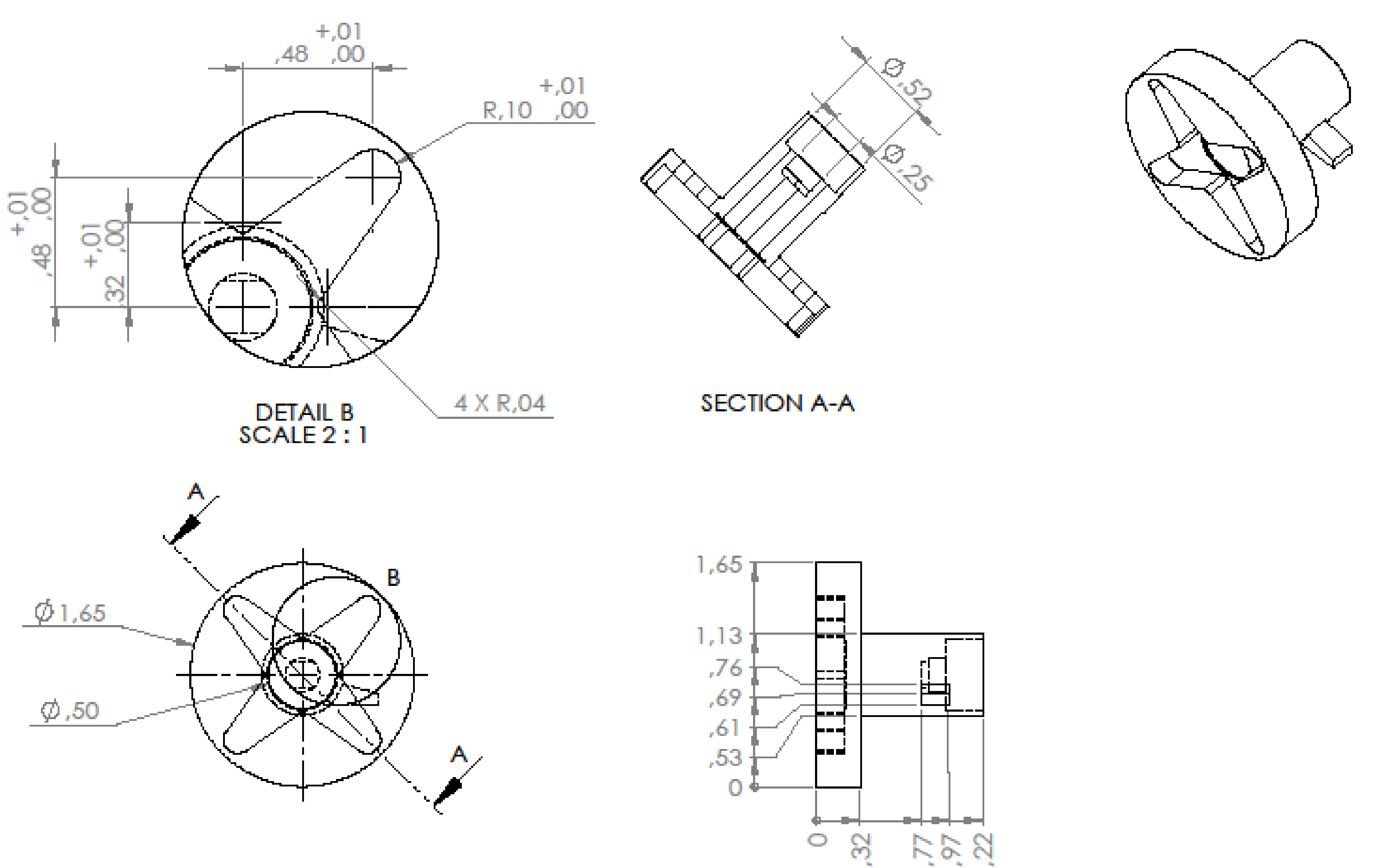
SECTION A-A



NOTES:

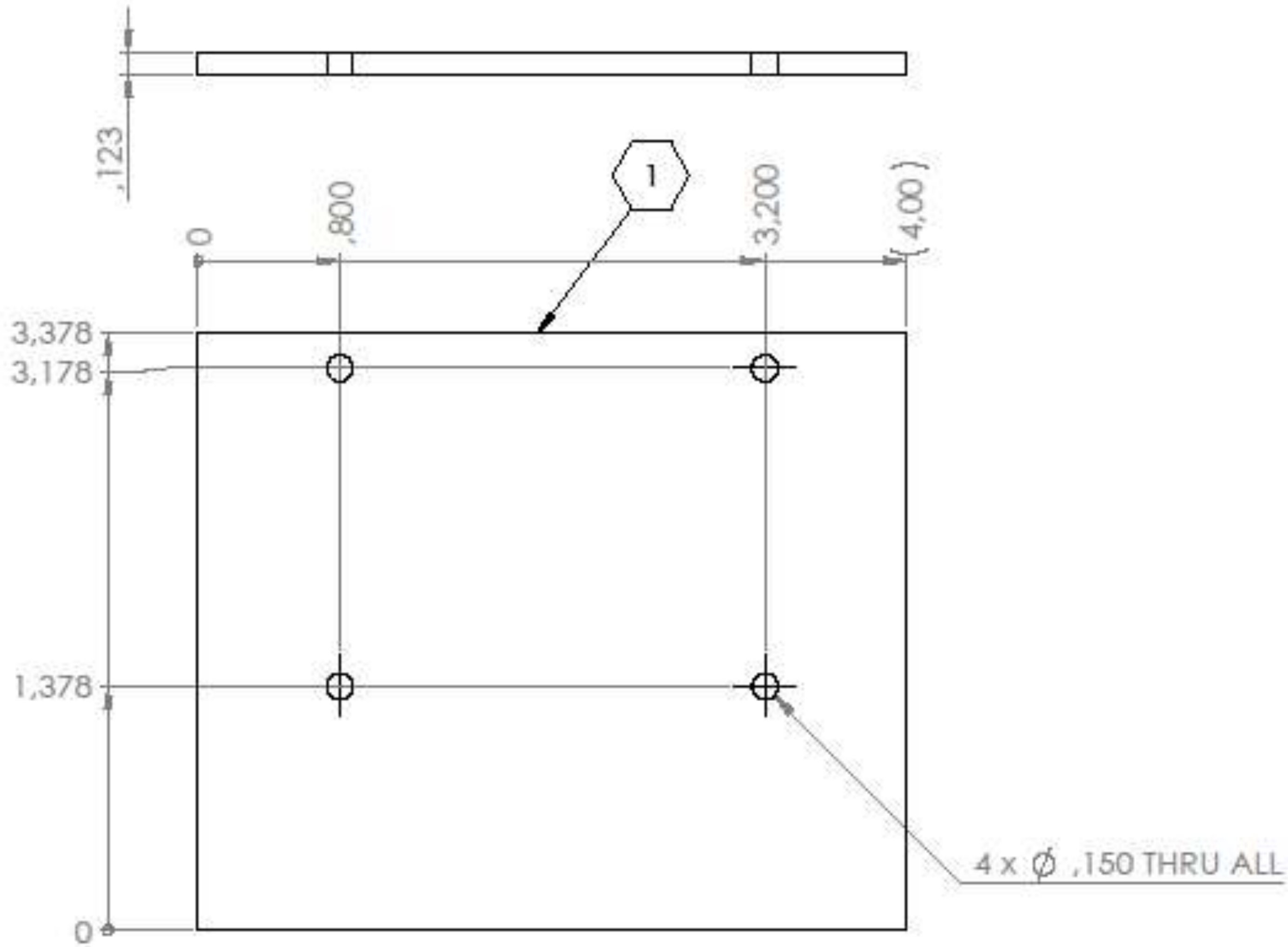
1. THREADED 5/16"-18
2. DEBUR LOCATION FOR ROTARY MAGNET FROM DRAWING 11521112
3. MACHINE PART FROM STOCK ALUMINUM ROD

Concordia University	MATERIAL GRADE 8 CAP SCREW		SURFACE ROUGHNESS √63	TITLE ROTARY MAGNET HOLDER
	FINISH CLEAN		TOLERANCES X ± .1, .X ± .05, .XX ± .015, .XXX ± .005, ANGLE ± .5°	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.BRACKEN	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE / DATE 2015-03-22 / USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED	SCALE 2:1	SHEET 1 / 1 / DWG NO. 11521218 / REV NO. 02



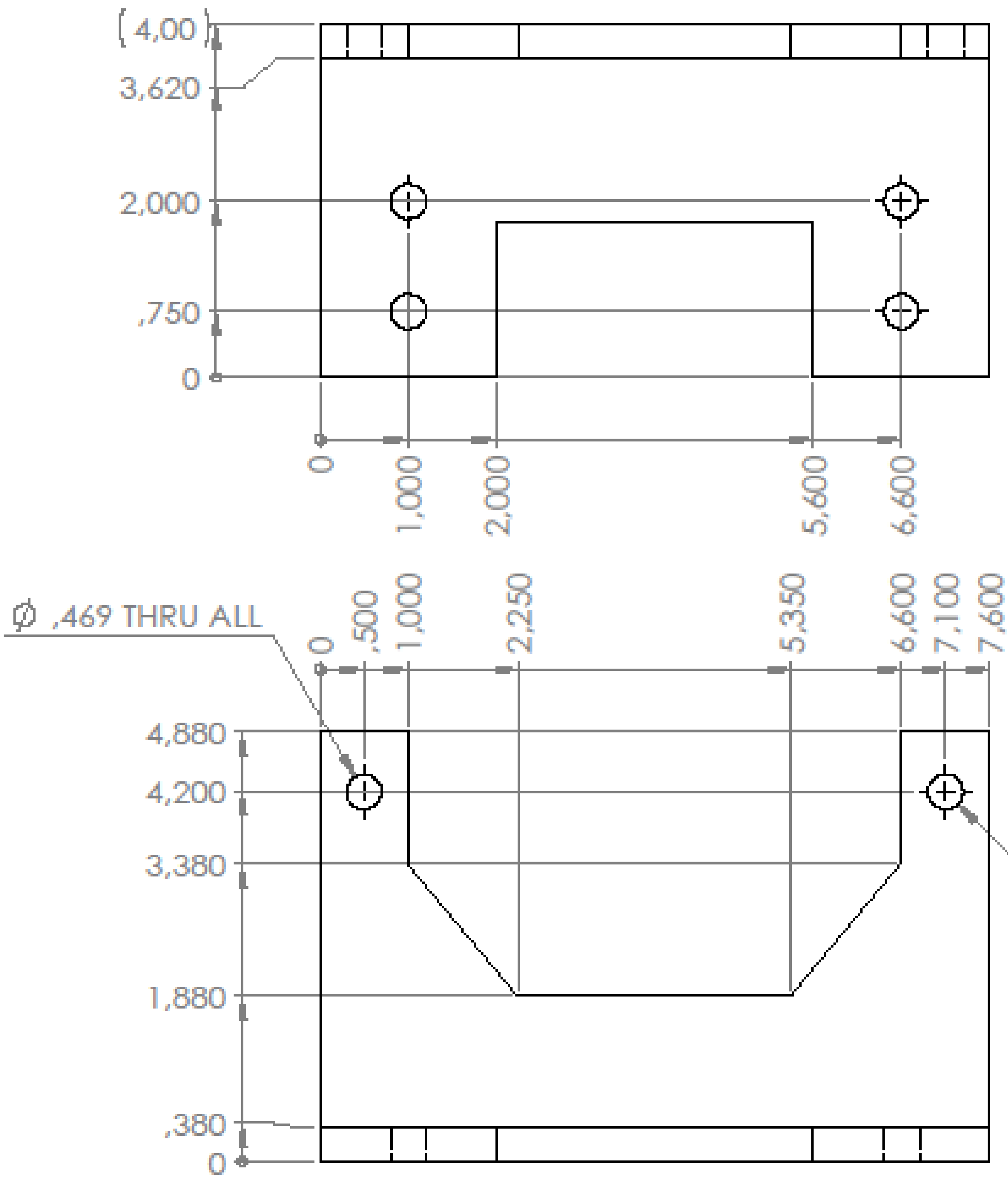
NOTE: TO BE 3D PRINTED BY STUDENT

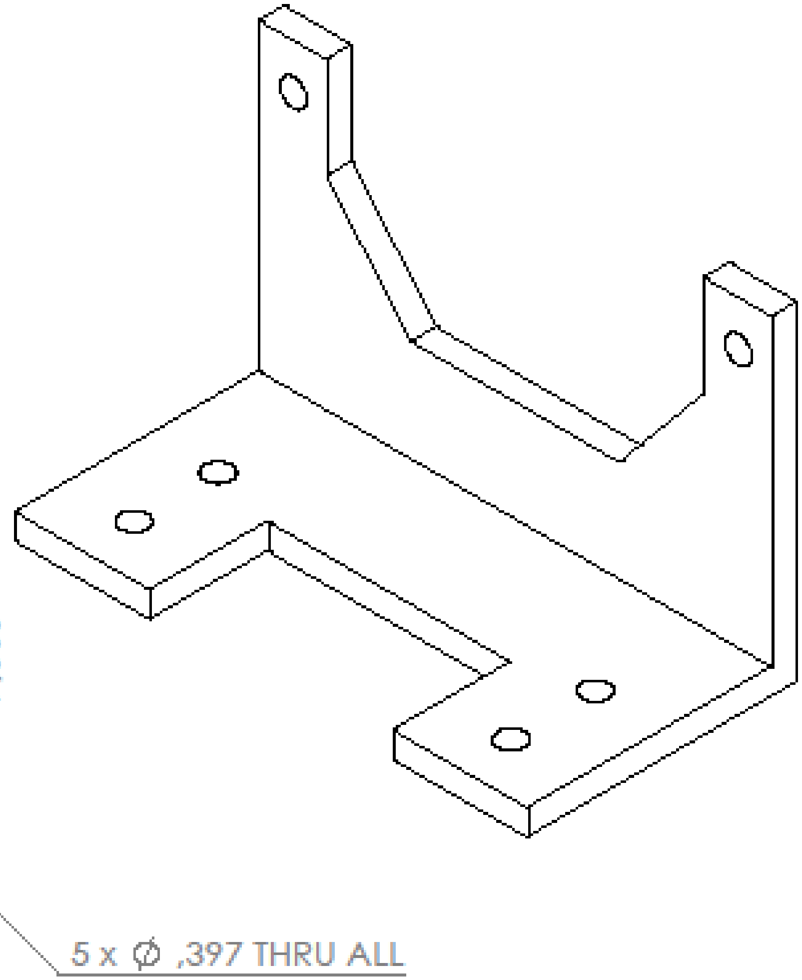
Concordia University	MATERIAL PLASTIC/ABS		SURFACE ROUGHNESS √250 TOLERANCES ALL ± ,01	TITLE SERVOMOTOR THROTTLE SHAFT
	FINISH			
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.BRACKEN	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE · DATE 2015-03-22 · USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED	SCALE 1:1	SHEET 1 / 1 · DWG NO. 11521219 · REV NO. 01



NOTES:
1. TO BE WELDED TO 11521213
2. BREAK ALL SHARP EDGES TO R1/32" MAX

Concordia	MATERIAL STEEL PLATE 4" X 4" X 1/8" THK		SURFACE ROUGHNESS √63	TITLE ROTARY SENSOR PLATE
	FINISH CLEAN		TOLERANCES X ± .1, .X ± .05, .XX ± .015, .XXX ± .005, ANGLE ± .5°	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE · DATE 2015-03-22 · USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED	SCALE 1:1	SHEET 1 / 1 · DWG NO. 11521220 · REV NO. 01





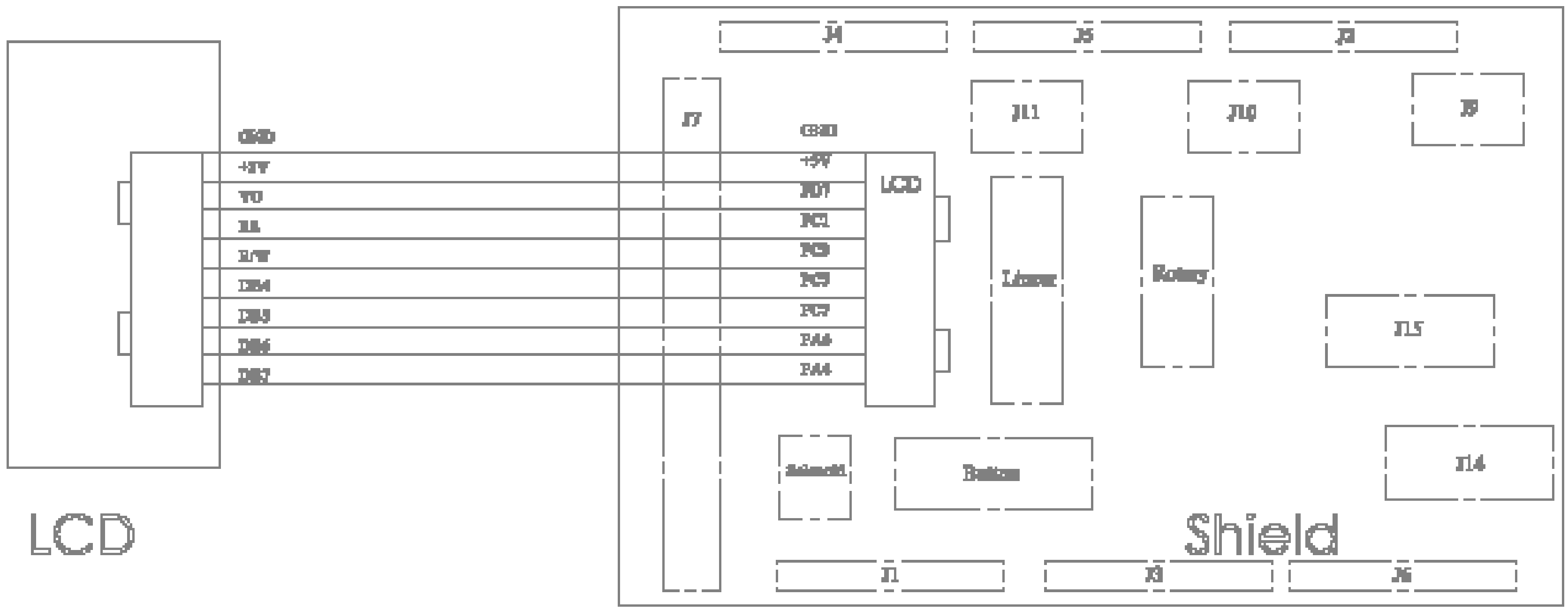
NOTES:

1. TO MACHINE FROM ANGLE IRON 4" X 6" LEGS
2. BREAK ALL SHARP EDGES TO R1/32" MAX

Concordia University	MATERIAL ANGLE IRON 4" X 6" LEGS		SURFACE ROUGHNESS √63	TITLE ALTERNATOR BRACKET
	FINISH CLEAN		TOLERANCES X ± .1 .X ± .05 .XX ± .015 .XXX ± .005 ANGLE ± .5°	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER J.PAGÉ	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE · DATE 2015-03-22 · USED ON CAMLESS ENGINE
	DRAFTER J.PAGÉ	APPROVED	SCALE 1:2	SHEET 1 / 1 · DWG NO. 11521222 · REV NO. 04

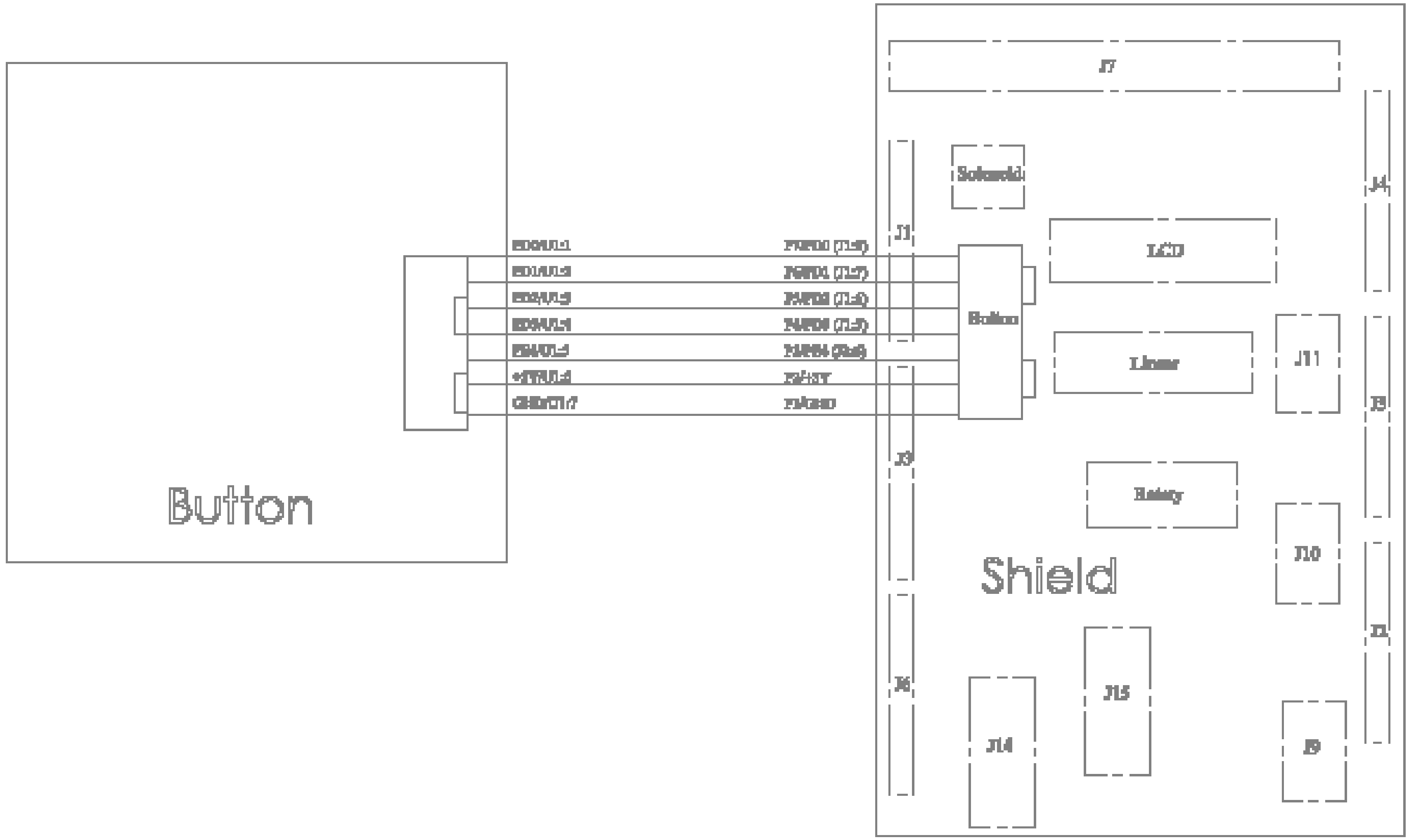
APPENDIX D : WIRING DIAGRAMS

11521301	BUTTON/SHIELD
11521302	SOLENOID/SHIELD
11521303	LINEAR/SHIELD
11521304	ROTARY/SHIELD



SolidWorks Student Edition.
For Academic Use Only.

Concordia			TOLERANCES X ± .1, .X ± .05, .XX ± .005, .XXX ± .002, ANGLE ± .5°	TITLE LCD/Shield
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER B.TETREAULT	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED		A SIZE — DATE 3/20/2015 — USED ON MECH 490
	DRAFTER B.TETREAULT		SCALE 1.6:1	SHEET 1 / 1 — DWG NO. 11521300 — REV NO. 02



Concordia University

DESIGNER B.TETREAULT

DRAFTER B.TETREAULT

ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED

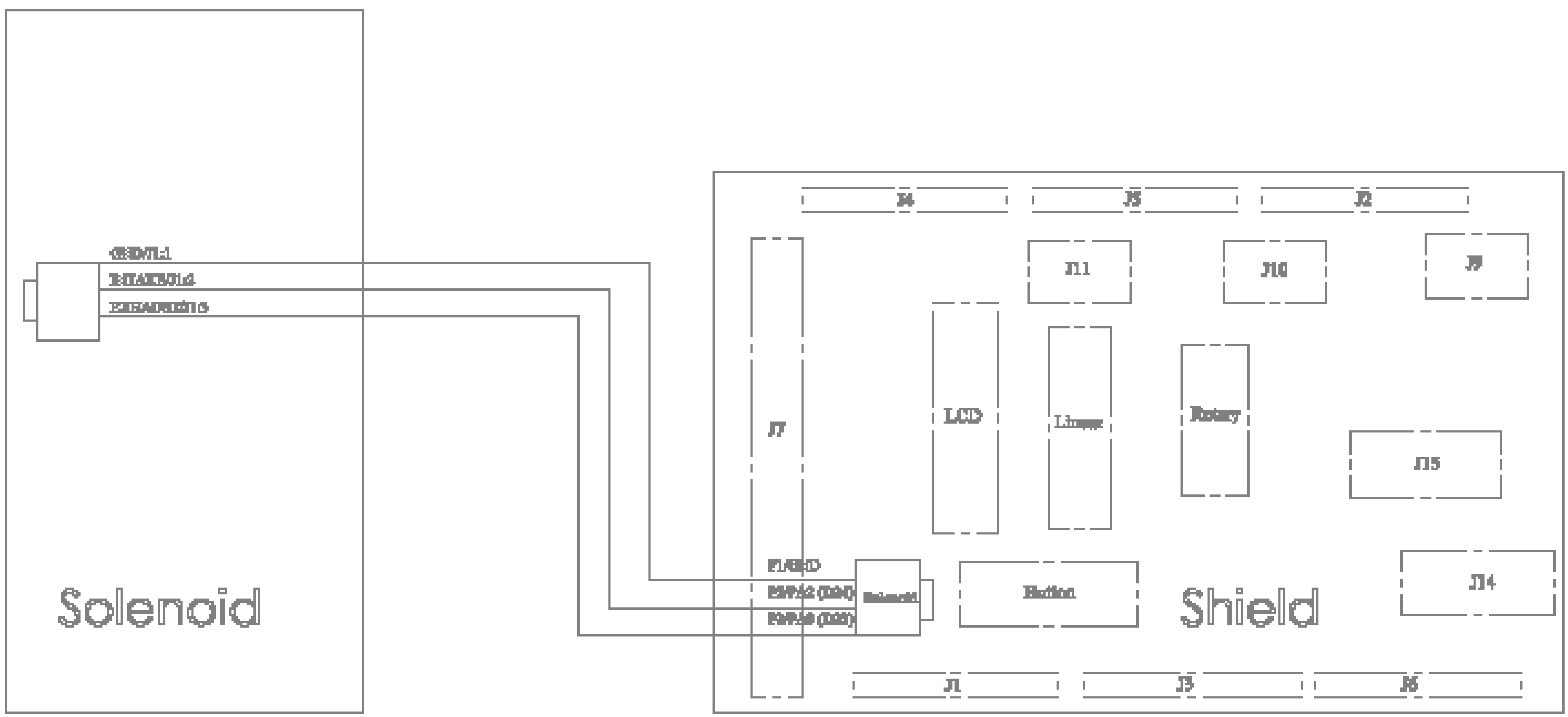
SURFACE ROUGHNESS

TOLERANCES

TITLE Button/Shield

A SIZE | DATE 3/20/2015 | USED ON MECH 490

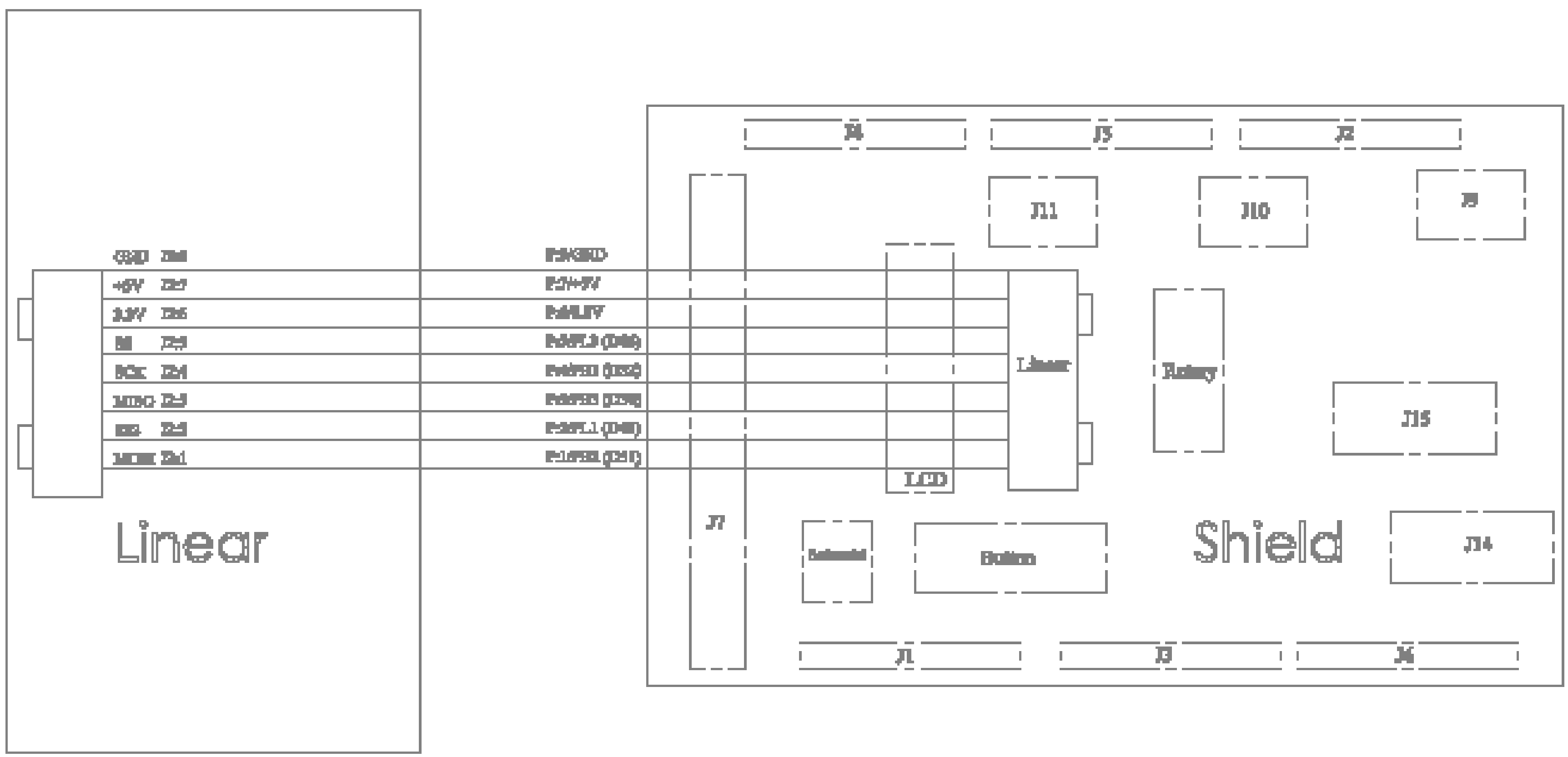
SCALE 1.6:1 | SHEET 1 / 1 | DWG NO. 11521301 | REV NO. 02



Concordia University

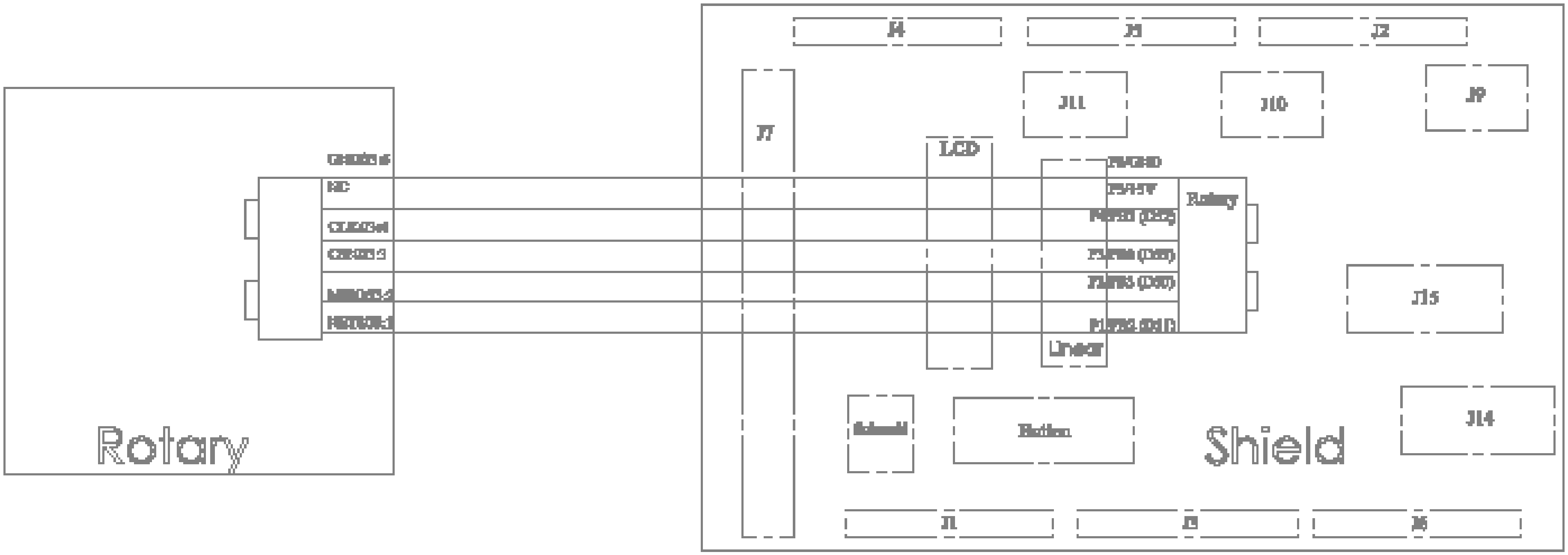
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY

DESIGNER B.TETREAULT	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED	SURFACE ROUGHNESS √ TOLERANCES X ± .1 .XX ± .05 .XXX ± .005 .XXXX ± .000 ANGLE ± .5°	TITLE Solenoid/Shield
DRAFTER B.TETREAULT		SCALE 1.6:1	A SIZE · DATE 3/20/2015 · USED ON MECH 490 · SHEET 1 / 1 · DWG NO. 11521302 · REV NO. 02



SolidWorks Student Edition.
For Academic Use Only.

