Project Number: <KZS - 1001>

Design of a Roof Inspection Robot III

A Major Qualifying Project Report

Submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

in Mechanical Engineering and in Electrical and Computer Engineering

by:

Samuel L. Drogin

Obadiah Barclay Kopchak

Daniel C. Schick

Chris

____

____

____

____

DATE:

Approved:

____

Prof. Kenneth Stafford, Advisor

____

Brad Miller, Co-Advisor

____

Prof. Taskin Padir, Co-Advisor

Acknowledgments

We would like to thank the following people for their help and support throughout the course of this project:

Nick Tuzzolo
Marius Andreica
David Wiig

Abstract

In 2006 through 2008 two teams of students developed an all-wheel drive robot capable of traversing a variety of roof geometries while visually displaying roof conditions for a sponsor. Our team performed numerous enhancements on the previous years' designs in order to grant the robot superior traction as well as an improved range and mobility, allowing the robot to operate over the crest of a roof. A new microcontroller and Wi-Fi camera were implemented to allow for two way communication between the robot and a computer, and to allow for a reliable video feed. A single man operable ascender system was designed which is able to be transported in a minivan and able to allow the robot access to a second story roof.

Executive Summary

In previous years, teams of students developed a robotic platform for performing roof inspections. The platform met many of the target goals for the project, however there were still several points which were left to be improved upon. These included designing a one-man operable ascender as well as providing two-way out of line of sight communication with the robot that includes a live video feed. Our goal is to design the ascender as well as construct the two-way communication system.

Project Task Specifications

The formal task specifications for the inspection robot and ascender system are described below.

1. Design and create an ascender that allows the robot to exit and enter roof environments. The ascender must be:
    - Operable by a single person
    - Must collapse to a length of 8ft or less to be able to fit in the back of either a truck or van used by inspectors
    - Extend to over 20ft to reach a roof with a pitch of 12/12 or less
2. Design and create a laptop based two way communication/control system that must:
    - Include a live video feed
    - Be able to operate out of line of sight throughout the area of a typical 12/12 residential roof area
3. Create a universal charging point
4. Improve traction

Table of Contents

Table of Figures

Table of Tables

Table of Equations

Introduction

When a job is dull, dangerous, or dirty, robots are there to make life a little easier. Purpose built machines can either replace or compliment a human worker's ability to get a job done. As technologies have advanced in the fields of drone and robotic technologies life has gotten safer in a range of fields. Bomb experts no longer have to touch the deadly device, pipeline inspectors can avoid going into toxic pipes, and pilots no longer have be in a physical cockpit to act as eyes in the sky, all because of rapid advancements in drone and robotic technology. This is possible due to our greater understanding of the techniques and technologies to dynamically communicate vast quantities of data between the controller and robot. While there is currently no truly standardized definition for the word robot the one generally accepted idea associated with robotics is that robots ha ve the ability interact with their environment and while exhibiting some kind of intelligent behavior.

After the chaos of disaster begins to fade, people work to get their lives back to normal. Insurance companies do their best to rapidly respond to disruptive events and help people rebuild what was lost. ~~R~~oof inspections can cost up to $1,500 per inspection. Inspecting a damaged roof is not only costly but potentially dangerous, where inspectors must traverse the roof looking for damage, and expose themselves to loose tiles, slick surfaces, or structures that can no longer properly support their weight. We will be implementing a platform to reduce costs as well as increasing inspector safety is achievable with the Roof Robot technology platform.

The complexity of the Roof Robot technology has meant that the design of the robot will be a multiyear endeavor to get to a point where the robot will be able compliment the abilities of inspectors. The original body of the robot had a design life of fifty hours of operation; as a prototype there was little guarantee that it would last longer than this time estimate. After three years since the original construction of the robot body, there is little question that the body of the robot will require some substantial repairs to ensure that it will be in as state of operational preparedness necessitated by continued development.

Our intention is to take the old roof robot pictured on the left and transform it into the new roof robot on the right.

Figure 0-1: The Alpha and the Omega Robots

1. Background

In 2006 a group of senior's at WPI began designing a roof inspection robot. This group was tasked with developing a robot that was capable of performing visual roof inspection, a task normally requiring two human inspectors. The robot has come very far in the past four years however at the beginning of this project it still did not meet all the desired specifications. The following section contains information on the initial status of that robot along with background information on how to make the robot conform to the required specifications.

1.1 Legacy Microcontroller

An important part of any robot is the "logic" it uses to make decisions. Currently the logic is controlled with a VEX microcontroller on the roof robot. This is a common microcontroller with a wide range of applications from educational robots to advanced robotics projects. However this microcontroller has several limitations.

The main limitation is that the current microcontroller only handles one way communication. The microcontroller is paired with a 75MHz RF remote control by inserting a crystal into both the microcontroller and the remote control. The remote control can be used to tell the robot where to drive, but the robot cannot tell the remote where it is, the remaining battery life, or any other information about its status. This is a major issue as we wish to document the information the robot finds as well as have real time data so that we can see what the robot sees. There is also an issue with the amount of memory on the current microcontroller. VEX 1.0 has 32Kb of available flash memory for code. The legacy code is able to fit on this microcontroller however; we would be unable to add anything to the code if we chose to stay with this microcontroller. In the following sections we will address how we fixed these issues.

1.2 Legacy Charging System

The legacy robot ran on three batteries, shown in

Figure 1-1. These batteries are 12 V 4.2 Amp hours each. Under maximum draw from the robot, climbing a 45˚ roof, these batteries last 45 min.

Figure 1-1: Batteries

The issue with the legacy system is not the batteries themselves; but rather it is the way in which they are charged and discharged. The legacy iteration has each battery being removed to be charged. This is unacceptable for a finalized product. The methodology section will explain how a single charging point was created for the robot.

1.3 Legacy Camera Design

At the beginning of the project the robot had a camera with a unidirectional antenna. This became an issue whenever the robot tried to crest a roof. As soon as the robot got over the crest the camera feed would cut out. This issue must be addressed in order to meet the current specification of operating out of line of sight. This issue will be addressed in the following sections of this report.

1.4 Legacy Sled Design

The original design for the robot transport mechanism, referred to colloquially as the sled, was a mechanically simple system that mounted to a modified 30 **ft** ladder. The most significant feature of the sled that has remained throughout the development process is the ability for the sled sub-system to pivot at the end point of the ascender. The primary concern of the Legacy Sled Design was creating a proof of concept to prove the Roof Robot's ability to deploy on a roof top environment in a mechanically simply fasion. This platform achieved its intial goal of allowing the Roof Robot to deploy on a roof environment but did meet the need for creating a platform that was able to be one man operable and assemble able. As the goal of this MQP was to design an ascender platform that would meet the single operator requirements, many characteristics that defined the Ascender Sled needed to be reanalyzed in the context of the design of the new Ascender Platform. . The challenges and solutions to these problems will be discussed further in sections 2.4 and 3.3.

1.5 Legacy Pole Design

For the standard house or commercial building there is not always guaranteed level access to the roof. As such it will be necessary for a device to be purchased, or built that will allow the robot access to roofs. There are currently a variety of commercially available lift mechanisms that could accomplish this task. Several of these mechanisms are discussed in this section and consideration is given to each based off of its cost, size, the height it can lift the robot up to, and its ease of use.

For the sake of size, and ease of use only man portable mechanisms were considered. All the lift mechanisms that fit into this category are general purpose lifts in which a platform follows a track on which the robot could be placed and lifted up. These lifts can come in any number of sizes, with a trend that the higher the robot needs to reach the more expensive the lift will cost. All of these lifts could be easily modified such that the robot could be lifted to any height, and then maneuver onto the roof. Although most of these systems have relative ease of use, and can be single or at most double man operated, the size is an issue. The system we need to design must be able to collapse such that it can fit into the back of a standard mini-van, and based on our background research no such device could be found.

2. Methodology

2.1 Project Specifications

Based on the project requirements the following specifications have been derived:

1. Design and create an ascender that allows the robot to exit and enter roof environments. The ascender must be:
    - Operable by a single person
    - Must collapse to a length of 8ft or less to be able to fit in the back of either a truck or van used by inspectors
    - Extend to over 20ft to reach a roof with a pitch of 12/12 or less
2. Design and create a laptop based two way communication/control system that must:
    - Include a live video feed
    - Be able to operate out of line of sight throughout the area of a typical 12/12 residential roof area

3. Create a universal charging point
4. Improve traction

The next section will cover how the robot design meets the project specifications.

2.2 Ascender Base

The challenges presented in the base design were more than one might think. For one, we tried to come up with a design that would allow for easy and ergonomic operation, while incorporating the fewest moving parts possible. There was also the goal of an automated locking mechanism to allow for ease of use. There was also the problem of supporting the ascender poles and the torques that they would produce.

We also had the criteria that according to ANSI A14.2-1982 a ladder cannot be placed on a slope of more than 6 degrees. We decided to adopt this standard as well, meaning that the base would have to be stable on a slope of that degree. Then there is the problem of making it so that the winches to operate the ascender can be placed such that the cables do not incur too many unnecessary direction changes. It is also desirable to place them in such a position that it is not uncomfortable to watch the poles extend while winching.

There is also the consideration that the poles need to be able to be extended at the vertical, and then lowered down onto a roof's edge without incurring roof damages. This means that you must be able to place the poles down onto the edge gently, as the edges of roofs are relatively frail.

2.3 Ascender Pole System

2.3.3 Deformation

The combined weight of the robot and the sled was found to be roughly 40 lbs. This weight was used in the initial calculation to find the deflection in each tube.

Equation 1 gives us the deflection in each individual tube based on inner and outer diameters. Using this equation we can find the total deflection along the length of the system based off of the forces calculated in Equation 1. To allow for the sled to properly ascend the system there could be no more than 6 inches of deflection along the entire system. Based off of this maximum deflection and the given lengths of the system a minimum wall thickness was determined.

Equation 1- Deflection in Poles

$$D = (L^3 * F) / (3 * E * MI) \qquad MI = (\pi * (OD4 - ID4)) / 64$$

D = Deflection L = Length F = Applied Force MI = Moment of Inertia

E = Modulus of Elasticity OD = Outer Diameter ID = Inner Diameter

The applied force can be calculated since the weight of the robot is known to be 40 lbs. In terms of deformation we are only interested in the forces in the x direction, as shown in Figure 3. The force in the x direction can be calculated using Equation 2Equation 2 - Force in the X Direction.

Equation 2 - Force in the X Direction

$$F = W * Cos(\theta)$$

$$F = Applied\ Force \qquad W = weight\ of\ the\ robot$$

2.3.4 Pulley System

In order for the poles to be able to telescope out of each other pulleys needed to be implemented into the system. A pulley was placed at the top of section one, then a cable ran from a winch at ground level, over the pulley at the top of section one, and then was attached to the bottom of section two. When the cable was cranked around the winch this would force the bottom of section two up towards the top of section one. Likewise if a cable was attached to the top of section one, and strung over a pulley at the top of section two and then attached to the bottom of section 3, when section two was lifted out of section so would section three be lifted out of section two. This allows for the entire system to be extended using only one winch, thus being one man operable. This also ensured that each section would rise at the same rate, giving the system added stability.

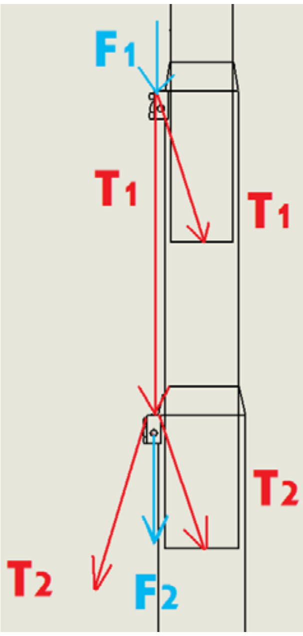


Figure 4 - Pulley Diagram

The forces on each pulley can be calculated using the equations found in Equation 2. Due to the amount of forces necessary a bracket needed to be designed that could hold the pulley in place and still allow it to rotate. The design of the pulley will be just a standard pulley, with bushings around a pin in the center allowing the pulley and pin to rotate inside the bracket.

Equation 3 - Forces on Pulleys During Pole Ascension

$$F_1 = 2*T_1 \qquad F_2 = 2*T_2 \qquad T_2 = F_1$$

Forces and tensions defined in Figure 4

The tensions are caused by the weights of the poles, and thus the forces are caused by the tension in the cable on either side of the pulley. To alleviate some of the force on the lower pulley the cable from the upper pulley can be attached to the top of the lower pulley causing an upward force equal to T_1 as defined in Figure 4. The force exerted on the winch which will vertically ascend the poles must be greater than the downward force caused by the weight of the poles, F_2 as defined in Figure 4.

When the sled is ascended a third pulley is utilized creating a third force and tension. Equation 4 details how this increases the forces on the original two pulleys. The force required to operate the winch that lifts and actuate the sled must be greater than the vertical force F_1 in Figure 5.

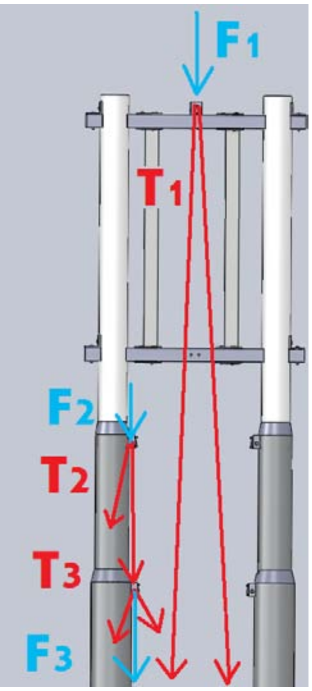


Figure 5 - Sled Pulley Diagram

Equation 4 - Forces on Pulleys due to Sled Ascension & Actuation

$$F_1 = 2*T_1 \qquad F_2 = 2*T_2 \qquad T_2 = F_1 \qquad F_3 = 2*T_3 \qquad T_3 = F_2$$

Forces and tensions defined in Figure 5

2.3.5 Roof Offset

To allow for the sled mechanisms to have clearance over the roof, and to allow the sled to actuate there needs to be at least 6 inches of offset between the poles and the roof. Thus two guides must be made to hold the poles off the roof.

2.3.6 Bearings

Due to the high amount of friction that occurs when one aluminum tube comes into contact with another there is need for a bearing to be placed in between the poles to prevent wear. This bearing if placed at the top of each section could also double as a transition between sections such that the sled rollers would be more able to ascend and descend the poles. The design for this bearing would be a cylindrical hollow tube that fits inside each section but also has a lip that comes over the top. A section of the bearing will need to be cut out so that the pulleys can fit inside the aluminum poles.

2.3.7 Pull Down System

In order to ensure the system collapses when the winches are released a pull down system is necessary. The amount of friction between the sections could at times require more force than gravity alone can provide to collapse the poles, so the cable used to lift the sled mechanism will also double as the pull down system for the poles.

2.4 Ascender Sled

The legacy ascender sled design while optimized for use on the customized ladder based ascender system still left a mechanically simple and effective base line solution for transporting the Roof Robot to and from roof top operating environments. The most significant design concern and consideration for the sled system was the complete and utter revision of the ascender. With the original ascender proof of concept the legacy ascender sled was attached to the ladder by simple Teflon coated 80/20 slider mechanisms. The Teflon slider was an effective solution for attaching to a system with a constant operating width, but not applicable to the variable diameter of the current Ascender pole system.

2.4.1 Ascender Sled Grip Mechanism

A major concern with designing the gripping system for the ascender sled was ensuring that the force used to keep the ascender sled's guides parallel to the telescoping poles but not so much that lifting the ascender sled was unnecessarily difficult. The two potential solutions that were considered initially were the use of springs or a four bar linkage. Using a four bar linkage was only briefly discussed after it was discovered that the first Roof Robot MQP team had determined using a four bar solution was viable as a real world solution.

Spring based solutions were considered. One major concern was that the majority of springs do not provide constant force when extended or compressed; the ascender sled must be able to remain in contact fully at all points along the ascender having variability in the grip force is a far from ideal solution. The second design concern associated with the gripping mechanism was that as the sled needed to be able to be removed from ascender system for transport and storage. With any spring based solution the distance of compression would need to be within the range of median human grip reach and strength.   A third option was formulated, the gripping force would be produced by the tension needed to lift the ascender sled itself.