Concordia University		SURFACE ROUGHNESS; TOLERANCES X ± .1, .X ± .05, .XX ± .005, .XXX ± .000, ANGLE ± .5°	TITLE Linear/Shield	
PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER B.TETREAULT	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED	A SIZE; DATE 3/20/2015	USED ON MECH 490
	DRAFTER B.TETREAULT	SCALE 1.6:1	SHEET 1 / 1; DWG NO. 11521303	REV NO. 02



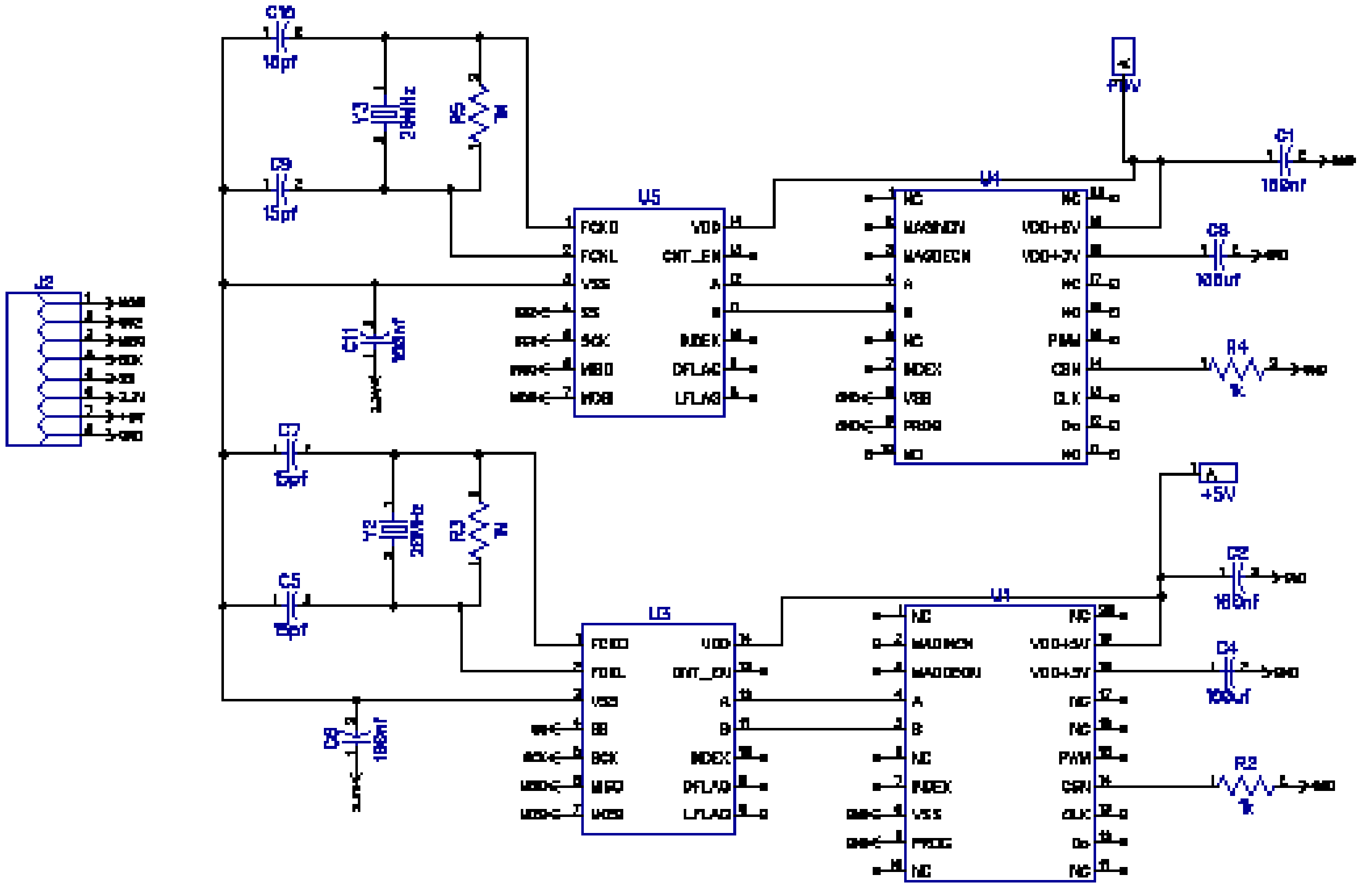
SolidWorks Student Edition.
For Academic Use Only.

PROPRIETARY INFORMATION NOT TO BE RELEASED WITHOUT WRITTEN AUTHORIZATION FROM CONCORDIA UNIVERSITY	DESIGNER B.TETREAULT	ALL DIMENSIONS IN INCHES UNLESS OTHERWISE SPECIFIED	SURFACE ROUGHNESS √6.3 TOLERANCES
	DRAFTER B.TETREAULT		SCALE 1.6:1

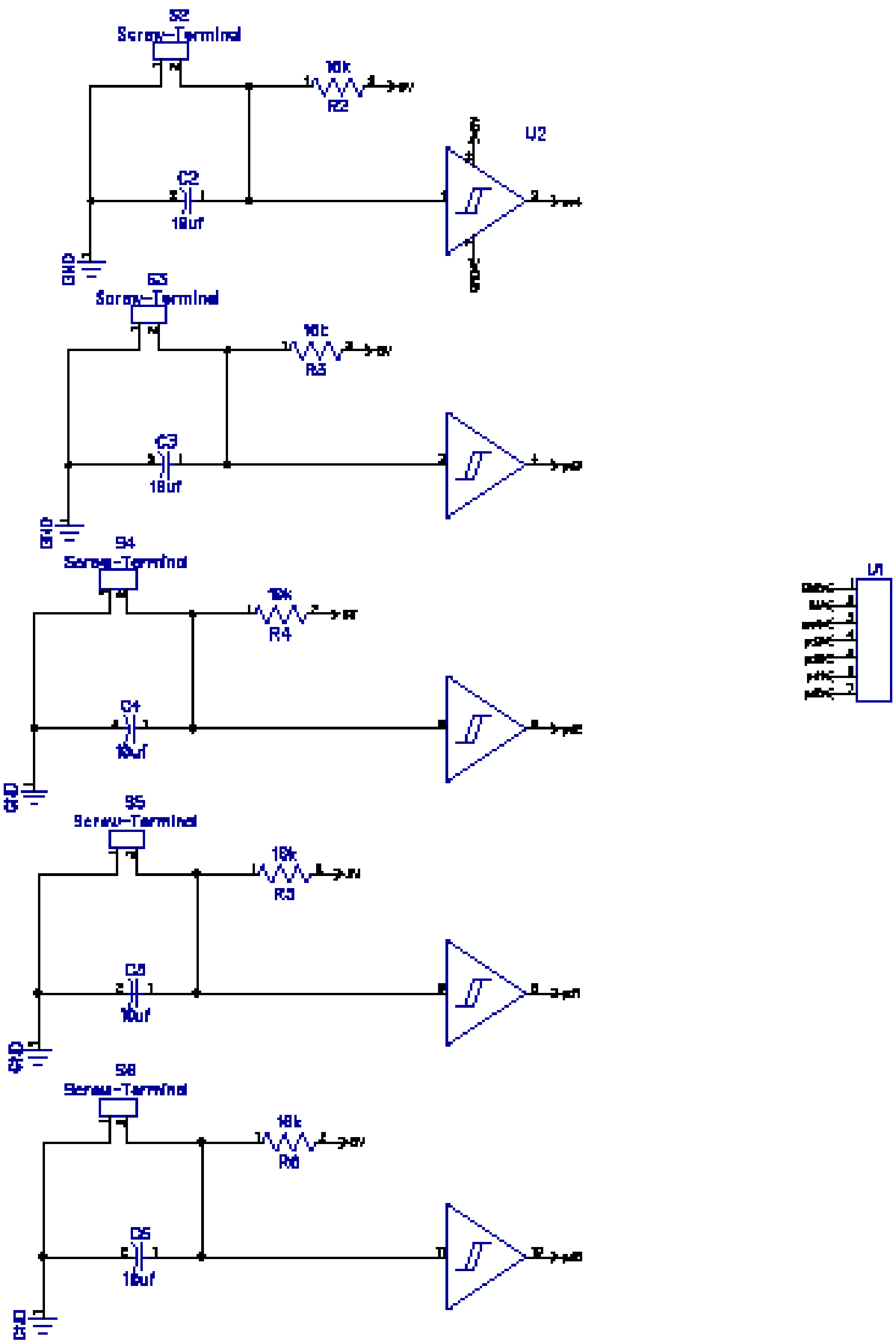
TITLE Rotary/Shield			
A SIZE	DATE 3/20/2015	USED ON MECH 490	
SHEET 1 / 1	DWG NO.11521304	REV NO. 02	

APPENDIX E : ELECTRICAL SCHEMATICS

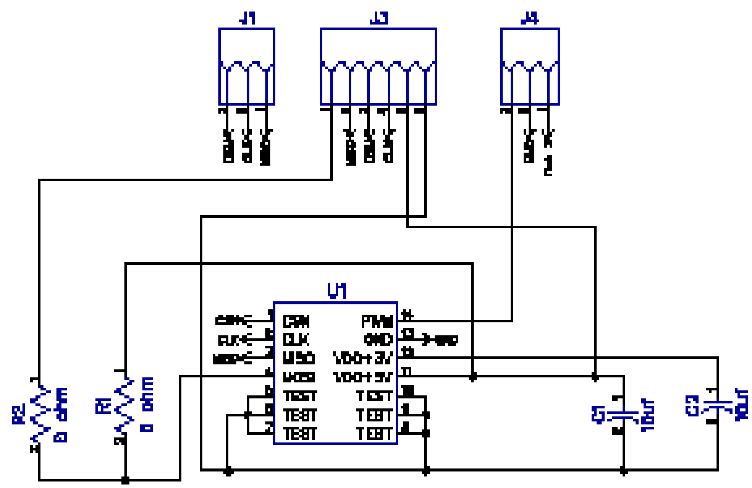
11521400 SHEILD BOARD
11521401 BUTTON BOARD
11521402 DRIVER
11521403 LINEAR SENSOR
11521404 ROTARY SENSOR



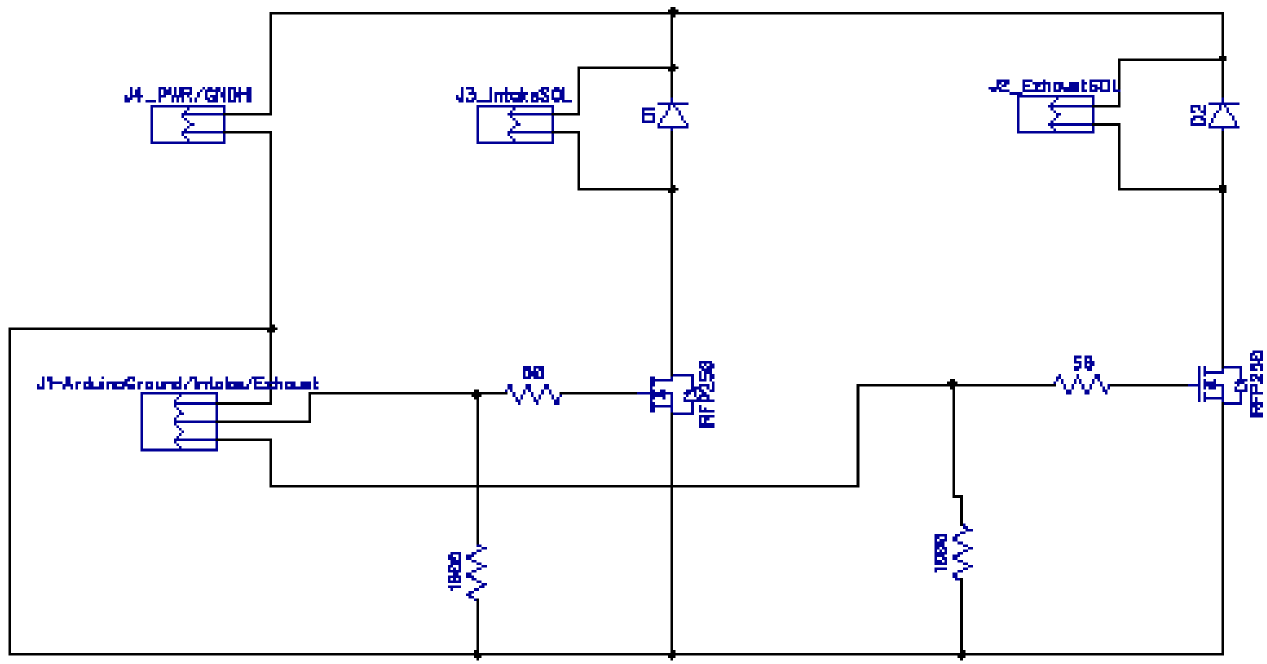
Title: LINEAR SENSOR		
Size t1	Number 11521403	Rev 00
Date:11/30/2014	Drawn by	
Filename: LINEAR-SENSOR	B.ROBINSON	



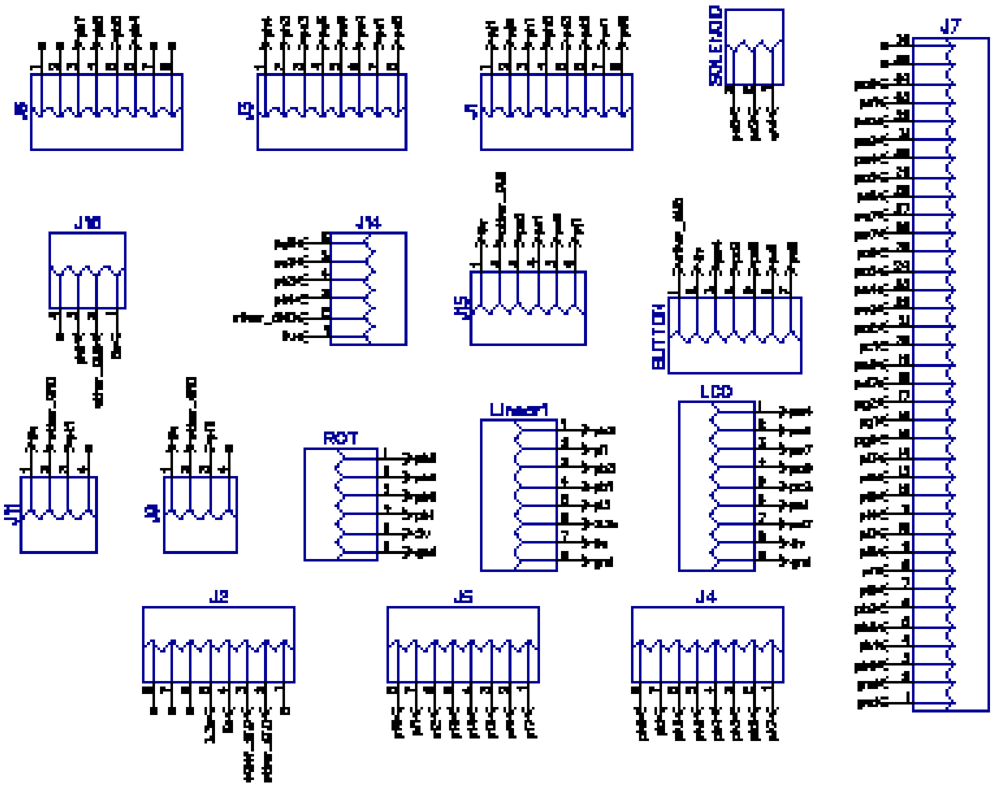
Title: PUSH BUTTON BOARD		
Size	Number	Rev
A1	11521401	00
Date: 11/30/2014	Drawn by	
Filename: PUSH_BUTTON	B.ROBINSON	



Title: ROTARY SENSOR		
Size 2:1	Number 11521404	Rev B0
Date:11/30/2014	Drawn by	
Filename: ROTARY-SENSOR	B.ROBINSON	



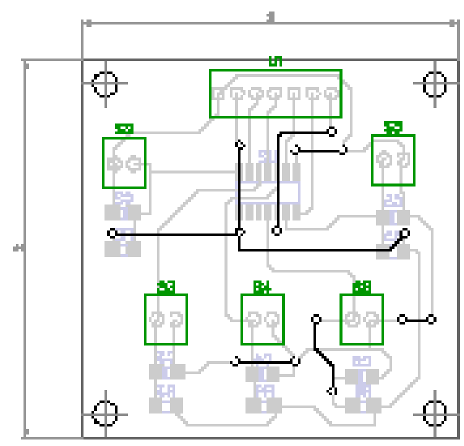
Title: SOLENOID DRIVER SCHEMATIC		
Size	Number	Rev
1:1	11521402	3
Date: 11/16/2014	Drawn by	
Filename SolenoidCircuit	J.BRACKEN	



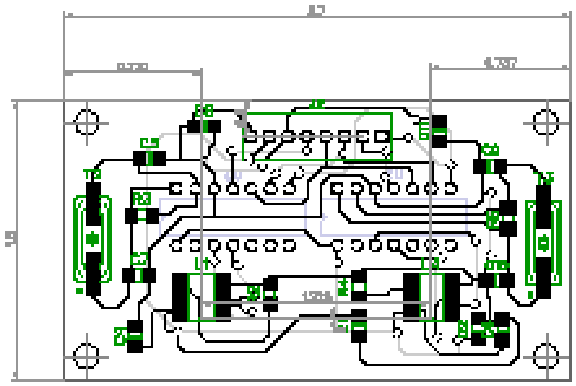
ARDUINO SHIELD		
Size 3:2	Number 1152140B	Rev 00
Date: 11/30/2014	Drawn by B.ROBINSON	
Filename: SHIELD		

APPENDIX F : PRINTED CIRCUIT BOARDS

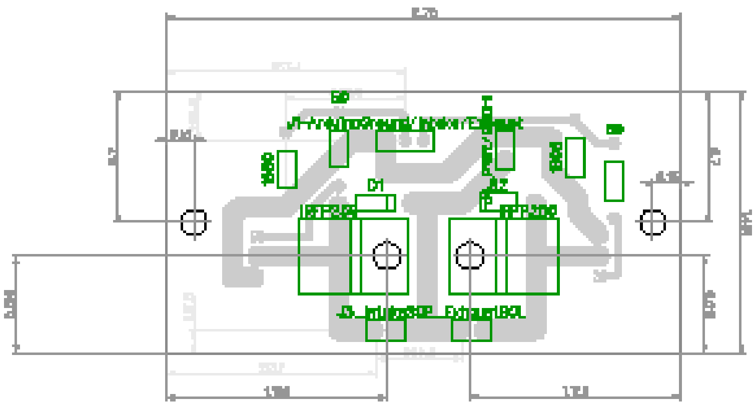
11521500 SHIELD
11521501 BUTTON BOARD
11521502 SOLENOID DRIVER
11521503 LINEAR SENSOR
11521504 ROTARY SENSOR