To translate the tension force lifting the sled body to the individual gripping mechanisms the cable used to pull the ascender sled is attached to two separate smaller cables (1) that divide to the left and right hand side of the pivot mechanism. The left and right hand cables translate a tensional force at the rear pivot axis (2) that will allow pivoting near the peak of the ascender's operating height. The force that is arrives at the individual gripping mechanisms (3) is simply redirected by the eyebolts down to the gripping element.

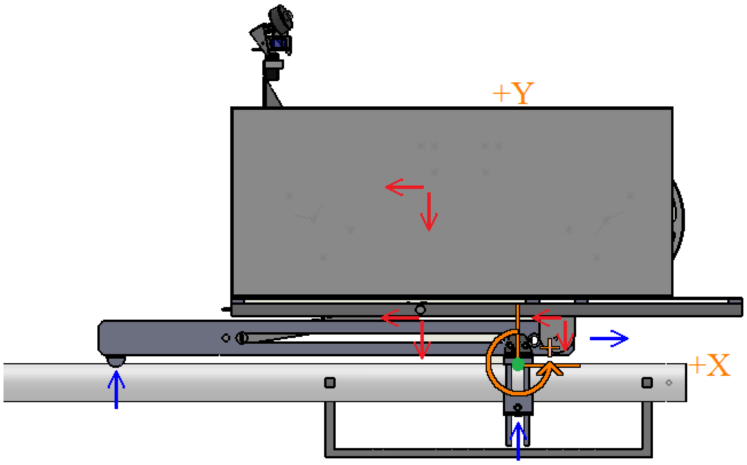


Figure 2-4 Lift Tension to Grip Force Free Body Diagram

$$\sum F_x = -F_{Sledx} - F_{Robotx} - F_{Batteryx} + F_{CableTensionx} = 0$$

Equation 7 Sum of Forces on the Ascender Sled's X axis

$$\sum F_y = -F_{Sledy} - F_{Roboty} - F_{Batteryy} + F_{Gripy} + F_{Guidey} = 0$$

Equation 8 Sum of Forces on the Ascender Sled’s Y axis

$$\sum M_o = -F_{sled} * r_{sled} - F_{robot} * r_{Robot} - F_{battery} * r_{battery} + F_{grip} * r_{grip} + F_{Guide} * r_{guide} = 0$$

Equation 9 Sum of Torques on the Ascender Sled’s origin

After determining how grip force would be produced to maintain contact with the ascender system it was then necessary to determine the potential for deformation that would occur as a result of compressing the grip mechanism. The length of the upper and lower segments of the rotational axles was dictated by the diameters of the telescoping segments as well as the clearance requirements of the pulleys used to expand the telescoping poles. For this aspect of the design 6061 Aluminum was chosen due to the fact that the characteristics of 6061 Aluminum are well known and widely available, its common use throughout industries, and its overall cost competitiveness as a metal. Of greatest concern with the grip subassembly’s deformations was that of the lower axle shown in Figure 2-5 Grip Sub-Assembly, as its ability to properly rotate affected the ease at which the end user would be able to lift the ascender sled.

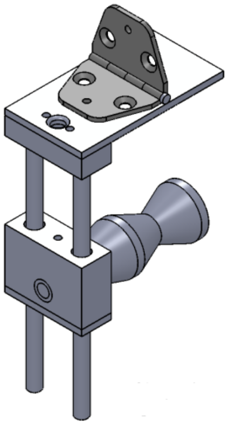


Figure 2-5 Grip Sub-Assembly

$$q(x) = R_1 * \langle x - 0 \rangle^{-1} - F_1 * \langle x - l \rangle^{-1}$$

$$v(x) = R_1 * \langle x - 0 \rangle^{0} - F_1 * \langle x - l \rangle^{0}$$

$$M(x) = R_1 * \langle x - 0 \rangle^1 - F_1 * \langle x - l \rangle^1$$

$$\theta(x) = \frac{1}{E * I}[\frac{R_1}{2} * \langle x - 0 \rangle^2 - \frac{F_1}{2} * \langle x - l \rangle^2]$$

$$y(x) = \frac{1}{E * I}[\frac{R_1}{6} * \langle x - 0 \rangle^3 - \frac{F_1}{6} * \langle x - l \rangle^3]$$

Equation 10: Singularity Functions for a Cantilever Beam

$$F_{grip} \ = Pounds\ Force \quad L = Length\ in\ inches \quad t_{plate} \ = plate\ thickness\ in\ inches$$

$$w_{plate} \ = platewidth\ in\ inches$$

$$E_{Al} = Modulus\ of\ Elasticity \qquad d_{shaft} \ = shaft\ diameter\ in\ inches$$

$$I_{shaft} \ = \frac{\pi}{64} * d_{shaft}^4 \qquad I_{plate} \ = \frac{F_{grip} * L_{plate}^3}{3 * E_{Al} * I_{plate}}$$

$$y_{maxdeflection} \ = \frac{F_{grip} * L_{plate}^3}{3 * E_{Al} * I_{plate}} \quad y_{maxdeflection} \ = \frac{F_{grip} * L_{shaft}^3}{3 * E_{Al} * I_{shaft}}$$

Equation 11 Equations for Calculation of Component Deflections

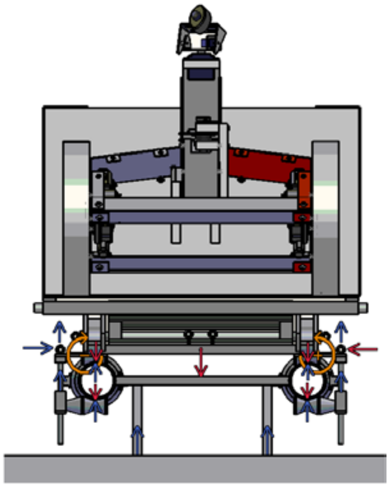


Figure 2-6 Force Diagram of Grip Assembly

2.4.2 Sled Pivoting Mechanism

The primary characteristics of pivoting for the sled were designed and implemented in the original Roof Robot MQP, where the tension force used in pulling up the Ascender Sled was also used to create the pivoting action of the system. In the original design the lift cable was connected to a rear axle of the sled by a single point of contact, this axle was locked into the non-pivoting stage during ascent using a locking mechanism that would release when the Ascender Sled arrived at terminus of the Ascender Assembly. For the updated Ascender Sled design several changes were proposed as to how allow for the pivoting action while still providing a tension force for the gripping mechanism. As both sides of the Ascender Sled would require an even application of tension force to ensure stability it was decided that the cable used to pull the Ascender Sled would connect to a central contact point that also connected to two separate cables that would evenly connect to the left and right sides of the system, as shown in Figure 2-7 Cable Path of the Ascender Sled.

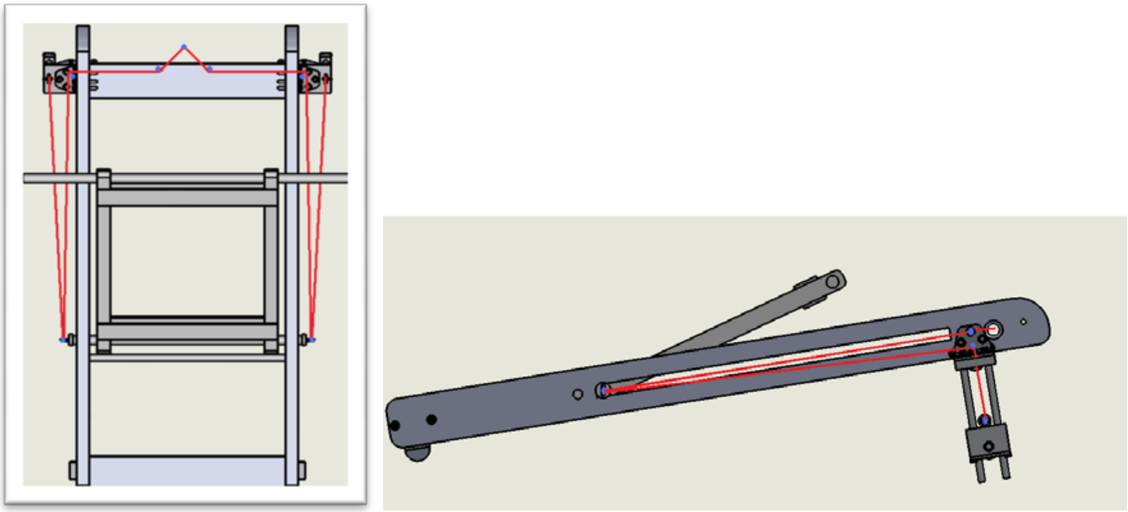


Figure 2-7 Cable Path of the Ascender Sled

The forces required to rotate the carriage during the deployment phase of ascent were recalculated to provide a minimal angle that the Carriage should be in relation to the frame of the Ascender Sled to ensure that any design modifications did not unnecessarily increase effort required to position the robot on the roof. The function and graph of this characteristic

Equation 12 and Figure 2-8, and indicate that the sled should never have a pre-deployment angle less than π/18 radians or 10˚ between the frame and the carriage as the tension force required would be over 250 lbs assuming the initial deployment weight estimate of 50 lbs, a 1.25 safety factor on the current designs.

$$W_{Sled+Robot} = pounds\ force \quad \theta = angle\ between\ Carriage\ and\ Ascender\ Sled\ Frame$$

$$F_{Tension} = \frac{W_{Sled+Robot}}{\sin\theta}$$

Equation 12 Deployment Tension Force Relative to Carriage Angle

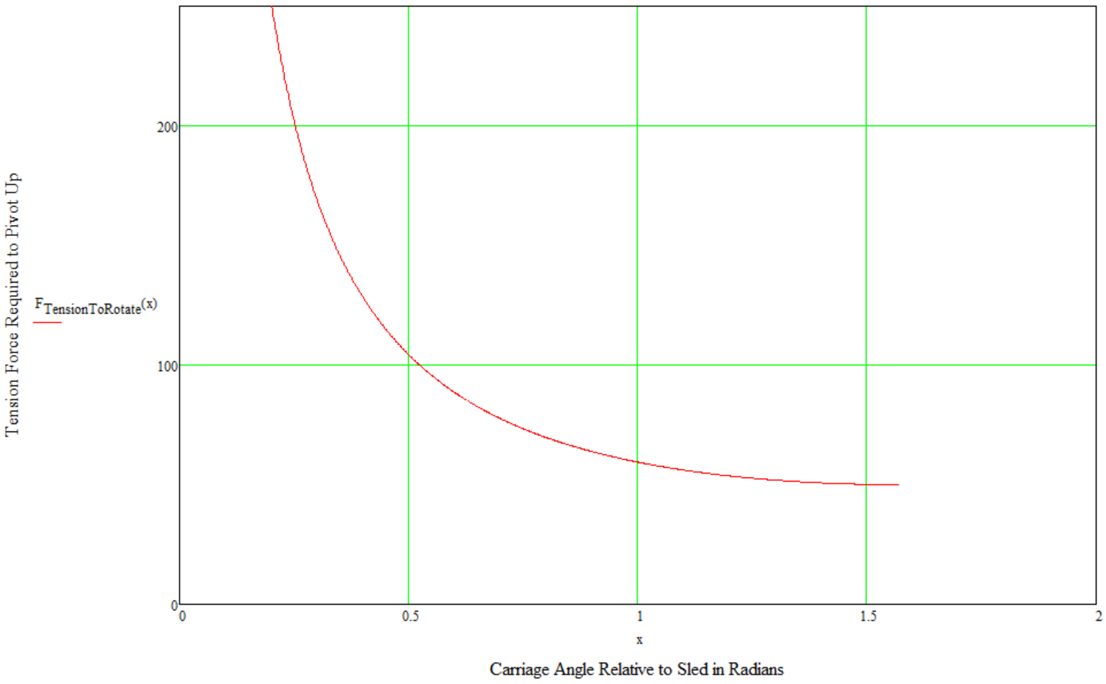


Figure 2-8 Tension Force Calculations

The red lines shown in Figure 2-7 Cable Path of the Ascender Sled represent the path of the cables, with each change in direction, as indicated by the blue dot is represented by a blue dot. These path changes occur 5 times and will require real world testing to determine the force amplification.

For the sled's return phase it is necessary to create a pull down force to ensure that the robot is able to properly return to ground level as shown in Figure 2-9 Free Body Diagram of Sled Pivoting. Please note the arrow indicating a downward force at the rear section of the Ascender Sled is indicative of the application of a pull down force during the return cycle using Dacron cord, this force should not be a major factor during the ascent and deployment phase if used correctly.

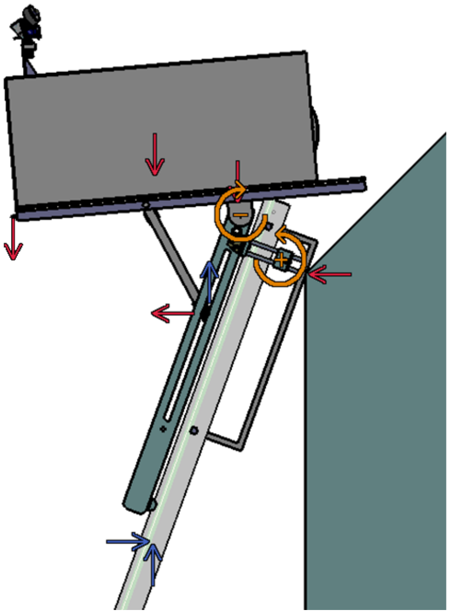


Figure 2-9 Free Body Diagram of Sled Pivoting

2.4.3 Electronic Mounting to the Ascender Sled

The original Ascender Sled served a singular purpose of proving the ability for transporting the Roof Robot from ground level to operational roof height. This met the objective of proving that there was a design that would allow for the transport of the Roof Robot, but did not meet the newer design requirements of this project, ensuring that the Roof Robot was still capable of communicating to the controller and transmitting video feed, when not in line of sight and over the crest of a roof. To ensure that the video camera and micro controller's Wi-Fi signals reach the operator a wireless repeater and its battery based power supply needed to be placed in a location that was as close to the roof top as possible. As the intention of this project was to increase the personal safety of the operator of the Roof Robot it became apparent that however the wireless repeater system was mounted it would need to be safely accessible by the operate during activation and deactivation, and still be moveable to the roof top. With these goals in mind it became apparent that the "belly" of the Ascender Sled Carriage was an advantageous location for mounting these components.

One consideration that needed to be made in the design process was the effect of the batteries weight on the Ascender Sled's force calculations with respect to sled stability, lift force requirements, and the pivoting stage. The general form of the force balance calculations are shown in Equation 7, Equation 8, and Equation 9 and covered in greater detail in section 3.3 The primary result of the calculations made using Force Balance Equations provided us with the regions of the underside would provide the preferred mounting region. As shown in Figure 2-10 Bottom View of Ascender Sled the available surface area for mounting, highlighted in red was more than sufficient for the repeater system proposed for the project while minimizing torques as a consequence of the batteries weight. Many off the shelf wireless repeater systems fit within a 8" *5"*1 ½ " volume, which would be able to fit in the red regions indicated by the white arrows. A 12 Volt battery capable of a 1 Amp draw, the preferred voltage and current characteristics of a wireless repeater, would be able to fit in the volume indicated by the black and white arrow.