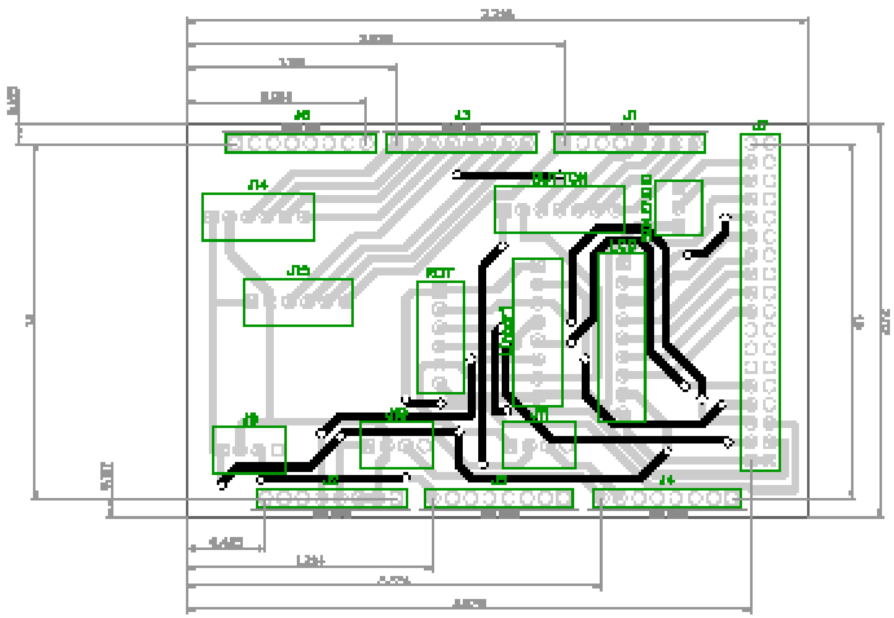
Title: BUTTON BOARD		
Size 3:1	Number 11521501	Rev 00
Date: 11/30/2014	Drawn by	
Filename: BUTTON-BOARD-1	B.ROBINSON	



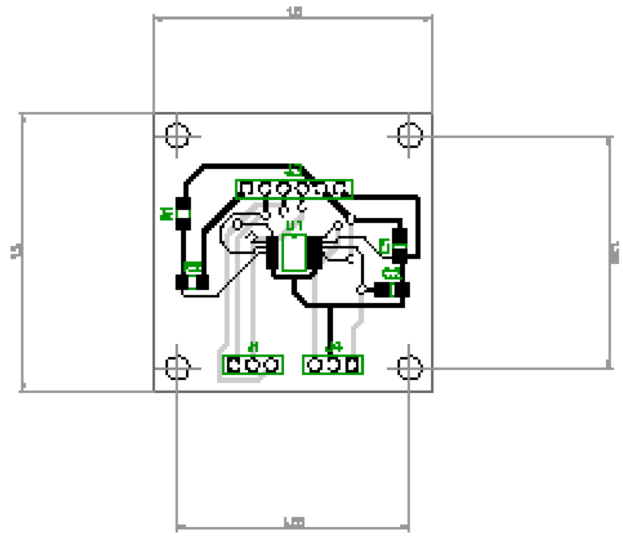
Title:LINEAR SENSOR		
Size 3:1	Number 11521503	Rev 00
Date: 11/30/2014	Drawn by	
Filename: LINEAR-SENSOR	B.ROBINSON	



Title: SOLENOID DRIVER PCB		
Size 1:1	Number 11521582	Rev 9
Date 11/28/2014	Drawn by	
Filename SolenoidPCB7	J.BRACKEN	



Title: ARDUINO SHIELD		
Size 2:1	Number 11521500	Rev 00
Date: 11/30/2014	Drawn by	
Filename: SHIELD	B.ROBINSON	



Title: ROTARY SENSOR		
Size 3:1	Number 11521504	Rev 00
Date: 11/30/2014	Drawn by	
Filename: Rotary_Sensor.dfp	B.Robinson	

APPENDIX G : PRELIMINARY TESTING PROCEDURES

G.1 MEASURE CAM PROFILE [COMPLETE]

G.1.1 Required Equipment

- Camshaft
- Lathe
- Dial Indicator

G.1.2 Safety

- Wear safety glasses in EDML
- Supervised by lab technician Mike Rembacz

G.1.3 Testing Procedure

1. Place camshaft in lathe using two centers to not scratch camshaft.
2. Set up dial indicator, and zero it on the low side of the cam.
3. Rotate camshaft, and using the cam gear as an angle indicator, check the dial indicator at each cam gear tooth.
4. Repeat for both cams.
5. Input data into excel.
6. Check cam offset by comparing cam peaks.

G.1.4 Results

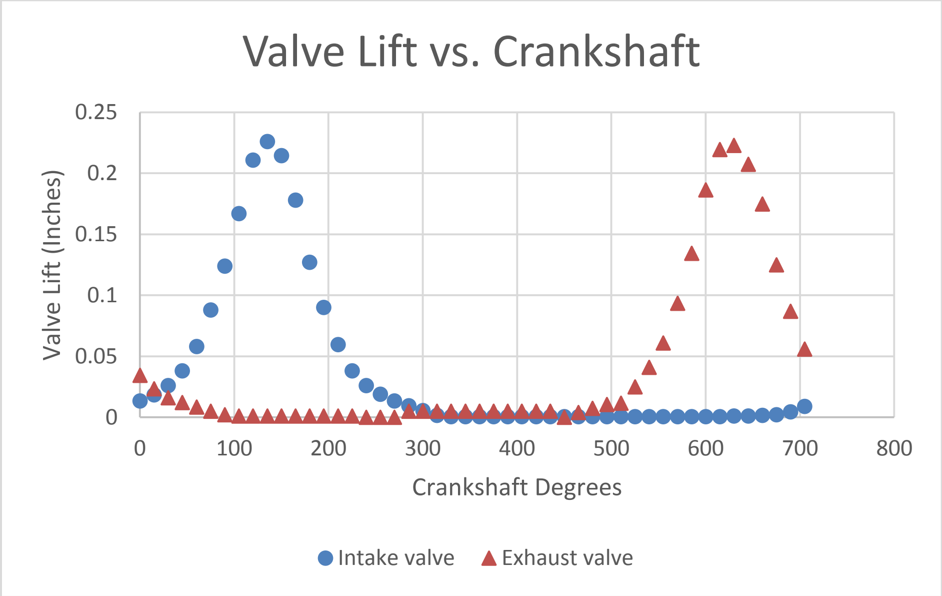


Figure 20: Measured camshaft profile

These results demonstrate an overlap of approximately 90 degrees on the crankshaft and both valve are open for 315 crankshaft degrees per cycle. The maximum valve lift is 0.223 inches. This is the data that will need to be respected during the system validation, and must be replicated as close as possible.

G.2 TEST MICRCOCONTROLER LOOP DELAY [COMPLETE]

G.1.5 Required Equipment

- Arduino Mega 2560
- Oscilloscope
- Frequency generator

G.1.6 Safety

- Safety glasses
- Supervised by lab technician Gilles Huard

G.1.7 Testing Procedure

Check time at beginning of loop using millis() and compare with the value of millis() at second loop iteration.

This needs to be checked continuously throughout the project to determine the delay associated with new code added.

G.1.8 Results

Current loop iteration: ~1 ms

G.3 TEST ANGLE POSTION SENSOR [COMPLETE]

G.1.9 Required Equipment

- Arduino Mega 2560
- Power supply
- Frequency generator
- 3D printed solenoid fixture for DC motor
- Osciliscope
- Strobelight

G.1.10 Safety

- Wear safety glasses
- Supervised by lab technician Gilles Huard

G.1.11 Testing procedure

Refer to Figure *21 for testing setup*

1. Calibrate DC motor with strobe light for 0-6V and determine RPM per volt
2. Assemble circuit for rotary angle sensor
3. Upload software and power Arduino microcontroller
4. Place DC motor and fixture over rotary sensor
5. Repeat for a measured distance and ensure software calibration is correct

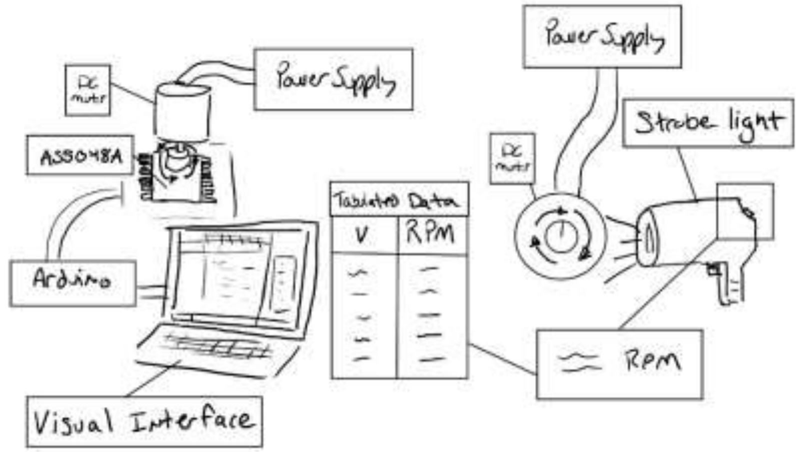


Figure 21: Angular positon sensor and DC motor calibration test setup

G.1.12 Results

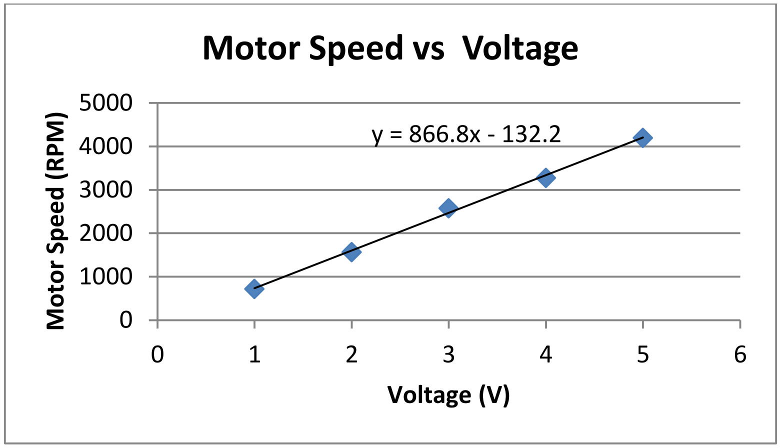


Figure 22: DC motor calibration curve

Software calibration works correctly; motor positions varies from 0-16384 (14 bit resolution)

G.4 TEST SOLENOIDS AND DRIVER [COMPLETE]

G.1.13 Required Equipment

- Power supply
- Frequency generator
- Driver circuit
- Solenoids
- Wires
- 3D printed solenoid fixture
- Ruler
- Spring

G.1.14 Safety

- Wear safety glasses

- Supervised by lab technician Gilles Huard

G.1.15 Testing Procedure

1. Mount the solenoid and the spring into the 3D printed fixture.
2. Connect the solenoid to the power supply.
3. Connect the solenoid to the driver circuit.
4. Connect the driver circuit to the frequency generator.Turn on power supply to 5 V and set the frequency to 1Hz.
5. Observe the distance travelled.
6. Repeat step 6 using increments of 1Hz until 20Hz is achieved.
7. Observe solenoid temperature throughout test.
8. Repeat test using DC motor as input to microcontroller, and microcontroller sending the signal to the solenoid to open and close. Use test setup of Figure *23* as reference.

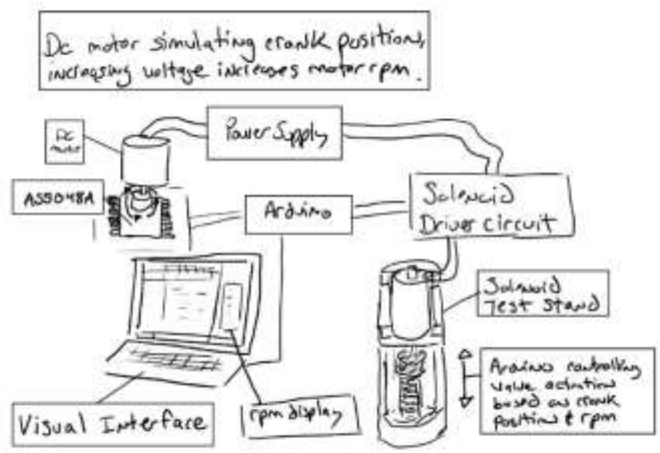


Figure 23: DC motor and solenoid test setup

G.1.16 Results

Solenoid operated visually up past 20 Hz. Solenoid displacement must be verified in the Test and Tune Actuators procedure.

The solenoid and microcontroller were able to communicate together and vary speed according to the DC motor. Validation of lift and phase will be completed in the Test and Tune Actuators procedure.

G.5 TEST LINEAR POSTION SENSOR [COMPLETE]

G.1.17 Required Equipment

- Arduino Mega 2560
- Oscilloscope
- Linear sensor and counter circuit

G.1.18 Safety

- Wear safety glasses
- Supervised by lab technician Gilles Huard

G.1.19 Testing procedure

1. Assemble circuit for linear sensor
2. Upload software and power Arduino microcontroller
3. Pass multi-pole magnet over sensor and observe if sensor and counter are acting correctly
4. Repeat for a measured distance and ensure software calibration is correct

G.1.20 Results

Sensor is working correctly, values are adding and subtracting in the correct fashion to the microcontroller. The sensor simply needs to be calibrated for counts/mm.

G.6 TEST AND TUNE ACTUATORS [COMPLETE]

G.1.21 Required Equipment

- Arduino Mega 2560
- Oscilloscope
- Frequency generator
- Solenoids
- Driver
- Test stand
- Spring
- Power supply
- Linear position sensor
- 3D printed fixture

G.1.22 Safety

- Safety Glasses
- Monitored by lab technician Gilles Huard

G.1.23 Testing Procedure

1. Mount the solenoid and the spring into the 3D printed fixture.
2. Connect the solenoid tip to the linear position sensor.
3. Connect the linear position sensor to the oscilloscope.
4. Connect the power supply to the solenoid.
5. Connect the solenoid to the driver circuit.
6. Connect the Arduino mega to the driver circuit.
7. Observe the motion of the solenoid at different frequencies and duty cycles.

G.7 PRIMARY VALIDTION [COMPLETE]

G.1.24 Required Equipment

- Arduino Mega 2560
- Oscilloscope
- Validation test setup
- Power supply

G.1.25 Safety

- Safety Glasses
- Monitored by lab technician Gilles Huard

G.1.26 Testing Procedure

See Figure 24 for validation test setup

1. Mount solenoid in engine head assembly attached to test stand
2. Connect the solenoid tip to the linear position sensor.
3. Connect the power supply to the solenoid.
4. Connect the solenoid to the driver circuit.
5. Connect the Arduino mega to the driver circuit.
6. Observe the motion of the solenoid at different RPM’s determined by a DC motor

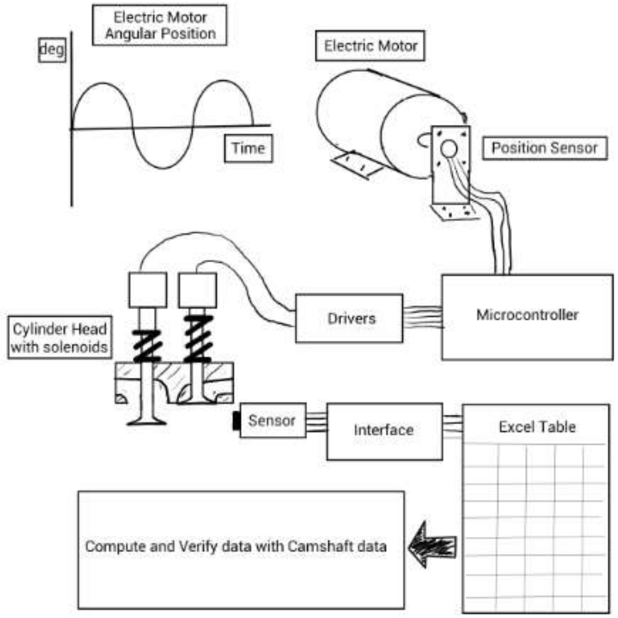


Figure 24: Primary Validation test setup

G.8 TEST CARBURATED ENGINE [COMPLETE]

G.1.27 Required Equipment

- Arduino Mega 2560
- Oscilloscope
- Test stand
- Modified engine head with solenoids and drivers
- Linear sensor
- Rotational sensor
- Power supply

G.1.28 Safety

- Safety Glasses
- Supervised by lab technician Gilles Huard
- Fire extinguisher

G.1.29 Testing procedure

1. Mount the spring and solenoid to the modified engine.
2. Connect linear sensor to the solenoid and oscilloscope.
3. Connect solenoids to driver.
4. Connect Arduino mega to driver.
5. Connect rotational sensor to the oscilloscope.
6. Start engine and Arduino program.
7. Observe rotational idle speed using the oscilloscope.
8. Modify duty cycle and record engine performance.
9. Modify solenoid frequency and record engine performance.