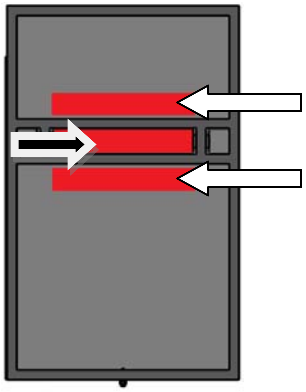


Figure 2-10 Bottom View of Ascender Sled Carriage

2.5 Camera System

The camera must meet the following specifications:

- Must be able to incorporate a live video feed
- Must be able to record roof inspections

In order to meet these specifications the robot now uses a Linksys Wireless-G Internet Home Monitoring Camera. There are several deciding factors in purchasing a camera including the ability to work out of line of sight and over the crest of a roof as well as being small enough to not significantly affect the center of gravity of the robot. This requires a Wi-Fi camera as opposed to another type of wireless camera. The reason for this is the ability to operate within range of other wireless devices as well as the ability for the signal to be boosted by a router. Specifically this camera must work on IEE 802.11G to work with the available router. Ideally we would also like for the camera to be as inexpensive as possible, for obvious reasons. We researched many varieties of cameras; however eventually we decided on the current camera which has the following specifications:

Width	3.5 in
Depth	1.5 in
Height	4.7 in
Weight	4.6 oz
Digital Video Format	MPEG-4, MJPEG, ASF
Cost	$85.00
Video Resolution	640 x 480, 320 x 240, 160 x 120
Recordable	Yes: remote record ability
Wireless	IEEE 802.11G
Audio Support	Yes : built-in microphone
Operating Voltage	5V

Table 1: Camera Specifications

The camera requires a router to work. Any device that can log on to a wireless network can use this device. This allows for the robot to be viewed through a laptop as well as through mobile devices such as an IPod touch. In order to establish a continuous wireless link between the camera and the control laptop a router is located within the sled. This allows for indirect ground communication bypassing the house, a major source of interference, entirely. Now instead of having to transmit through the house the camera can transmit to the edge of the roof, where the sled and router are located, and from there the signal is transmitted to ground level. To access the camera all one has to do is log on to the RR3 network produced by our router. From there the user logs into a predetermined IP address, in this case 192.168.1.4. From this screen, Figure 2-11, you can view video in real time as well as record video for future use. The camera system is completely independent from the microcontroller.

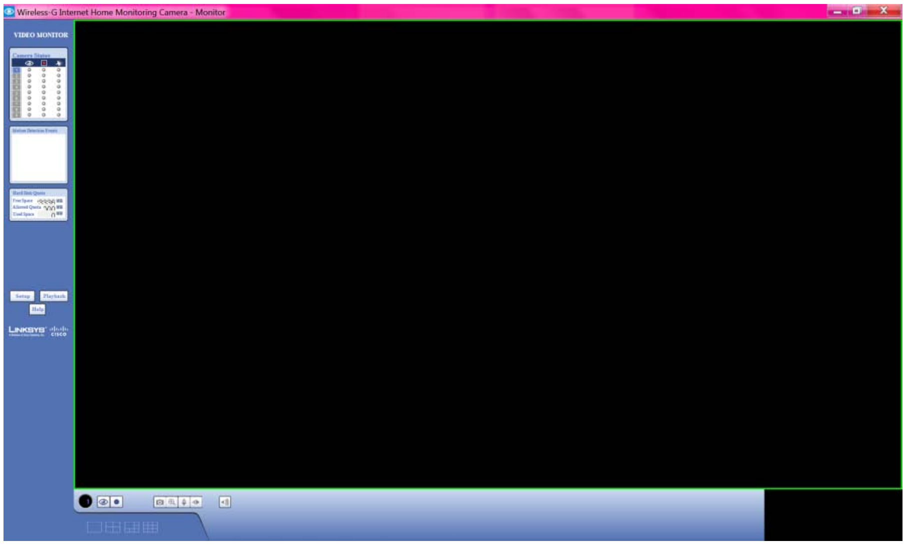


Figure 2-11: Wireless Camera Software

2.6 Two-Way Communication

The robot communicates with the ground through two separate wireless systems. The first system is for the wireless camera. The camera transmits a signal to a Netgear DG834Gv5 wireless router. From here the signal is transmitted to a laptop or other wireless device on the ground, where an operator can view the image as well as record any or all of the video feed. The VEX microcontroller operates on a separate system altogether. The remote control and microcontroller are used to drive the robot. They communicate with each other via two USB wireless-lan devices. One device is located on the robot while the other is located on the sled. Each device is hardwired to the microcontroller and the remote control respectively. The following, Figure 2-12, not to scale drawing shows these two systems in action. Red arrows represent the camera system while blue arrows represent the VEX system. A dashed line is a wireless signal and a solid line is a wired signal. The operator is able to stand on the ground and inspect the roof while only looking at a computer monitor.

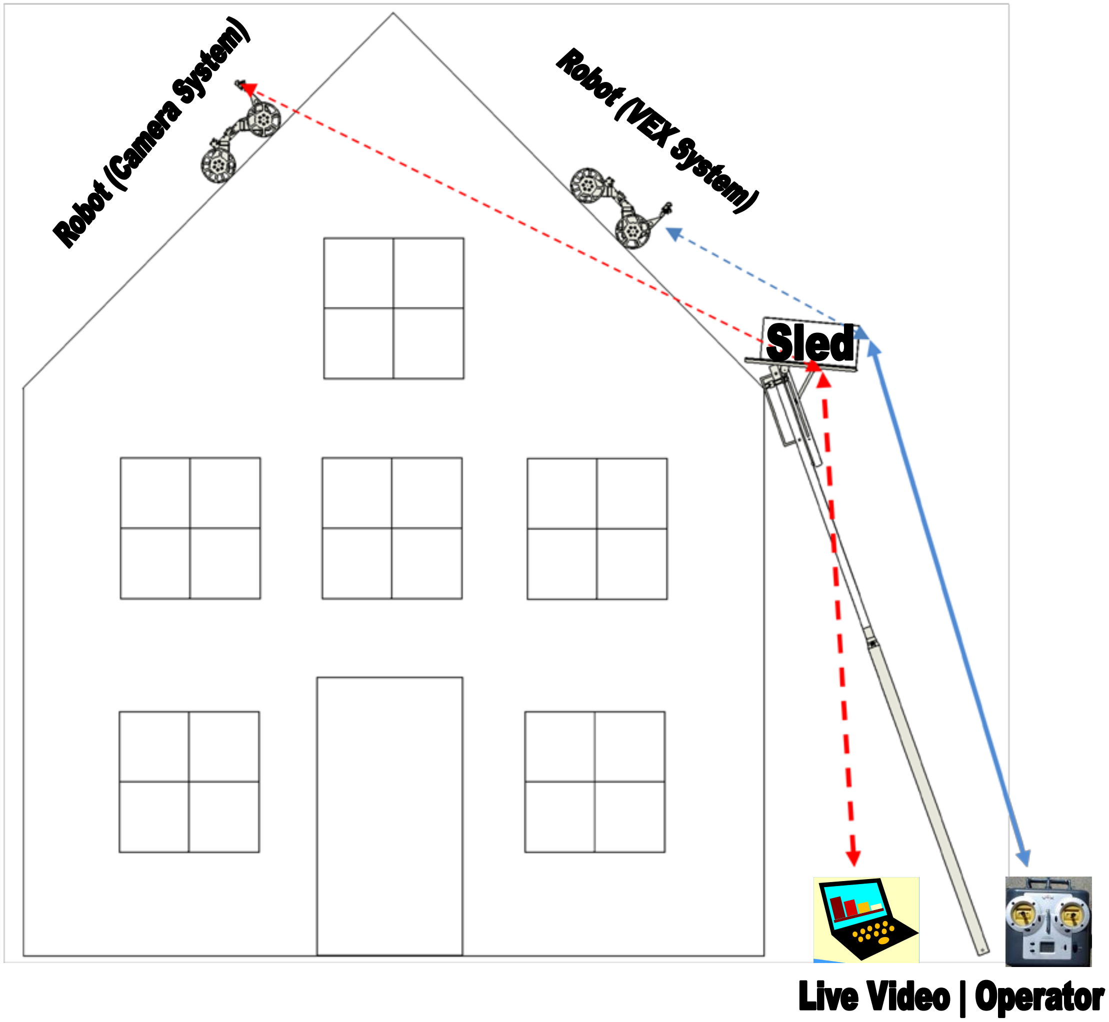


Figure 2-12: Wireless Path

2.6.1 Microcontroller Selection

The robot must meet the following specifications:

- Be able to operate out of line of sight
- Be able to have two-way communication

As mentioned earlier the legacy version of the robot used the VEX microcontroller. While researching various microcontrollers it was discovered VEX had recently released wireless upgrade that would allow for the two-way communication required. This solution was deemed the quickest way to upgrade the system. This would allow for the most time for increasing the range, the primary goal of the upgrade. Initially the plan was to upgrade to VEXnet, a plug and play upgrade that allows for wireless two-way communication over IEEE 802.11g using the VEX microcontroller. When the remote control is plugged into a computer this information can be displayed onscreen. The user can also control individual robotic components from this display. This is very useful for troubleshooting.

While adding the wireless functionality to the legacy code it was discovered that the existing microcontroller ran out of room. With 32Kb of available flash memory for programming, the microcontroller was simply not large enough. While we were trying to reduce the size of the code Vex produced a beta version of Vex 2.0. This incorporated the wireless upgrade as well as adding memory to the microcontroller allowing for over ten times the program space at 384Kb. We were able to get a hold of one of the beta versions of this microcontroller for the project. We now have a microcontroller that can handle two-way communication as well as handle the size of the code we wished to use. As you can see from Table 2 below, the two microcontrollers are physically nearly identical. The similarities in weight, size, and battery input allowed for a seamless transition. This along with the fact that it allowed us to meet all of our specifications made the decision clear that VEX 2.0 was the optimal microcontroller for the project.

The microcontroller and the remote control communicate via two USB dongles. Initially they were having the same issue as the camera. However unlike the camera we were unable to route the signal through the router. This required two USB extenders. One moved the microcontroller dongle out of the coils of wires inside the robot and up to the camera mount. The second dongle is attached to the sled via a 10m extension. The dongle is then raised along with the sled, router, and robot, to the roof. This allowed for the robot to be maneuvered out of line of sight.

VEX 1.0	VEX 2.0
Battery In	
* Voltage:7.2 volts nominal, 5 to 12 volts min/max.	* Voltage:7.2 volts nominal, 5 to 12 volts min/max.
* Type:Six AA batteries or 7.2V Robot Battery	* Type:Six AA batteries or 7.2V Robot Battery
* Current:62 mA for Controller & Receiver plus Motors & Servos	* Current:62 mA for Controller & Receiver plus Motors, Servos & VEXnet
Microcontroller	
Microchip PICmicro PIC18F8520	STMicroelectronics ARM Cortex-M3
* Speed:10 MIPS (Million Instructions Per Second)	* Speed:90 MIPS (Million Instructions Per Second)
* RAM:1800 bytes + 1024 bytes EE2	* RAM:64KB
* Flash:32K program space	* Flash:384KB program space
Size	
4.5in W x 3.9in L x 1.1in H	4.5in W x 3.9in L x 1in H
Weight	
0.302 lbs (137 grams)	0.302 lbs (137 grams)

Table 2: VEX 1.0 vs. VEX 2.0

2.6.2 Code

The code is fully documented in Appendix B- Robot Code. This section will contain an overview of the general functionality of the code. Below you will find a flow chart detailing the general functionality of the code

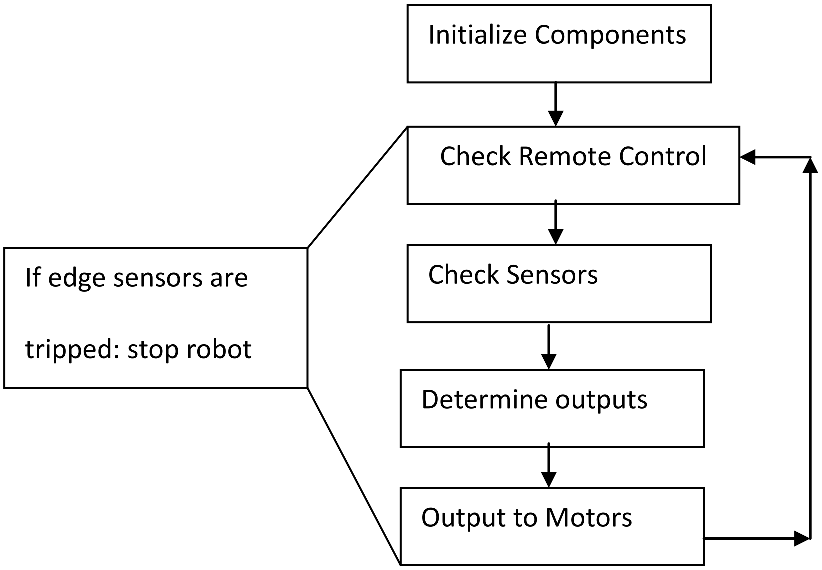


Figure 2-13: Code Flow Chart

The basic code structure is an infinite loop. First it checks the position of the joystick. It then determines the desired output speed based on the current speed of each motor. The motors are then activated. This code continues to loop checking sensors to ensure correct operation.

The legacy code provided the basic functionality of the however with the introduction of the new microcontroller as well as the additions to robot new code was required. The changes to the legacy code to incorporate the new microcontroller did not affect the overall flow of the code. The new microcontroller could not handle the functions of math.h. These include advanced math functions such as those required to take the tangent of an angle. The unique turning characteristics of this robot shown in Figure 2-14 require the use of tangent.

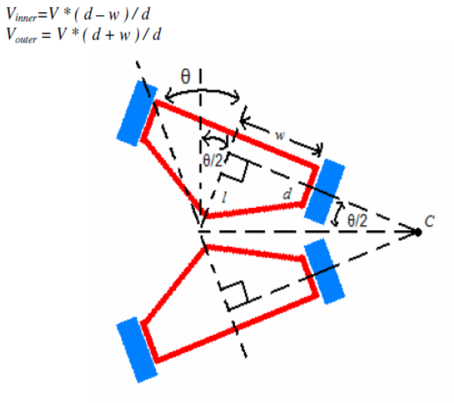


Figure 2-14: Turning Math

An array with the values of the tangent of numbers from 0 to 50 was added to the code to allow this operation to be performed. The traction control also required some minor adjustments for the new wheel with. This included adjusting neutral values. The camera mount was also replaced requiring adjustments to the camera code. This included setting new values for neutral and setting new maximum values to ensure centering of the camera on turns.

2.7 Single Charging Point

The robot requires three 12V batteries to run in parallel to operate. This setup was determined in a previous MQP which ensured at least forty-five minutes of operating time. This result was verified with the new components. Rechargeable batteries require independent charging. This required a way to run the batteries in parallel while charging them individually. The batteries have two connectors. One is used for charging and one is used for running. The circuit diagrams below, Figure 2-15 and Figure 2-16, show how this was done. Three single pull single throw switches operated by a single lever allow the run mode connectors to be connected during run mode and disconnected during charge mode.

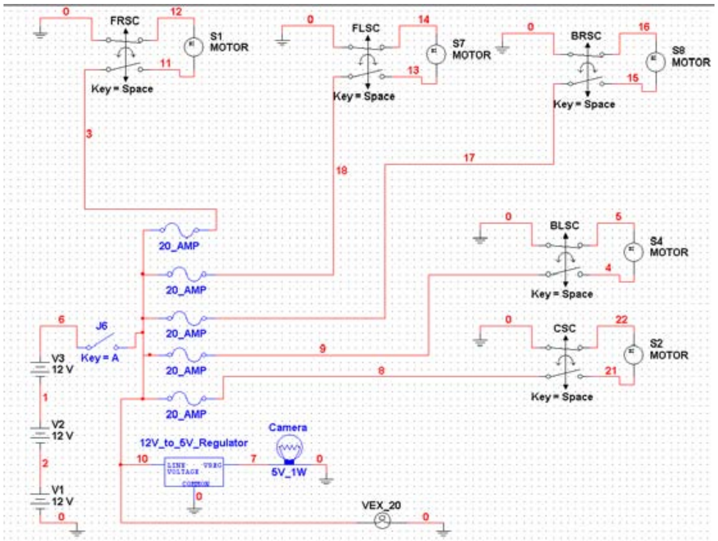


Figure 2-15: Run Mode Schematic

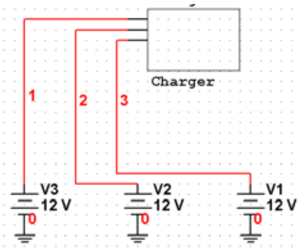


Figure 2-16: Charge Mode Schematic

The new setup also moved all the charging connections to one point on the robot. This ensured that the batteries would only need to be removed when they died. This can be seen in Figure 2-17 below.