APPENDIX H : DECISION MATRICES

Actuator Decision matrix

Criteria ->	General				MANUFACTURABILITY		COST RISK	Engineering	SCORE
Sub Criteria	Frequency	Size	Stroke length	Force	Material	Lead time	hardware	Quality of Analysis	
ıment for sub criteria	16 Hertz	2 in diameter	0.25 in	0.5-1 lbf	steel	Less than 1 month	Less than 100$		
Coarse Weight	100				40		100	25	
Fine Weight	20	5	35	40	20	20	100	25	
percentage	0.07547	0.01887	0.13208	0.15094	0.07547	0.07547	0.37736	0.09434	
Option description									
Voice coil	9	5	8	10	7	5	1	6	5.18868
Low profile solenoid	9	7	8	8	7	7	8	8	7.90566
Tubular solenoids	**8**	**7**	**8**	**8**	**7**	**7**	**9**	**7**	**8.11321**
Soft shift solenoid	9	7	8	7	7	7	4	8	6.24528

Figure 25- Decision Matrix for Actuator

Microcontroller Desiscion Matrix

Criteria ->	General				COST RISK	SCORE
Sub Criteria	EEPROM	Flash	Clock Spe	Number o	Price	
comment for sub criteria						
Coarse Weight	100				100	
Fine Weight	40	5	40	15	100	
percentage	0.2	0.025	0.2	0.075	0.5	
Option description						
Arduino UNO	4	5	6	9	8	6.8
Raspberry Pi	1	1	9	6	6	5.475
BeagleBone	1	8	9	6	4	4.65
Arduino Mega	**8**	**7**	**8**	**9**	**8**	**8.05**

Figure 26: Decision Matrix for Microcontroller

Decision matrix (Actuator type)										
	Criteria ->	General				MANUFACTURABILITY		COST RISK	Engineering	SCORE
	Sub Criteria	Frequency	Size	Simplicity	Force	Extra components	Lead time	hardware	Quality of Analysis	
CHECK SUM	comment for sub criteria	16 Hertz	1.5 in diameter	setup	0.5-1 lbf	Less than 2	Less than 1 month	Less than 100$		
265	Coarse Weight	100				40		100	25	
265	Fine Weight	30	25	15	30	20	20	100	25	
	percentage	0.113207547	0.094339623	0.05660377	0.113207547	0.075471698	0.075471698	0.377358491	0.094339623	
Option ->	Option description									
A	Pneumatic actuator	9	7	5	8	6	8	4	7	6.09433962
B	Electric actuator	7	8	9	7	8	8	8	8	**7.83018868**
C	Hydraulic actuator	9	7	5	8	6	8	4	7	6.09433962
D	Piezoelectric actuator	6	8	9	7	8	8	5	8	6.58490566
Option ->	WEIGHTED SCORES									
A	Pneumatic actuator	1.018867925	0.660377358	0.28301887	0.905660377	0.452830189	0.603773585	1.509433962	0.660377358	
B	Electric actuator	0.79245283	0.754716981	0.50943396	0.79245283	0.603773585	0.603773585	3.018867925	0.754716981	
C	Hydraulic actuator	1.018867925	0.660377358	0.28301887	0.905660377	0.452830189	0.603773585	1.509433962	0.660377358	
D	Piezoelectric actuator	0.679245283	0.754716981	0.50943396	0.79245283	0.603773585	0.603773585	1.886792453	0.754716981	
								Rating system		
NOTES:	User fills in only areas highlighted colors below							Not applicable	0	
		-> text						Very bad	1--2	
		-> entries in this row can add up to any number, weights are relative to each other						Bad	3--4	
		-> each column in this color should add up to the same number						Average	5--6	
NOTES 2: Be careful when inserting and copying rows as somem formulas use 'fixed' cell references								Good	7--8	
In the figure below, J8 should have been K8								Very good	9--10	

APPENDIX I : MINI- PDM

Concordia University
ENCS
Engineering & Computer Science

Year	2014-2015
Team	21
Device	Camless Engine

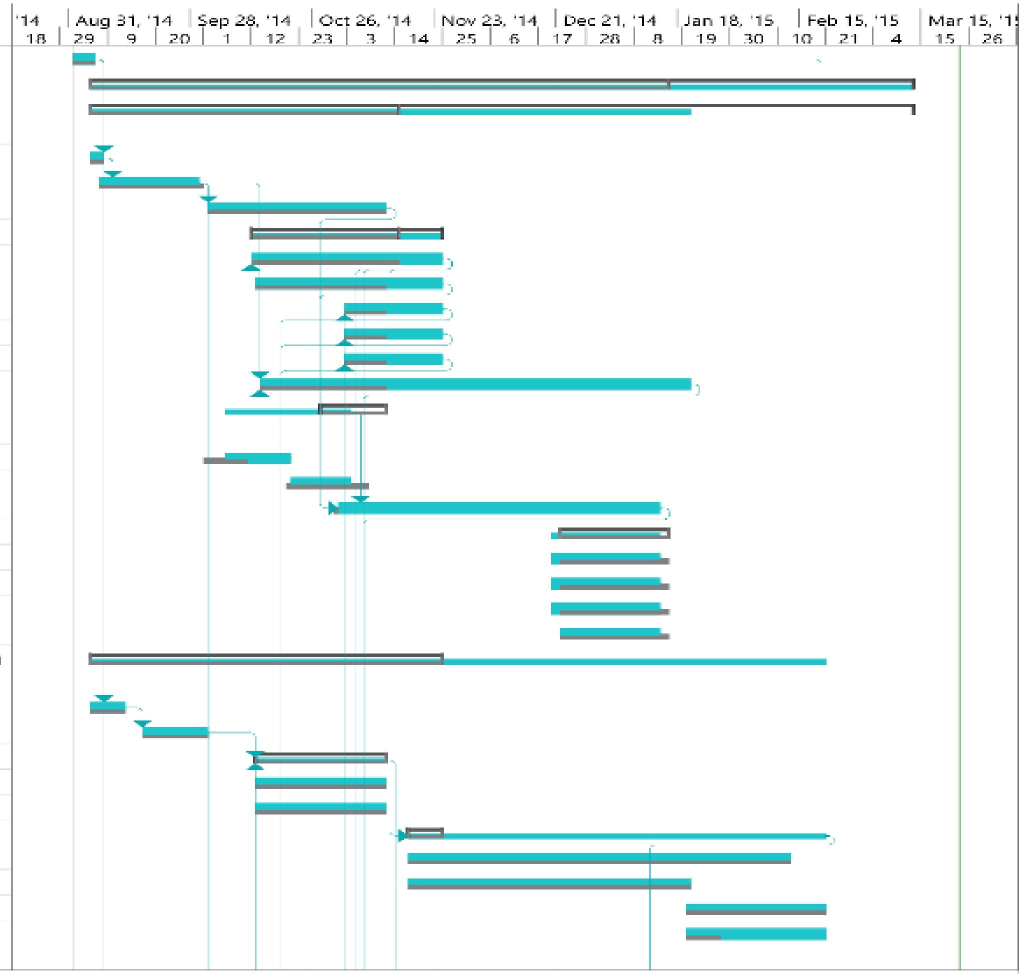
Item	Description	Drawing#/Part#	MNF/OEM/SPL	QTY	Value per Unit [$]	Cost [$]	Price Paid [$]	Shipping/Taxes[$]
	Primary Objective							
1	OHV Engine	Honda GX200	Student	1	190.00	190.00	150.00	
2	Solenoids	195207-224	Ledex Solenoids	2	27.20	54.40	54.40	
3	Arduino	Mega 2560		1	15.65	15.65	15.65	3.11
4	Linear Sensor	AS5311	Austriamicrosystems	2	5.70	11.40	0.00	
5	Linear Magnet	AS5000-MS10-300	Austriamicrosystems	1	7.50	7.50	0.00	
6	Linear Driver	LS7366		1	49.32	49.32	49.32	23.81
7	Rotary Sensor	AS5048A & AS5048B	Austriamicrosystems	6	18.60	111.60	0.00	18.39
8	Rotary Magnet	AS5000-MD6H	Austriamicrosystems	3	3.96	11.88	0.00	
9	Mosfet	CSD18504KCS	Texas Instruments	3	1.96	5.88	0.00	
10	Toolcart Bench		EDML C	1				
11	SMT Breakout PCB	1207-ADA	Abra electronics Corp	1	4.54	4.54	4.54	
12	5V POS VOLT REG	7805T	Abra electronics Corp	2	0.46	0.92	0.92	
13	Resistor	R1/2-1K	Abra electronics Corp	10	0.06	0.60	0.60	
14	Ceramic DISC Capacitor	CD333	Abra electronics Corp	10	0.09	0.90	0.90	
15	Ceramic DISC Capacitor	CD104	Abra electronics Corp	10	0.09	0.90	0.90	
16	Capacitor	100R6.3	Abra electronics Corp	3	0.15	0.45	0.45	
17	Capactor	10R16	Abra electronics Corp	3	0.13	0.39	0.39	
18	Snappable Header	SH-2	Abra electronics Corp	1	0.79	0.79	0.79	
19	12V, 8 amp Power Supply	RS-100-12	EDML C	1	30.45	30.45	0.00	
	Circuit Board Linear Sensor							
20	20 MHz Crystal	XC1254CT-ND	Digi-Key	2	0.68	1.36	1.36	23.27
21	15 pF Capacitor	399-9314-1-ND	Digi-Key	4	0.17	0.68	0.68	
22	1M OHM Resistor	P1.0MECT-ND	Digi-Key	2	0.13	0.26	0.26	
23	100 nF Capacitor	399-1249-2-ND	Digi-Key	4	0.12	0.48	0.48	
24	100 uF Capacitor	445-6007-1-ND	Digi-Key	2	2.03	4.06	4.06	
25	1K Resistor	P1.0KECT-ND	Digi-Key	2	0.13	0.26	0.26	
26	8 POS Molex Connector	WM4206-ND	Digi-Key	1	0.84	0.84	0.84	
27	8 POS Molex Receptacle	WM2006-ND	Digi-Key	1	0.41	0.41	0.41	
	Circuit Board Rotary Sensor							
28	6 POS Molex Connector	WM4204-ND	Digi-Key	1	0.53	0.53	0.53	
29	6 POS Molex Receptacle	WM2004-ND	Digi-Key	1	0.31	0.31	0.31	
30	10 uF Capacitor	587-1353-1-ND	Digi-Key	2	0.25	0.50	0.50	
31	0.0 OHM Resistor	P0.0ECT-ND	Digi-Key	2	0.13	0.26	0.26	
	Circuit Board Solenoid Driver							
32	2 POS Molex Connector	WM4200-ND	Digi-Key	3	0.24	0.72	0.72	
33	3 POS Molex Connector	WM4201-ND	Digi-Key	1	0.36	0.36	0.36	
34	2 POS Molex Recepticle	WM2000-ND	Digi-Key	3	0.13	0.39	0.39	
35	3 POS Molex Recepticle	WM2001-ND	Digi-Key	1	0.19	0.19	0.19	
36	50 OHM Resistor	CF18JT51R0CT-ND	Digi-Key	2	0.13	0.26	0.26	
	Shield							
37	6 POS Molex Connector	WM4204-ND	Digi-Key	3	0.53	1.59	1.59	
38	8 POS Molex Connector	WM4206-ND	Digi-Key	2	0.84	1.68	1.68	
39	4 POS Molex Connector	WM4202-ND	Digi-Key	3	0.45	1.35	1.35	
40	7 POS Molex Connector	WM2705-ND	Digi-Key	1	1.04	1.04	1.04	
41	6 POS Molex Receptacle	WM2004-ND	Digi-Key	3	0.31	0.93	0.93	
42	8 POS Molex Receptacle	WM2006-ND	Digi-Key	2	0.41	0.82	0.82	
43	4 POS Molex Receptacle	WM2002-ND	Digi-Key	3	0.20	0.60	0.60	
44	7 POS Molex Receptacle	WM2005-ND	Digi-Key	1	0.36	0.36	0.36	
45	Through Hole SQ Post Socket	SSQ-1-08-03-T-S	Samtec	5				
46	Through Hole SQ Post Socket	SSQ-118-03-T-D	Samtec	1				
47	Through Hole SQ Post Socket	SSQ-110-03-T-S	Samtec	1				

	Hardware							
48	M3 x0.5mm x 4mm	90116A147	McMaster-Carr	8	0.06	0.49	0.49	
49	M8 x 1.25mm x 65mm	93635A383	McMaster-Carr	4	0.09	0.38	0.38	
50	M8x1.25 Steel Hex Nut	90592A022	McMaster-Carr	4	0.05	0.18	0.18	
51	8-32 3/16"	91772A189	McMaster-Carr	2	0.09	0.18	0.18	
52	Spacer	952-2190-ND	Digi-Key	6	0.42	2.52	2.52	
53	Flanged Head Cap Screw 25mm	98093A445	McMaster-Carr	4	0.11	0.44	0.44	
54	M4 x 0.7mm x 16mm	91280A136	McMaster-Carr	8	0.07	0.53	0.53	
	Button Board							
55	10k Resistor	P10KECT-ND		5	0.10	0.50	0.50	
56	10 uF Capacitor	587-1353-1-ND	Digi-Key	5	0.25	1.25	1.25	
57	Schmitt	74LVX14MXCT-ND	Digi-Key	1	0.57	0.57	0.57	
58	7 POS Molex Connector	WM2705-ND	Digi-Key	1	1.04	1.04	1.04	
59	7 POS Molex Receptacle	WM2005-ND	Digi-Key	1	0.36	0.36	0.36	
60	Button	501PB-ND	Digi-Key	4	1.21	4.84	4.84	
61	20" x 4" LCD Screen	NHD-0420DZ-FL-GBW-ND	Digi-Key	1	17.54	17.54	17.54	
	Miscellaneous							
62	Interface Bottom	11421701	10th floor labs	1				
63	Interface Top	11421702	10th floor labs	1				
64	Molex Pin 100 pack	WM2312-ND	Digi-Key	100	0.18	17.76	17.76	
65	Solenoid Driver	DRV103	Texas Instruments	3	5.20	15.60	0.00	
66	Current Controlller	LT3741	Texas Instruments	2	4.98	9.96	0.00	
67	Spring	WF20-30	Misumi	3	2.24	6.72	6.72	
68	Machine Scew M3	335-1150-ND	Digi-key	16	0.48	7.73	7.73	12.39
69	Machine Scew M3	335-1157-ND	Digi-Key	6	0.57	3.42	3.42	
70	Magnet D= 1/4" H=1/16"	469-1005-ND	Digi-Key	16	0.27	4.29	4.29	
71	Bumper Square	SJ5508-0-ND	Digi-Key	64	0.08	5.09	5.09	
72	Con Header 6POS	WM4304-ND	Digi-Key	2	0.82	1.64	1.64	
73	Con Terminal Female 22-30AWG	WM1114-ND	Digi-Key	50	0.14	7.15	7.15	
	Secondary Objective Carburated							
74	Switch		EDML 8	3				
75	Throttle Potentiometer		Student	1	2.00	2.00	0.00	
76	M8 x 1.25mm x 35mm Brass Bolt	93270A545	McMaster-Carr	1	4.00	4.00	4.00	
77	Alternator		Student	1	80.00	80.00	0.00	
78	Servo Motor		Student/EDML C	1	15.00	15.00	0.00	
79	Pressure Sensor	PCB 112A22	Hall 10th Floor	1				
80	Lovejoy L-090 .750 , 3/16 keyway	68514476291	General Bearing Service	1	13.46	13.46	13.46	
81	Lovejoy L-090 .500	68514410769	EDML C	1	12.00	12.00	0.00	
82	Lovejoy L-090 rubber coupler		EDML C	1	6.00	6.00	0.00	
83	Temperature Sensor	P1K0.161.6W.B.010	Newark Element14	1	19.68	19.68	19.68	
84	Spacer/Standoff Hex 6mmx6mm	M1303-3005-N	Newark Element14	8	0.31	2.51	2.51	
85	Spacer M3 Screw Mount Lock-In	06.5.086	Newark Element14	8	0.70	5.62	5.62	
86	Wire Board Connector Header 2POS	22-23-2021	Newark Element14	5	0.23	1.16	1.16	
87	Socket Housing Connector 2POS	22-01-3027	Newark Element14	5	0.15	0.76	0.76	
88	Wire Board Connector Header 16POS	22-23-2161	Newark Element14	1	0.94	0.94	0.94	
89	Socket Housing Connector 16POS	22-01-2161	Newark Element14	1	1.36	1.36	1.36	
90	1/4 x 1/8 Union M	123-6001-0	Canadian Tire	1	3.99	3.99	3.99	1.80
91	CR7HSA Spark Plugs	018-3670-6	Canadian Tire	1	7.99	7.99	7.99	
92	Bolt 5/16-18 3.5"	2738	Home Depot	1	1.43	1.43	1.43	0.60
93	Nut 5/16-18	2602	Home Depot	1	0.22	0.22	0.22	
94	Lock Washer 5/16"	2633	Home Depot	1	0.16	0.16	0.16	
95	Flat Washer 5/16"	3256	Home Depot	2	0.13	0.26	0.26	
96	Bolt 3/8 -16 3.5"	3042	Home Depot	1	1.49	1.49	1.49	
97	Nut 3/8"-16	2601	Home Depot	1	0.16	0.16	0.16	
98	Lock Washer 3/8"	2632	Home Depot	1	0.13	0.13	0.13	
99	Flat Washer 1/4"	3255	Home Depot	2	0.11	0.22	0.22	
100	Castrol 10W30 Oil 1L	028-9201-4	Canadian Tire	1	4.29	4.29	4.29	0.64
101	Lock Washer 3/8"	2633	Home Depot	20	0.16	3.20	3.20	0.86
102	Flat Washer 5/16"	3256	Home Depot	10	0.13	1.30	1.30	