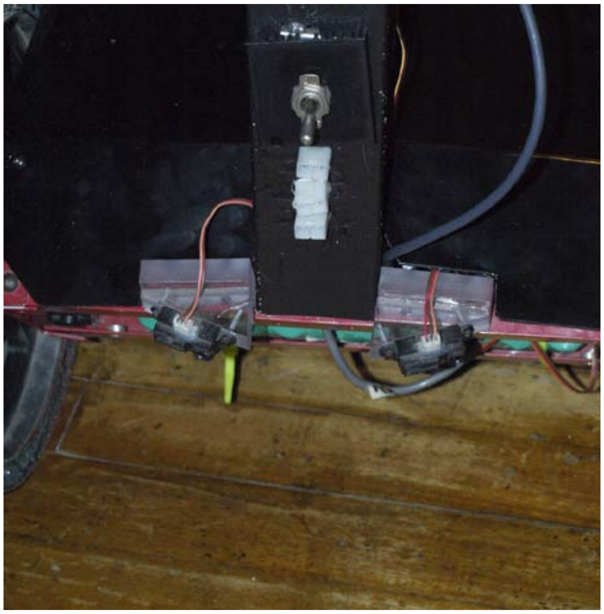


Figure 2-17: Universal Charging Port and Charge/Run Switch

The final schematic shows the position of the individual components followed by a list of where they are connected to the VEX 2.0 microcontroller.

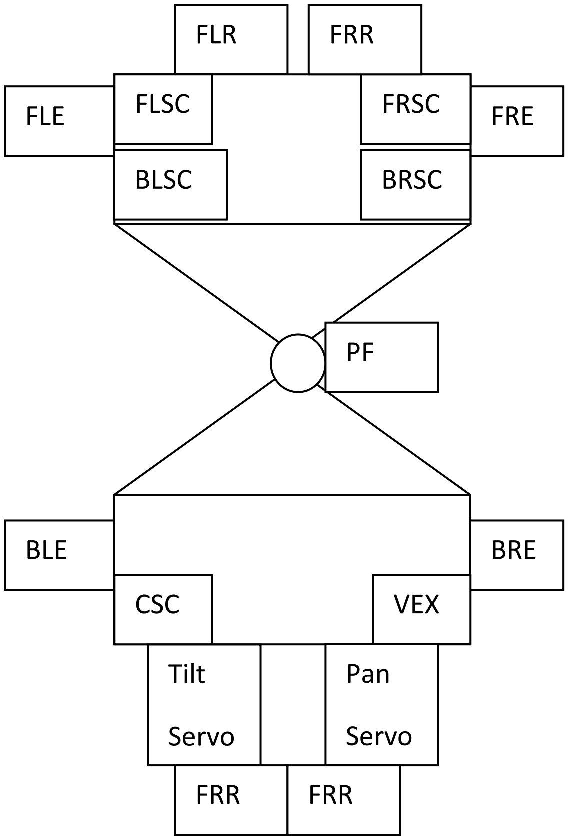


Figure 2-18: Logic

Port Type	Port Number	Function
Analog Input	1	(PF) Potentiometer feedback
Analog Input	2	(FLR) Front Left Rangefinder
Analog Input	3	(FRR) Front Right Rangefinder
Analog Input	4	(BLR) Back Left Rangefinder
Analog Input	5	(BRR) Back Right Rangefinder
Digital Input	6	(FLE) Front Left Encoder
Digital Input	7	(FRE) Front Right Encoder
Digital Input	8	(BLE) Back Left Encoder
Digital Input	9	(BRE) Back Right Encoder
PWM	2	(FLSC) Front Left Speed Controller
PWM	3	(FRSC) Front Right Speed Controller
PWM	4	(BLSC) Back Left Speed Controller
PWM	5	Empty
PWM	6	(BRSC) Back Right Speed Controller
PWM	7	(CSC) Center Speed Controller
PWM	8	Tilt
PWM	9	Pan

Table 3: VEX Logic

2.7 Improved Traction

The traction is a problem that has plagued the roof robot since its inception. The original roof robot team specified that they should be able to climb a roof at a 12-12 pitch. This was never achieved. That was a goal that was never fully realized. While it was not one of our main priorities there was a need to improve the traction from the state it was in, because at the point it was at the wheels were not stable to be stagnant on a 45 degree pitch and the robot certainly couldn't move from a stop on that steep of an angle.

In order to improve traction we decided that we would need to test a variety of new materials for their performance, and then directly compare all of the options. We tested 4 new materials and compared them on a tilt table using a variety of different substrate materials in order to ensure that we were getting the best results for our materials.

Figure 2-19

3. Results

This section describes how the robot met the specifications laid out in the methodology section of the paper. It will also describe the general upgrades to the robot. These include things like rewiring, recoding, and rebuilding different sections of the robot in order to increase usability.

3.1 Ascender Base

In its final state, **Error! Reference source not found. Error! Reference source not found.**, the base is a 3 foot collapsible structure. The height was chosen as an approximate ergonomic height. The base is assembled out of 6063 aluminum square tubing. We used a section of 3/4" x 3/4" with a 1/8" wall thickness. The vast majority of the base is assembled out of 6063 aluminum alloy with 1" square tubing with a 1/16" wall thickness. It is assembled with welds at the joints, as well as bolts attaching the winches.

Half of the base structure is solid while the other matching triangular sections are hinged to allow them to collapse to fit into the back of a van. When the sections swing out they add a large amount of stability to the system. The base is also staked down to the ground using tent stakes to allow for a degree of inaccuracy in the levelness of the surface on which the base is placed.

The base has capability to allow the poles to lean either toward or away from the house, which is necessary for ascender assembly. The other way around this problem would have been to make the base able to pivot at any point. If we had gone this route the base would have been set up parallel to the house and then turned toward it. As it stands, the base does not pivot and is always perpendicular to the house. The topmost sections are removable as shown below in Figure 3-1 to allow for 180 degrees of rotation about the base. The winches are contained on the removable section further from the house. The other removable section serves to catch the poles and hold them in the vertical position while they are extended to their full length.