103	Flat Washer 1/4"	3255	Home Depot	6	0.11	0.66	0.66	
104	Lock Washer 5/16"	2632	Home Depot	2	0.13	0.26	0.26	
105	Lock Washer 1/4"	2631	Home Depot	2	0.11	0.22	0.22	
106	Nut 1/4"-20	2600	Home Depot	2	0.11	0.22	0.22	
107	3D Printer Filament-1.75mm	10553	Monoprice	1	21.42	21.42	21.42	14.25
108	Spring	WP0019960100	MISUMI	2	18.04	36.07	36.07	
	Manufactured Components							
	Mechanical							
109	BOX REAR	11521200	EDML B / EDML C	1				
110	BOX TOP	11521201	EDML B / EDML C	1				
111	BOX WALL	11521202	Student	1				
112	BOX FRONT	11521203	Student	1				
113	SOLENOID CAP	11521204	EDML A	2	17.33	34.66	0.00	
114	5.5mm PRINTED SPACER	11521205	Student	1				
115	SOLENOID STOPPER	11521206	Student	1				
116	PCB HOLDER	11521207	EDML C	1				
117	LEFT SOLENOID TUBE	11521208	EDML A	1	11.62	11.62	0.00	
118	RIGHT SOLENOID TUBE	11521209	EDML A	1	11.62	11.62	0.00	
119	SOLENOID TUBE PLATE	11521210	EDML A	1	46.81	46.81	0.00	
120	THREADED SOLENOID PLATE	11521211	EDML A	1	65.58	65.58	0.00	
121	PRINTED SOLENOID MAGNET CAP	11521212	Student	2				
122	MOUNT PLATE	11521213	EDML B	1	78.00	78.00	0.00	
123	MAP SENSOR PART	11521214	EDML B	1	10.00	10.00	0.00	
124	GUARD	11521215	EDML B	1	4.83	4.83	0.00	
125	ROTARY SENSOR BRACKET BASE	11521216	EDML C	1				
126	ROTARY SENSOR BRACKET TOP	11521217	EDML C	1				
127	ROTARY SENSOR MAGNET HOLDER	11521218	EDML B	1	2.65	2.65	0.00	
128	SERVOMOTOR THROTTLE SHAFT	11521219	Student	1				
129	ROTARY SENSOR PLATE	11521220	EDML B	1	12.48	12.48	0.00	
130	MAP SENSOR PLATE	11521221	EDML B	1	19.28	19.28		
131	ALTERNATOR BRACKET	11521222	EDML B	1	15.54	15.54	0.00	
	PCB's							
132	SHEILD BOARD	11521500	Elec. CNC	1				
133	BUTTON BOARD	11521501	Elec. CNC	1				
134	DRIVER	11521502	Elec. CNC	1				
135	LINEAR SENSOR	11521503	Elec. CNC	1				
136	ROTARY SENSOR	11521504	Elec. CNC	1				
	Consulting	Gilles Huard						
	Consulting	Rudi Krueger						
	Consulting	Robert Oliver						
	Consulting	Peter Sakaris						
	Consulting	Dr. Henry Hong						
	Consulting	Dr. Hoi Dick Ng						
	Consulting	Dominic Ng						
	Consulting	Michael Rembacz						
	Consulting	Dmitry Rozhdestvenskiy						
	TOTALS					1189.25	516.91	99.12
	TOTAL ESTIMATED COST	1189.25						
	TOTAL ACTUAL COST	616.03						

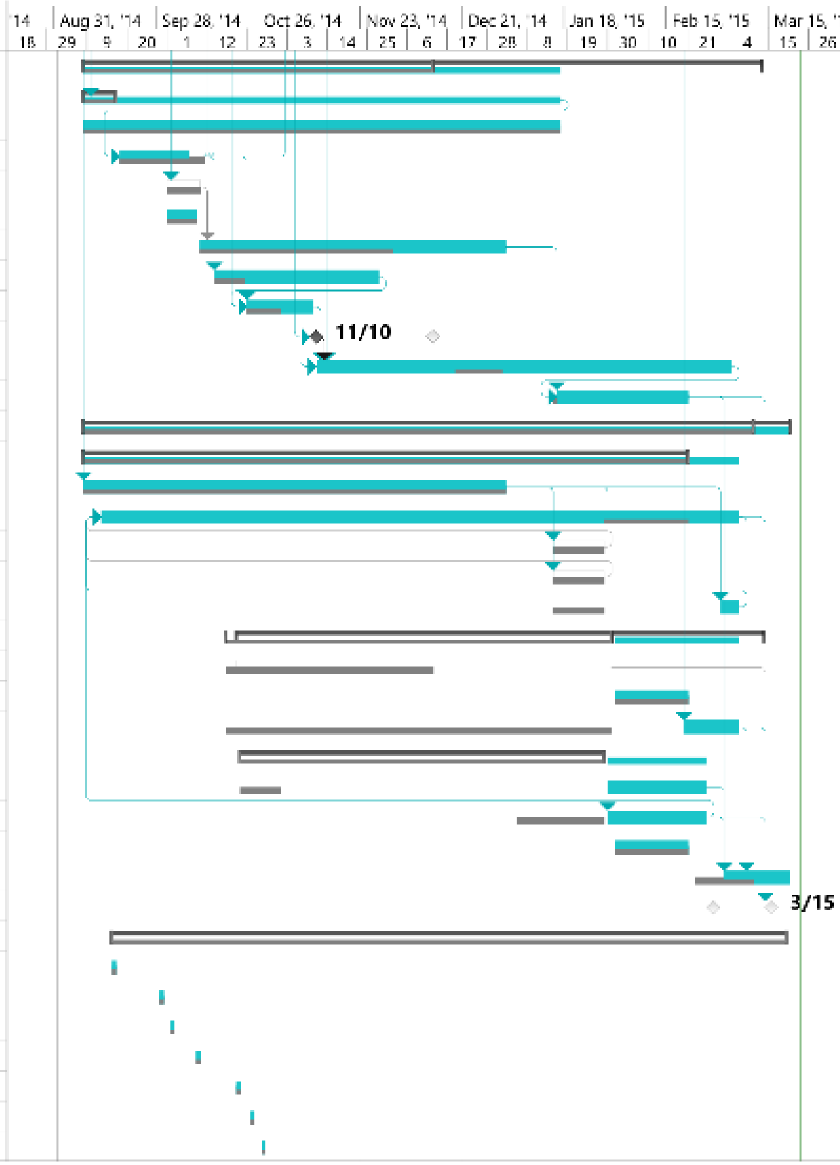
APPENDIX J : PROJECT SCHEDULE GANNT CHART

ID	Task Name
1	Submit Proposal
2	**Primary objective**
3	**Moderling And Fabrication of test stands and engine head modifactions**
14	Disassemble Engine
15	Model engine Parts
16	Design Test stand for primary validation
4	**Design and Approval of PCB's**
7	Linear Sensor (Validation hardware)
5	Rotary Sensor
6	Solenoid Driver
8	Shield Board
9	Button Board
21	Engine head solenoid mod
17	**Design DC motor setup for primary validation with crankshaft sensor**
19	First Protype
18	Final
22	Fabricate Validation Test Stand
10	**Manufacture Primary Wiring Harnesses**
11	Linear Sensor
12	Driver
13	Rotary Sensor
20	**Manufacturing of PCB's and Assembly**
23	**Software for electronic valve actuation and primary objective validation**
24	Choose Microcontroller
25	Preliminary software design
32	**Design Software for specific hardware**
33	Rotary Sensor
34	Linear Sensor
27	**Design Validation Software**
28	Data export from microcontroller
31	VBA data input and plot generator
29	Fix Linear sensor drift
30	Account for system delays
26	~~Protype Software~~

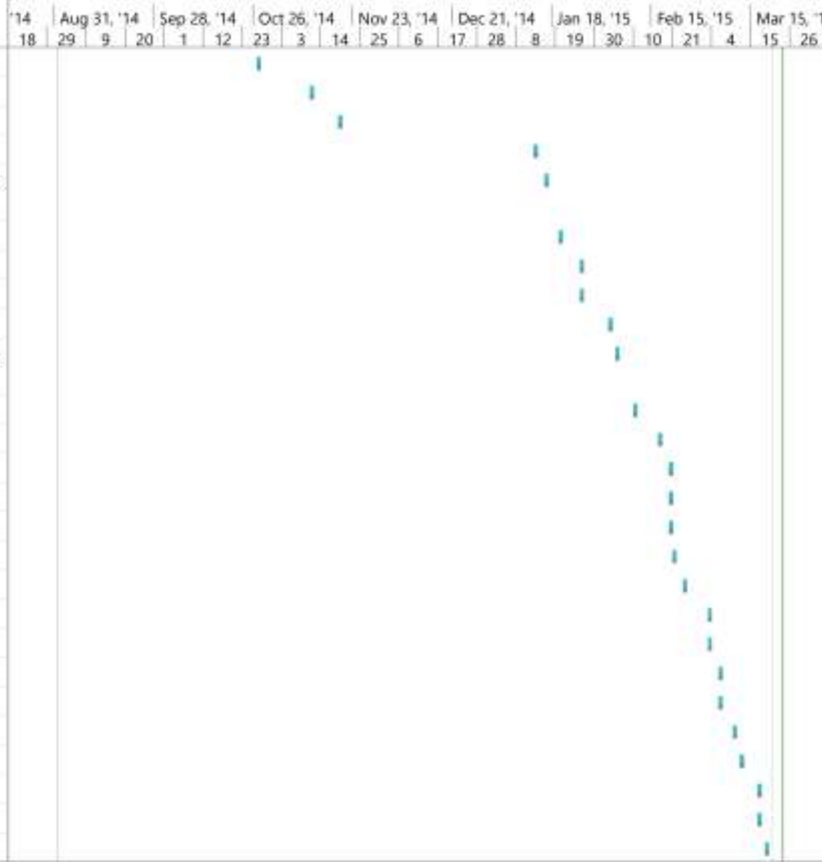


Page 1

ID	Task Name
35	**Hardware selection and testing for primary objective validation**
43	**Measure Camshaft**
44	**Re-Measure Camshaft**
41	Find and order solenoids
38	~~Reasmble Engine~~
39	Find rotary sensor
40	Verify Idle RPM
42	Find solenoid spring
37	Test Solenoids
36	Test Hardware (Sensors)
45	Test and Tune Actuators
46	Validation
47	**Secondary objective**
52	**Hardware testing**
54	Determine sensors
53	Get baseline engine specs
55	Test exhaust temperature sensor
57	~~Test O2 sensor~~
56	Test MAP sensor
58	**Software design for testing**
61	Design valve map
59	LCD + Interface box SW
60	Software to run carburator
48	**Moderling And Fabrication**
50	Fabricate Motor Test stand
51	Setup for sensors
49	Control console
62	Carburator engine test
63	~~Carburated engine and valve map test~~
64	**Tests**
65	Measured Camshaft on lathe
67	SW Solenoid Driver design
68	Arduino time delay
69	testing rotary sensor analog output
70	Testing rotary sensor SPI output
71	Validate DC motor RPM
72	Design of driver circuit and test



ID	Task Name
73	Test solenoid with DC motor
74	Linear sensor + counter
75	Testing solenoid with Current controller
76	Testing PCB's
66	Re-meausred camshaft on mill with digital dial indicators and rotary table
77	Testing solenoid on engine head
78	SW testing of VBA data collection and transformation
79	Tested solenoid system delay
80	Fixing Linear sensor drift
81	Testing flash memory for test data storage to reduce load on serial prints
83	VBA Excel integration
82	Testing box buttons + LCD
84	LCD testing
85	Sensor drift
86	Map sensor test
87	optimize code
88	Testing storing data function
89	Validation testing complete
90	Camshaft testing on engine
91	Camshaft testing one engine day #2
92	camless engine test day #1
93	camless engine test day #2
94	camless engine test day #3
95	camless engine test day #4
96	Cylinder pressure test
97	Intake solenoid operation + camshaft exhaust



65 **Measured Camshaft on lathe**
Mapped 720 degrees with 3 degree increments

67 **SW Solenoid Driver design**
PWM cannot be divided by floating point, it cannot go below 20 hz and must use on-off system based on crank position

68 **Arduino time delay**
Inrease Serial print baud rate to reduce system delays and account for solenoid delays.

69 **testing rotary sensor analog output**
Analog output ramp was ok. Rotary sensor working.

70 **Testing rotary sensor SPI output**
SPI output works to arduino, RPM function not working very accuruate (solved in February), possibly use DAC to go to arduino, to limit listening time on SPI.

71 **Validate DC motor RPM**
Success, used strobescope to verify SW and HW RPM

72 **Design of driver circuit and test**
Driver circuit needs to operate at full open >5v

73 **Test solenoid with DC motor**
System works up to 3000 RPM - need to test with linear. Possibly use engine velocity and acceleration to predict valve events.

74 **Linear sensor + counter**
Needed to add counter to reduce system load on arduino. Linear sensor reads 512 pulses/mm

75 **Testing solenoid with Current controller**
Our solenoids are too non linear to use current controller. Current control would allow for implementation of PID control for system.

76 **Testing PCB's**
Rotary board had issues with motor + magnet alignment issues. New motor holder 3D printed. Linear board had bad via solder and one track was not milled properly. All boards working.

66 **Re-meausred camshaft on mill with digital dial indicators and rotary table**
Increased precision to 1 degree increments. For camless valve system 0.5mm is considered open

77 **Testing solenoid on engine head**
Solenoids move slowly after time. Issue was due to solenoid plunger being too far away from core. Add 5.5mm 3d printed spacer to bring solenoid closer.

78 **SW testing of VBA data collection and transformation**
Data can be collected from Arduino Serial port and transformed to graphs.

79 **Tested solenoid system delay**
Added 16ms offset to opening. Delay * RPM = crankpos offset for valve event.

80 **Fixing Linear sensor drift**
Found solenoid holders moving; some drift removed.

81 **Testing flash memory for test data storage to reduce load on serial prints**
Did not work; cannot write to progrmemm after compile.

83 **VBA Excel integration**
Made a VBA functions to perform integration on data captured from arduino.

82 **Testing box buttons + LCD**
Emergency stop button working, and interface buttons working with LCD and other hardware.

84 **LCD testing**
Fixed start menu to see crank position and intake/exhaust lift for pre operation hecks.

85 **Sensor drift**

Found solenoid return spring too fast for linear sensor. Return speed ~1.22m/s; sensor is max 0.6 m/s

86 **Map sensor test**

2 bar Map sensor tested using an oscilloscope. Functions properly.

87 **optimize code**

Reduced SW task frequency of non essential tasks to allow for greater sampling of crank position and sampling. Can look into predicting valve event based of RPM and acceleration.

88 **Testing storing data function**

Storing date functions changed to dynamic arrays, as to not crash CPU if RAM is used up.

89 **Validation testing complete**

System works up to 2000 RPM with little decrease in performance.

90 **Camshaft testing on engine**

TDC was innacurate in some tests, fixed for Friday. Pressure sensor was not working. Bad coax cable.

91 **Camshaft testing one engine day #2**

Finished testing for various RPM as baseline stats. Recorded cylinder pressure, TDC, and MAP.

92 **camless engine test day #1**

No continous ignition.

93 **camless engine test day #2**

Tests T0220-T0229. Seems no fuel is being sucked in, less vacuum than camshaft engine.

94 **camless engine test day #3**

Tests T0230-T0257. Best run 8 seconds; different valve timing. Engine does not seem to pull vacuum.

95 **camless engine test day #4**

Tests T0259-T0271. one good run, however most have intermittent combustion. Fixed vacuum issue, caused by SW not account for system delays correctly. Exhaust valve does not seem to open.

96 **Cylinder pressure test**

Found valves will not open about 7 PSI exhaust gas residual.

97 **Intake solenoid operation + camshaft exhaust**

Found increasing intop to reduce overlap increased motor rpm, increasing intake close after BDC causes engine to starve and reduces engine rpm.

APPENDIX K : MICRCOCONTROLLER CODE

```
#include<avr/pgmspace.h>     //flash memory
#include <LiquidCrystal.h>   // include the LCD library
#include "TimerThree.h"

LiquidCrystal lcd(28,26, 30, 32, 34 , 36, 38);

#include "SPI.h"             // for SPI
double result = 0;
unsigned int result1 = 0;
unsigned int result2 = 0;

int Exhaust_Solenoid = 22;   // Declare pins
int Intake_Solenoid = 24;
int blink_pin = 13;          // for testing timing
int SScounter1 = 48;
int SScounter2 = 46;

int servo_pin = 3;
int TDC_pin = 14 ;

signed long encoder1count = 0; //Hold count for encoder
signed long encoder2count = 0;

double rpm = 5.0;           // initialize variables of inputs
double rpm_crank_init = 0;
double rpm_time_init = 0;
double MaxL = 0.0;           // lcd
double Mo = 0.0;             // lcd mode
unsigned long lastnow=0, now=0, timing=0;    // timing
volatile int crankpos=0;
volatile double last_crankpos=0.0;
volatile double crank_diff;
volatile bool rollover = false; //checks if crankpos >360
const int TDC = 310;
int Crankpos_offset = 310 - 215;// offset for engine crankpos
const int exop = 99 -20;     //exhaust valve open
const int excl = 331 -0 ;    // exhaust valve close -22
 int intop = 307 + 0 ;       // intake valve open
 int intcl = 554 - 70;       // intake valve close -7

unsigned long RPM_START_TIME; // in check_crankpos() to
unsigned long RPM_START_TIME_LAST; //calculate rpm
```

```
int INSTANT_RPM = 0;
int AVERAGE_RPM = 0;
double crank_position_last = 0.0;

long offset_on;              // calculating offsets
long offset_off;
int intop_ACT;
int intcl_ACT;
int exop_ACT;
int excl_ACT;
bool added720Exhaustopen=0;
bool added720Exhaustclose=0;

volatile unsigned long time_now=0, last_time;

volatile bool update_720 = 0;
bool rpm_zero = 0;

// flash memory
/*
PROGMEM prog_uint32_t time[2000];
PROGMEM prog_uint16_t crank_pos[2000];
PROGMEM prog_int32_t exhaust_solenoid_pos[2000];
PROGMEM prog_uchar exhaust_solenoid_cmd[2000];
PROGMEM prog_int32_t intake_solenoid_pos[2000];
PROGMEM prog_uchar intake_solenoid_cmd[2000];
*/


int mode_toggle_SW1 = 7;     // Declare pins for BUTTONS
int mode_toggle_SW2 = 6;

int throttle_pin = 8;
int rpm_CMD;

int start_init;              // SWITCH flags
int run_init;
int cam_init;

bool right_pressed = 0; // setting cursor with interrupt
bool left_pressed = 0;
bool up_pressed = 0;
```

```
bool down_pressed = 0;
bool enter_pressed = 0;

int cursor_position_col = 0;
int cursor_position_row = 0;
int ROW = 0;

int manual_exop = 0;
int manual_excl = 0;
int manual_inop = 0;
int manual_incl = 0;


int previous_ROW = 0;
int previous_curs_col = 0;
int previous_curs_row = 0;

//serial print stuff
//String serialprintstuff;

void setup()
{       Serial.begin(115200); // for testing 500000 works
        //  LCD_setup();
// pinMode(A1,INPUT);      // analog inputting the
potentiometer at the moment. We want this to be the
crankshaft position sensor. This is what we're going to be
using for testing
        spi_setup();

        pinMode(Exhaust_Solenoid, OUTPUT);
        pinMode(Intake_Solenoid, OUTPUT);

        clearEncoderCount();
        //Serial.println("starting test245");

        pinMode(mode_toggle_SW1, INPUT);    // buttons
        pinMode(mode_toggle_SW2, INPUT);

        pinMode(throttle_pin, INPUT);       //throttle
        pinMode(servo_pin, OUTPUT);
        pinMode(TDC_pin, OUTPUT);

        attachInterrupt(2,right_fun, RISING); // interupt
        attachInterrupt(5, left_fun, RISING);
        attachInterrupt(4, up_fun, RISING);
        attachInterrupt(3, down_fun, RISING);
        attachInterrupt(0, enter_fun, FALLING);


        start_init = 1;
        run_init = 1;
        cam_init = 1;

// START-UP MODE
while (digitalRead(mode_toggle_SW1) == HIGH &&
digitalRead(mode_toggle_SW2) == LOW)
        {
                while (start_init) // display start lcd once
                {
                        LCD_setup_start();
                        start_init = 0;
                }
                LCD_START_UPDATE();

        //Serial.print("Inside_start,");
        //Serial.print(digitalRead(mode_toggle_SW1));
        //Serial.print(",");
        //Serial.println(digitalRead(mode_toggle_SW2));

                if (down_pressed == 1)
{       digitalWrite(Intake_Solenoid,
!digitalRead(Intake_Solenoid));
down_pressed = 0;
                        Serial.println("intake loop");
                }
                if (up_pressed == 1)
{       digitalWrite(Exhaust_Solenoid,
                                !digitalRead(Exhaust_Solenoid));
                        up_pressed = 0;
                        Serial.println("exhaust loop");

                }
                if (enter_pressed == 1)
{       clearEncoderCount();
                        enter_pressed = 0;
                        Serial.println("zeroed loop");
                }
```

```
		static int throttle_pos = 0;
		//analogWrite(servo_pin,
(analogRead(throttle_pin)/4));
//Serial.println((analogRead(throttle_pin)
/ 4));
		// include setting throttle to zero
		throttle_pos = analogRead(throttle_pin);
		throttle_pos = map(throttle_pos, 0, 1023,
145,255 );	// input range from 0 to 1023 is converted to
rpm scale from 0 to 2500
analogWrite(servo_pin, throttle_pos);

		// open valves
		//Serial.println("high");
		//digitalWrite(Exhaust_Solenoid, HIGH);
		//digitalWrite(Intake_Solenoid, HIGH);
		delay(100);
	}

// RUN MODE
	while (digitalRead(mode_toggle_SW1) == HIGH  &&
digitalRead(mode_toggle_SW2) == HIGH)

		static unsigned long last_time_1 = 0;
		static unsigned long last_time_2 = 0;
		static unsigned long last_time_3 = 0;
		static int throttle_pos = 0;

		while (run_init)
			{	LCD_setup();
				run_init = 0;
			}

		throttle_pos = analogRead(throttle_pin);

intop = map(throttle_pos, 0, 1023, 260, 400);
// input range from 0 to 1023 is converted to rpm scale from
0 to 2500

		// analogWrite(servo_pin, throttle_pos);

		if ((RPM_START_TIME - (last_time_1)) > 500000)
```

```
{	// lcd.setCursor(11, 0);
			//lcd.print((int)result); // Start at 11
			//lcd.print("   ");
			// Serial.println(result);
			//Serial.print("last time:");
			//Serial.println((last_time_1));
			//last_time_1 = RPM_START_TIME;
			//Serial.print("rpm time:");
			//Serial.println(RPM_START_TIME);
			//Serial.println(crankpos);

update_LCD();

	// 2ms loop time -- need more data
		for (int x = 0; x <= 10; x++)
{	calc_crank();
			CALC_RPM();
			//Serial.println(INSTANT_RPM);
			valve_offset();
			//Serial.print("crankpos");
			//Serial.println(crankpos);
			valve_event();
			//if (x == 0)  Serial.print(micros());
			//if (x == 1) Serial.print(',');
			//if (x== 2) Serial.print(crankpos);
			//if (x == 3) Serial.print(',');
			//if (x == 4) encoder1count =
//			readEncoder(1);
			//if (x == 5)
			//	Serial.println((encoder1count));
			//if (x == 6) Serial.print(',');
			//if (x == 7) encoder2count =
			//	readEncoder(2);
			//if (x == 8)
			//	Serial.println(encoder2count);
			//if (x == 9) Serial.print(',');
			//if (x == 10){
			// Serial.println(digitalRead(
Intake_Solenoid));
			//	}
			//	*/
		}

		if (enter_pressed == 1)
```

```
{       rollover = !rollover;
                enter_pressed = 0;
        }

   /*  if ( update_720 = 1)  // add = here
        {       static int rpm_counter = 0;
                update_LCD();
                //Serial.println(crankpos);
                if (right_pressed || left_pressed ||
up_pressed || down_pressed)
                {       update_button_status();
                        Serial.print("row: ");
      Serial.print(cursor_position_row);
                        Serial.print(", col ");
                        Serial.println(cursor_position_col);
                        update_cursor_location();
                }
        if (cursor_position_row >= 3 && enter_pressed)
        {       update_button_functions();
}
//valve_offset(); // add rpm based offset for valve timing
        if (rpm_counter == 20)
{       throttle_pos();
                rpm_counter = 0;
        }
        rpm_counter++;
    }   */
    //count++;
    }

/// RUNNING ENGINE WITH CAMSHAFT
    while (digitalRead(mode_toggle_SW1) == LOW &&
digitalRead(mode_toggle_SW2) == HIGH
    {
        static unsigned long last_time_1=0;
        static unsigned long last_time_2=0;
        static unsigned long last_time_3=0;

        while (cam_init)
        {       LCD_setup_cam();
                cam_init = 0;
        }

        check_crank_pos();
```

```
                CALC_RPM();

                static int throttle_pos = 0;

//analogWrite(servo_pin,
//      (analogRead(throttle_pin)/4));
                //Serial.println((analogRead(throttle_pin)/
//      4));

                throttle_pos = analogRead(throttle_pin);


                throttle_pos = map(throttle_pos, 0, 1023, 145,
255);    // input range from 0 to 1023 is converted to rpm
scale from 0 to 2500

                analogWrite(servo_pin, throttle_pos);

                if ((RPM_START_TIME - (last_time_1)) > 500000)
{       lcd.setCursor(11, 0);
                        lcd.print((int)result); // start at 11
                        lcd.print("   ");
                        // Serial.println(result);
                        Serial.print("last time:");
                        Serial.println((last_time_1));
                        last_time_1 = RPM_START_TIME;
                        Serial.print("rpm time:");
                        Serial.println(RPM_START_TIME);
                }

                if (RPM_START_TIME > (last_time_2 + 570000))
{       lcd.setCursor(9, 1);
                        lcd.print(INSTANT_RPM); //instant rpm
                        lcd.print("   ");
                        //Serial.print("inst:");
                        last_time_2 = RPM_START_TIME;
                        //Serial.println(CALC_RPM(1));
                }

                if (RPM_START_TIME > (last_time_3 +
                        600000))
{       lcd.setCursor(9, 2); //PRINT AVERAGE RPM
                        lcd.print(AVERAGE_RPM);
                        lcd.print("   ");
```

```
                        last_time_3 = RPM_START_TIME;
                        //Serial.print("avg:");
                        //Serial.println(CALC_RPM(0));
                }
        }

}               // end of main loop


//Functions

int loop_time()
{       static long last_time;
        static  long time = 0;
        static int count = 0;

        last_time = time;
        time = micros();

        Serial.println(time - last_time);
}

double calc_crank()
{       last_crankpos = crankpos;
        crankpos = check_crank_pos();

        if (rollover) crankpos = crankpos + 360.0;
        if (((crankpos - last_crankpos) < -2))
{       if (rollover ) // if rollover from 720 to 360
{       crankpos = crankpos - 360;
                        //calculate rpm
rpm = (((720 + crankpos –
rpm_crank_init)*500000.0) / 3.0)
 / ( - rpm_time_init);
//store initial time
                        rpm_time_init = micros();
//store initial crankpos
                        rpm_crank_init = crankpos;

                        update_720 = 1;
                        //update_LCD();

                }
                if (!rollover)  // if rollover from 360 to 0
```

```
{       crankpos = crankpos + 360;
}

rollover = !rollover;
        }
        /*
        if (crankpos > (TDC-2) && (crankpos < (TDC +2)))
{       digitalWrite(TDC_pin, HIGH);
                //Serial.print(crankpos);
                //Serial.println("high");
        }
        else { digitalWrite(TDC_pin, LOW);
                //Serial.print(crankpos);
                //Serial.println("low");
              }
        */

return crankpos;
}

int valve_offset() // calculate offset outside of valve event
{       CALC_RPM();

        added720Exhaustopen=0;
        added720Exhaustclose=0;

        offset_on = ((double)INSTANT_RPM * 360.0 * 13.0) /
60000.0; // rpm in deg/ms with 16 ms delay
        offset_off = ((double)INSTANT_RPM * 360.0 * 18.0 ) /
60000.0; // rpm in deg/ms with 20 ms delay

        /*
        Serial.print("rpm:");
        Serial.println(INSTANT_RPM);
        Serial.print("offset:");
        Serial.println(offset_off);
        */

        // add offset to intake and exhaust
        intop_ACT = (intop + manual_inop) - offset_on;
// check if offset is negative
        if (intop_ACT < 0) {intop_ACT= 720 + intop_ACT;}

        intcl_ACT = (intcl + manual_incl)- offset_off;
```

```
	if (intcl_ACT < 0) {intcl_ACT= 720 + intcl_ACT;}

	exop_ACT = (exop + manual_exop) - offset_on;
	if (exop_ACT < 0)
{	exop_ACT= 720 + exop_ACT;
	added720Exhaustopen =1;
	}

	excl_ACT = (excl + manual_excl) - offset_off;
	if (excl_ACT < 0)
{	excl_ACT= 720 + excl_ACT;
		added720Exhaustclose =1;
	}

	return 0;
}

int valve_event()
{
	if ( (crankpos > intop_ACT) && (crankpos < intcl_ACT))
{ digitalWrite(Intake_Solenoid, HIGH); }
	else	{ digitalWrite(Intake_Solenoid, LOW); }

	if (added720Exhaustopen && !added720Exhaustclose)
{	if ( crankpos > (exop_ACT) || crankpos <
(excl_ACT) )
{digitalWrite(Exhaust_Solenoid, HIGH);}
		else	{digitalWrite(Exhaust_Solenoid, LOW);}
	}
	else
{	if ( crankpos > (exop_ACT) && crankpos <
(excl_ACT) )
{digitalWrite(Exhaust_Solenoid, HIGH);}
		else	{digitalWrite(Exhaust_Solenoid, LOW); }
	}

	return 0;
}


void LCD_setup(){
		lcd.begin(20, 4);	// LCD 20 cols and 4 rows
		lcd.setCursor(0, 0);
		lcd.print("RPM:");
			lcd.setCursor(11, 0);
			lcd.print("CMD:");
			lcd.setCursor(0, 1);
			lcd.print("EO:");
			lcd.print(exop);
			lcd.setCursor(7, 1);
			lcd.print("+0");// changes with interrupt
			// manual change of exhaust opening set to 0
manual_exop = 0;
			lcd.setCursor(10,1);
			lcd.print("EC:");
			lcd.print(excl);
			lcd.setCursor(17, 1);
			lcd.print("+0");
			manual_excl = 0;
			lcd.setCursor(0, 2);
			lcd.print("IO:");
			lcd.print(intop);
			lcd.setCursor(7, 2);
			lcd.print("+0");
			manual_inop = 0;
			lcd.setCursor(10, 2);
			lcd.print("IC:");
			lcd.print(intcl);
			lcd.setCursor(17, 2);
			lcd.print("+0");
			manual_incl = 0;
			lcd.setCursor(0, 3);
			lcd.print("   Functions   ");
}

void LCD_setup_cam()
{	lcd.begin(20, 4);
	lcd.setCursor(0, 0);
	lcd.print("CRANK POS:");
	lcd.setCursor(0, 1);
	lcd.print("INST RPM:");
	lcd.setCursor(0, 2);
	lcd.print("AVG RPM:");
}

void LCD_setup_start()
{	lcd.begin(20, 4);
	lcd.setCursor(0, 0);
```

```
        lcd.print("CRANK POS:");
        lcd.setCursor(0, 1);
        lcd.print("EX LIFT:");
        lcd.setCursor(0, 2);
        lcd.print("IN LIFT:");
        lcd.setCursor(0, 3);
        lcd.print("  Functions   ");
}

int update_LCD()
{       static unsigned int update = 0;
        if(update = 1)
{       lcd.setCursor(4, 0);
                lcd.print(INSTANT_RPM);
                lcd.print(" ");
        }
        if (update = 1)
        {       lcd.setCursor(15, 0);
                lcd.print((int)intop);
                lcd.print(" ");
        }
        //Serial.println(update % 4);
        update++;
        return 0;
}


int LCD_START_UPDATE()
{
        lcd.setCursor(10, 0);
        lcd.print((int)check_crank_pos());
        lcd.print("   ");
        lcd.setCursor(8, 1);
        lcd.print(readEncoder(1));
        lcd.print("     ");
        lcd.setCursor(8, 2);
        lcd.print(readEncoder(2));
        lcd.print("     ");
        lcd.setCursor(0, 3);
        lcd.print("  Functions    ");
}


int max_lift(int lift)
{       if (lift > MaxL) { MaxL= lift; }
        return MaxL;
}


// setup for SPI communication -- used for angle sensor
void spi_setup()
{       // Set slave selects as outputs
        pinMode(SScounter1, OUTPUT);
        pinMode(SScounter2, OUTPUT);
        // Raise select pins
        // Communication begins when you drop the individual
select signsl


SPI.begin();
SPI.setBitOrder(MSBFIRST);
SPI.setDataMode (SPI_MODE0) ;

// Initialize encoder 1
                // Clock division factor: 0
                // Negative index input
                // free-running count mode
                // x4 quatrature count mode (four counts per
quadrature cycle)
                // NOTE: For info on commands, see datasheet
        digitalWrite(SScounter1,LOW); // Begin SPI convo
        SPI.transfer(0x88); // Write to MDR0
        SPI.transfer(0x03); // Configure to 4 byte mode
        digitalWrite(SScounter1,HIGH); // Terminate SPI convo
        // Initialize encoder 2
        // Clock division factor: 0
        // Negative index input
        // free-running count mode
        // x4 quatrature count mode (four counts per
quadrature cycle)
        // NOTE: For info on commands, see datasheet
        digitalWrite(SScounter2,LOW); // Begin SPI convo
        SPI.transfer(0x88); // Write to MDR0
        SPI.transfer(0x03); // Configure to 4 byte mode
        digitalWrite(SScounter2,HIGH); // Terminate SPI convo

SPI.setDataMode (SPI_MODE1) ;
}
```

```
// read SPI signal for angle sensor
double check_crank_pos()
{       SPI.setDataMode (SPI_MODE1);
        //crank_position_last = result;
        RPM_START_TIME = micros();
        digitalWrite(SS, LOW);

        //Reading 8 bit frame
// (ie. the first half of 16-bit SPI transfer)
        result1 = SPI.transfer(0b00000000);
        //Serial.print("byte 1: ");
        //Serial.println(result1, BIN);

        // removing (masking) first 2 bit
        result1 &= 0b00111111;
// Serial.print("byte 1 masked: ");
        // Serial.println(result1, BIN);

        //shifting 8 bit to left to create empty space for
//last 8 bit transfer
result1 = result1 << 8;
// Serial.print("byte 1 shifted: ");
        //Serial.println(result, BIN);

        // getting last 8 bits
// (ie.the last half of 16-bit SPI transfer)
        result2 = SPI.transfer(0b00000000);
        //Serial.print("byte 2: ");
        //Serial.println(result2, BIN);

        // merging
        result = result1 | result2;
        // Serial.print("Result: ");
        //Serial.print(result, BIN);
        //Serial.print(" = ");
        //Serial.println(result, DEC);
        //Serial.println();

digitalWrite(SS, HIGH);

result = (abs((result*360.0 / 16384) - 360)+
Crankpos_offset);

//Serial.println(result);

if (result > 360) {result = result - 360;}
if (result > (TDC - 5) && (result < (TDC + 5)))
{       digitalWrite(TDC_pin, HIGH);
                //Serial.print(crankpos);
                //Serial.println("high");
}
else
{       digitalWrite(TDC_pin, LOW);
                //Serial.print(crankpos);
                //Serial.println("low");
}

return result; // convert to degrees
}


//RPM_TYPE 1 = INSTANT
//RPM_TYPE 0 = AVERAGE
int CALC_RPM()
{       static double crank_position_last_avg = 0;
        static double RPM_START_TIME_LAST_AVG = 0;

        if (RPM_START_TIME > (RPM_START_TIME_LAST +5000))
{       // IF 5MS HAVE PASSED

                static double INSTANT_RPM_STORE = 0;
                double delta_crank;

                delta_crank = result - crank_position_last;

                // CHECK FOR ROLLOVER
                if (delta_crank < 0) delta_crank += 360;
                if (delta_crank < 0.4)
{       INSTANT_RPM = 0;
                        return 0;
//if the crankshaft moved less than 10
degrees, output 0 rpm
                }

                INSTANT_RPM_STORE = INSTANT_RPM;
                INSTANT_RPM = ((delta_crank)*500000.0 / 3.0) /
      ((RPM_START_TIME - RPM_START_TIME_LAST));
```

```
		if ((abs(INSTANT_RPM - INSTANT_RPM_STORE))>200)
{INSTANT_RPM = INSTANT_RPM_STORE;}

		//Serial.println(INSTANT_RPM);
		crank_position_last = result;
		RPM_START_TIME_LAST = RPM_START_TIME;

		return INSTANT_RPM;
	}

	if (RPM_START_TIME > (RPM_START_TIME_LAST_AVG +
10000)) // for average rpm check every 10 ms
{	 double delta_crank;

		 delta_crank = result -crank_position_last_avg;

		 // CHECK FOR ROLLOVER
		if (delta_crank < 0) delta_crank += 360;
		if (delta_crank < 0.2)
{	AVERAGE_RPM = 0;
			return 0;
//if the crankshaft moved less than 10
degrees, output 0 rpm
		}

		 RPM_START_TIME_LAST_AVG = RPM_START_TIME;
		 crank_position_last_avg = result;

		 return AVERAGE_RPM;
	 }

	 return 0;
}


long readEncoder(int encoder)
{	SPI.setDataMode (SPI_MODE0) ;

	// Initialize temporary variables for SPI read
	unsigned int count_1, count_2, count_3, count_4;
	long count_value;

	if (encoder == 1) // Read encoder 1
{	digitalWrite(SScounter1,LOW); //Begin SPI convo
		SPI.transfer(0x60); // Request count
		count_1 = SPI.transfer(0x00); //Read highest
order byte
		count_2 = SPI.transfer(0x00);
		count_3 = SPI.transfer(0x00);
		count_4 = SPI.transfer(0x00); // Read lowest
						order byte
		digitalWrite(SScounter1,HIGH);//Term. SPI convo
	}
	else if (encoder == 2) // Read encoder 2
{	digitalWrite(SScounter2,LOW);
		SPI.transfer(0x60);
		count_1 = SPI.transfer(0x00);
		count_2 = SPI.transfer(0x00);
		count_3 = SPI.transfer(0x00);
		count_4 = SPI.transfer(0x00);
		digitalWrite(SScounter2,HIGH);
	}

	// Calculate encoder count
	count_value = (count_1 << 8) + count_2;
	count_value = (count_value << 8) + count_3;
	count_value = (count_value << 8) + count_4;
	return count_value;
}


void clearEncoderCount()
{	SPI.setDataMode (SPI_MODE0) ;
	// Set encoder1's data register to 0
	digitalWrite(SScounter1,LOW); // Write to DTR
	SPI.transfer(0x98); // Load data
	SPI.transfer(0x00);
	SPI.transfer(0x00);
	SPI.transfer(0x00);
	SPI.transfer(0x00);
	digitalWrite(SScounter1,HIGH);
// provides some breathing room between SPI convos
	delayMicroseconds(100);
	// Set encoder1's current data register to center
```

```
        digitalWrite(SScounter1,LOW); // Begin SPI convo
        SPI.transfer(0xE0);
        digitalWrite(SScounter1,HIGH);
        // Set encoder2's data register to 0
        digitalWrite(SScounter2,LOW);
        SPI.transfer(0x98);
        SPI.transfer(0x00);
        SPI.transfer(0x00);
        SPI.transfer(0x00);
        SPI.transfer(0x00);
        digitalWrite(SScounter2,HIGH);
        delayMicroseconds(100);
        // Set encoder2's current data register to center
        digitalWrite(SScounter2,LOW);
        SPI.transfer(0xE0);
        digitalWrite(SScounter2,HIGH);
}

void right_fun() { right_pressed = 1; }

void left_fun()  { left_pressed = 1; }

void up_fun()    { up_pressed = 1; }

void down_fun()  { down_pressed = 1; }

void enter_fun() { enter_pressed = 1; }

void update_button_status()
{
if (right_pressed)
        {       previous_curs_row = cursor_position_row;
                previous_curs_col = cursor_position_col;
                cursor_position_col += 1;
                if (cursor_position_col == 2)
                        {cursor_position_col = 0; }
right_pressed = 0;
        }
        else if (left_pressed)
        {       previous_curs_row = cursor_position_row;
                previous_curs_col = cursor_position_col;
                cursor_position_col -= 1;
                if (cursor_position_col < 0)
                        { cursor_position_col = 0;}
                left_pressed = 0;}

else if (up_pressed)
        {       if (enter_pressed) {update_button_valve_up();}
else
{       previous_curs_row = cursor_position_row;
                        previous_curs_col = cursor_position_col;
                        cursor_position_row -= 1;
                        if (cursor_position_row < 0)
                        {cursor_position_row = 0;}
}
up_pressed = 0;
        }
        else if (down_pressed)
        {       if (enter_pressed){update_button_valve_down();}
else
                {       previous_curs_row = cursor_position_row;
                        previous_curs_col = cursor_position_col;
                        cursor_position_row += 1;
                        if (cursor_position_row > 6)
                                {cursor_position_row = 6;}
                }
                down_pressed = 0;
        }

        if (!enter_pressed)
        {       if (cursor_position_row == 3)
                {       lcd.setCursor(0, 3);
                        lcd.print(" Test valve         ");
                }
                else if (cursor_position_row == 4)
                {       lcd.setCursor(0, 3);
                        lcd.print("  function 2    ");
                }
                else if (cursor_position_row == 5)
                {       lcd.setCursor(0, 3);
                        lcd.print("   function 3       ");
                }
        }
}
```

```
void update_cursor_location()
{
	static int arrow_pos[6][2] = { {6,6}, {6,16}, {6,16},
	                               {0,0}, {0,0}, {0,0},};

	previous_ROW = ROW;
	ROW = cursor_position_row;
	if (ROW > 3)  { ROW = 3; }
	lcd.setCursor((arrow_pos[previous_curs_row]
[previous_curs_col]), previous_ROW);
	lcd.print(' ');

	if (ROW >= 1)
	{	lcd.setCursor((arrow_pos[cursor_position_row]
[cursor_position_col]), ROW);
		lcd.print((char)126);
	}
}


void update_button_valve_up()
{
	if (cursor_position_col==0 && cursor_position_row==1)
	{	manual_exop += 1;	// EVO
		if (manual_exop < 0) // negative sign
		{	lcd.setCursor(7, 1);
			lcd.print(manual_exop);
		}
		else
		{	lcd.setCursor(7, 1);
			lcd.print("+");
			lcd.print(manual_exop);
		}
	}
	else if(cursor_position_col==1 &&
cursor_position_row==1)
	{	manual_excl += 1;

		if (manual_excl < 0)
		{	lcd.setCursor(17, 1);
			lcd.print(manual_excl);
		}
		else
		{
			lcd.setCursor(17, 1);
			lcd.print("+");
			lcd.print(manual_excl);
		}
	}
	else if (cursor_position_col == 0 &&
cursor_position_row == 2)
	{	manual_inop += 1;

		if (manual_inop < 0)
		{	lcd.setCursor(7, 2);
			lcd.print(manual_inop);
		}
		else
		{	lcd.setCursor(7, 2);
			lcd.print("+");
			lcd.print(manual_inop);
		}
	}
	else if (cursor_position_col == 1 &&
cursor_position_row == 2)
	{	manual_incl += 1;

		if (manual_incl < 0)
		{	lcd.setCursor(17, 2);
			lcd.print(manual_incl);
		}
		else
		{	lcd.setCursor(17, 2);
			lcd.print("+");
			lcd.print(manual_incl);
		}
	}

	enter_pressed = 0;
}


void update_button_valve_down()
{
	if (cursor_position_col==0 && cursor_position_row==1)
	{	manual_exop -= 1;

		if (manual_exop < 0)
```

```
		{	lcd.setCursor(7, 1);
			lcd.print(manual_exop);
		}
		else
		{	lcd.setCursor(7, 1);
			lcd.print("+");
			lcd.print(manual_exop);
		}
	}
	else if (cursor_position_col == 1 && cursor_position_row == 1)
	{	manual_excl -= 1;

		if (manual_excl < 0)
		{	lcd.setCursor(17, 1);
			lcd.print(manual_excl);
		}
		else
		{	lcd.setCursor(17, 1);
			lcd.print("+");
			lcd.print(manual_excl);
		}
	}
	else if (cursor_position_col == 0 && cursor_position_row == 2)
	{	manual_inop -= 1;

		if (manual_inop < 0)
		{	lcd.setCursor(7, 2);
			lcd.print(manual_inop);
		}
		else
		{	lcd.setCursor(7, 2);
			lcd.print("+");
			lcd.print(manual_inop);
		}
	}

	else if (cursor_position_col == 1 && cursor_position_row == 2)
	{	manual_incl -= 1;

		if (manual_incl < 0)
		{	lcd.setCursor(17, 2);
			lcd.print(manual_incl);
		}
		else
		{	lcd.setCursor(17, 2);
			lcd.print("+");
			lcd.print(manual_incl);
		}
	}

	enter_pressed = 0;
}


void update_button_functions()
{
	if (cursor_position_row == 3) // do function 1
	{	Serial.println(" entered 3");
		lcd.setCursor(0, 3);
		lcd.print(" Running Test ");

		test_valve_pos();
		lcd.setCursor(0, 3);
		lcd.print(" Test valve   ");
		}
	else if (cursor_position_row == 4) // do function 2
	{	Serial.println(" entered 4 ");
		lcd.setCursor(0,3);
		lcd.print("     Team 21     ");
	}
	else if (cursor_position_row == 5) // do function 3
	{	Serial.print(" entered 5 ");
		lcd.setCursor(0, 3);
		lcd.print("     Team 21     ");
	}

	enter_pressed = 0;
}

void test_valve_pos()		// TESTING
{
	long* Time_us = 0;
	int* Crank_deg = 0;
	int* Intake_valve_pos_ACT = 0;
	bool* Intake_valve_pos_CMD = 0;
```

```
	int* Exhaust_valve_pos_ACT = 0;
	bool* Exhaust_valve_pos_CMD = 0;

	int x = 0;		// clear encoder count
	long start_time;

	const int Data_Size = 500;
	Time_us = new long[Data_Size];
// cast some items to int to reduce size
	Crank_deg = new int[Data_Size];
	Intake_valve_pos_ACT = new int[Data_Size];
	Intake_valve_pos_CMD = new bool[Data_Size];
	Exhaust_valve_pos_ACT = new int[Data_Size];
	Exhaust_valve_pos_CMD = new bool[Data_Size];


	// check allocation of memory
	if (Time_us == NULL)
{	Serial.println("1memory allocatiopn error");
return;
}
	if (Crank_deg == NULL)
{	Serial.println("2memory allocatiopn error");
return;
}
	if (Intake_valve_pos_ACT == NULL)
{	Serial.println("3memory allocatiopn error");
return;
}
	if (Intake_valve_pos_CMD == NULL)
{	Serial.println("4memory allocatiopn error");
return;
}
	if (Exhaust_valve_pos_ACT == NULL)
{	Serial.println("5memory allocatiopn error");
return;
}

	if (Exhaust_valve_pos_CMD == NULL)
{	Serial.println("6memory allocatiopn error");
return;
}

	while(crankpos != 650) //check size
{	calc_crank();
		valve_offset();
		valve_event();
	}

	clearEncoderCount();

	calc_crank();
	valve_offset();
	valve_event();

	start_time = micros();
	Time_us[x] = micros() - start_time;
	Crank_deg[x] = crankpos;
	Intake_valve_pos_ACT[x] = (int)readEncoder(2);
	Intake_valve_pos_CMD[x]=
(bool)digitalRead(Intake_Solenoid);
	Exhaust_valve_pos_ACT[x] = (int)readEncoder(1);
	Exhaust_valve_pos_CMD[x] =
(bool)digitalRead(Exhaust_Solenoid);
	x++;

	for ( x ; x < Data_Size; )
{	calc_crank();
		valve_offset();
		valve_event();

		if ( abs((crankpos - (Crank_deg[x - 1]))) > 1)
{	Time_us[x] = micros() - start_time;
			Crank_deg[x] = crankpos;
			Intake_valve_pos_ACT[x] =
     (int)readEncoder(2);
			Intake_valve_pos_CMD[x] =
     (bool)digitalRead(Intake_Solenoid);
Exhaust_valve_pos_ACT[x] =
     (int)readEncoder(1);
			Exhaust_valve_pos_CMD[x] =
    (bool)digitalRead(Exhaust_Solenoid);
			x++;
		}
	}

	for (x = 0; x < Data_Size; x++)
{	for ( int j = 0; j <= 10 ; j++)
```

```
{  calc_crank();
                valve_offset();
                valve_event();
                if (j == 0)Serial.print(Time_us[x]);
                if (j == 1) Serial.print(',');
                if (j == 2) Serial.print(Crank_deg[x]);
                if (j == 3) Serial.print(',');
                if (j == 4)
                   {Serial.print(Exhaust_valve_pos_ACT[x]);}
                if (j == 5) Serial.print(',');
                if (j == 6)
     {Serial.print(Exhaust_valve_pos_CMD[x]);}
                if (j == 7) Serial.print(',');
                if (j == 8)
     {Serial.print(Intake_valve_pos_ACT[x]);}
                if (j == 9) Serial.print(',');
                if (j == 10)
     {Serial.println(Intake_valve_pos_CMD[x]);}
}
       }

       calc_crank();
       valve_offset();
       valve_event();

       Serial.print(" , , ,");
       Serial.println(INSTANT_RPM);

       if (Time_us!=NULL){delete Time_us; Time_us = NULL;}
       else{ Serial.println("delete error");}

       if (Crank_deg!=NULL){delete Crank_deg;Crank_del=NULL;}
       else{ Serial.println("delete error"); }

       if (Intake_valve_pos_ACT != NULL)
{      delete Intake_valve_pos_ACT;
Intake_valve_pos_ACT = NULL;
}
       else{ Serial.println("delete error"); }

       if (Intake_valve_pos_CMD != NULL)
{      delete Intake_valve_pos_CMD;
       Intake_valve_pos_CMD = NULL;
}
```

```
       else{ Serial.println("delete error");}

       if (Exhaust_valve_pos_ACT != NULL)
{      delete Exhaust_valve_pos_ACT;
Exhaust_valve_pos_ACT = NULL;
}
       else{ Serial.println("delete error"); }

       if (Exhaust_valve_pos_CMD != NULL)
{      delete Exhaust_valve_pos_CMD;
       Exhaust_valve_pos_CMD = NULL;
}
       else{ Serial.println("delete error"); }

       calc_crank();
       valve_offset();
       valve_event();
}


void throttle_pos()
{
       rpm_CMD = analogRead(throttle_pin);
       static int servo_angle = 145;

       rpm_CMD = map(rpm_CMD, 0, 1023, 0, 2500);

       if(rpm < rpm_CMD)     servo_angle--;
       if (rpm > rpm_CMD)    servo_angle++;

       if (servo_angle < 145) servo_angle = 145;
       if (servo_angle > 255) servo_angle = 255;

       analogWrite(servo_pin, servo_angle);

       //Serial.print("servo:");
       //Serial.println(servo_angle);
}
```

APPENDIX L: CALCULATIONS

Spring calculation:

$$X = V_o(t) + \frac{1}{2}a(t)^2 \qquad (2)$$

where,

$X \ = \ displacement\ of\ the\ plunger\ (in)$
$Vo = Initial\ velocity\ of\ the\ plunger\ (in/s)$
$a \ = \ acceleration\ of\ the\ plunger\ (in/s^2)$
$t \ = \ time\ (s)$

$$0.25 = 0 + \frac{1}{2}(a)(0.018)^2$$

$$a = 1543.20\,\frac{in}{sec^2}\ or\ 39.19\,\frac{m}{sec^2}$$

$$F = m \cdot a \qquad (3)$$

where,

$F \ = \ force\ of\ the\ plunger\ (N)$
$m \ = \ mass\ of\ the\ plunger\ (kg)$
$a \ = \ acceleration\ of\ the\ plunger\ (m/s^2)$

$$F = (0.05) \cdot (39.19)$$

$$F = 1.96\ N$$

$$F = K \cdot \Delta x \qquad (4)$$

where,

$F \ = \ force\ of\ the\ plunger\ (N)$
$K \ = \ spring\ constant\ (N/m)$
$\Delta x \ = \ change\ in\ displacement\ (m)$

$$1.96 = K \cdot 0.00635$$

$$K = 308.58\ \frac{N}{m}\ or\ 1.76\,\frac{lbs}{in}$$