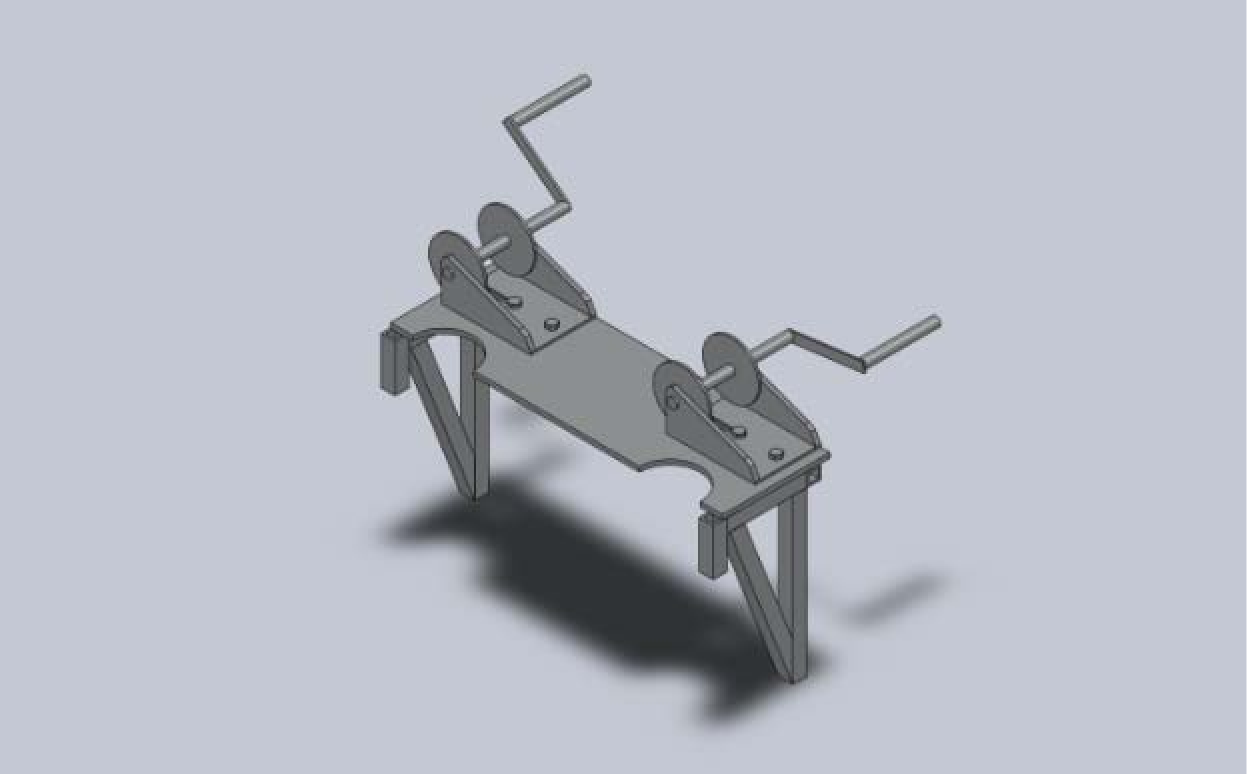


Figure 3-1: Topmost Removable Section

Figure 3-2: Hand Winch

The winches used were ordered from NorthernTools.com and have a 600lb capacity. It has a gear ratio of 3.1 to 1. It also has an automatic braking system so that it is able to stop the poles or the sled from falling back down after they are winched up. The winch is shown to the right in Figure 3-2.

As far as connecting the poles to the base, the necessary pin size in SAE 1070 steel was less than 5/64. For ease of ordering they were designed to use a pin with ¼” diameter. These pins are rated in a double shear situation to withstand 7050 lbs. Since our required strength was in the range of 500-600 lbs, this is clearly more than sufficient.

3.2 Ascender Pole System

3.2.1 Material selection for poles

Based off of weight and cost requirements 6061 alloy aluminum tubes were chosen for the telescoping poles. Spectra cable was chosen to be used to telescope these sections over the pulleys because a .060 diameter cable has a breaking strength of 750 lbs, and because of its resistance to stretching. The cover for section needed to be light and nonconductive but did not need to support any forces so PVC pipe was chosen. Finally lexan poles will be used as the cross bars due to their nonconductive nature.

3.2.2 Pole Geometry

Using the data gathered from the solidworks deformation calculations it was found that the smallest possible inner diameter allowed while keeping less than 6” of deformation would be 1.875”, with a wall thickness of .065”. The rest of the sections specifications were then calculated assuming each section has a wall thickness of at least .065” and allowing for a 1/8” clearance between sections. To reach the height needed four sections were required, each of them 7’6” long with a 1’ overlap between them. Table 4gives the details of each section. These specifications were chosen based off of the minimum requirements for deformation, and based off of availability.

Table 4

	OD	ID	Piece	wall
sec 1	3	2.75	3.0000 x 0.125	0.125
sec 2	2.5	2.37	2.5000 x 0.065	0.065
sec 3	2	1.875	2.0000 x 0.065	0.0625
sec 4	1.875	1.709	1.8750 x 0.083	0.083

Section 1 has a thicker wall thickness because when collapsed it covers the other sections, but has no protection itself from any damage caused through transportation. Due to the need for a nonconductive cover over section 4 a different type of transition was required between section 3 and 4 as compared to the other transitions. It was decided that the

Figure 3-3 Section 3 – 4 Transition

best way to transition between section 3 and 4 was not to have section 4 telescope out of section 3, but rather the outer diameter would be slightly less than the inner diameter of section 3 such that section 4 could slide in and out of section 3. The cover which will be placed over section four will have the same outer diameter as section 3 and will prevent section 4 from sliding too far into section 3. This is illustrated in Figure 3-3.

3.2.3 Pulley System

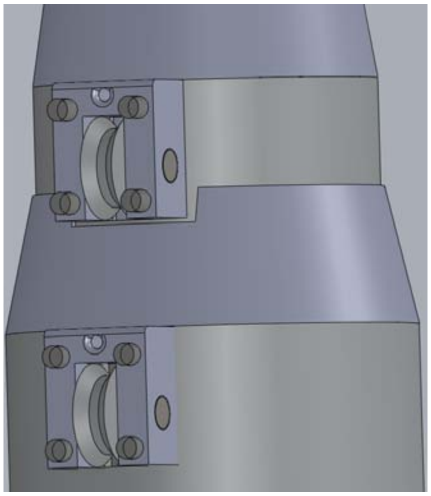


Figure 3-4 - Pulleys

The pulleys will be standard aluminum pulleys with a .287' radius. The pin that fits into them is a stainless steel pin held in place with bushings. This will allow the pin to rotate inside of the brackets. The brackets themselves will be aluminum as well and will have multiple purposes. They will not only hold the pulley in place, but also keep the cable from being easily dislodged from the pulley. The bracket at the top of section 2 will also act to keep section two from collapsing too far into section 1, as shown in Figure 3-4. Four tapped holes will be drilled through the top of section 1 and section 2 and through the brackets placed at the top of these sections. Pan Head screws will be screwed through these holes holding the bracket in place. The total force on the top pulley is 200 lbs, which comes from actuating the sled mechanism. From **Error! Reference source not found.** this means that the total force on the bottom pulley is 800 lbs. This is well below the breaking strength of both the aluminum pulley, and the steel pin.

3.2.4 Bearings

The material chosen for the bearing was Teflon tubing due to its very low coefficient of friction. The top section of the bearing allows for a transition between the telescoping pole sections, and the bottom section prevents the poles from coming into contact with each other. The top cut out portion of the bearing shown in Figure 3-5 is where the second pulley bracket will fit when the system is collapsed. The bottom cut out section is where the pulley on section 1 will sit.

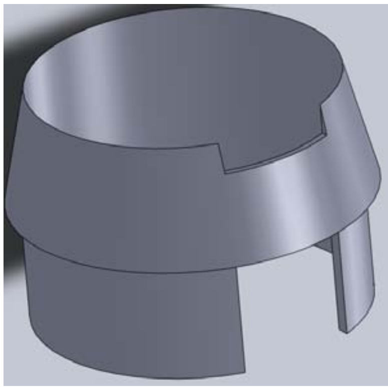


Figure 3-5 - Bearing

3.3 Ascender Sled

3.3.1 Design Requirements

The Ascender Sled's design allows for a dynamic grip action while ascending and descending the Ascender Pole system. This characteristic was achieved by translating the tension force acting on the Ascender Sled body, to pulling on the pivoting mechanism for the Ascender Sled Carriage and the Grip mechanism. The new Ascender sled design maintains the original design characteristic of only requiring a single directional tension force to achieve linear motion as well as pivoting at the interaction point of the Ascender Pole System and the target roof.

3.3.2 Grip Mechanism

With the new gripping mechanism shown in Figure 2-5 Grip Sub-Assembly the Ascender Sled has been modeled to allow for proper clearance of the pulleys required by the Ascender Pole System and can be readily adapted as a design larger Ascender designs. The 6061 Al Alloy specified in the schematics for the Lower Grip and the Grip Guides are calculated to experience slightly more than 1/16" of deflection while in use. The nylon skids and wheel mounted to the Grip Mechanism are estimated to have a coefficient of friction ranging between 0.2 and 0.4 when interfacing with the upper non-conductive sleeve and the Aluminum surface of the lower ascender poles respectively. The design for the rear skid is such that it can be replaced as needed.

3.3.3 Ascender Sled Pivoting

The force required for pivoting the Ascender Sled Carriage during the deployment phase was reduced, by shortening the sliding region that the Sled Pivot was able to operate in. This force reduction was a consequence of the minimal relative angle of the Carriage Assembly and the Ascender Sled Rail Guide being limited to 15˚ as opposed to the original 9.5˚. The angle change provided a tensional force reduction of an estimated 90+ pounds, 297 pounds force to 200 (calculated, see attached MathCad File Pivot Force Calculation), during the pivoting stage of the Ascender Sled.

3.3.4 Electronics Mounted to the Ascender Sled

The Ascender Sled design currently provides a defined mounting volume for the same class of Nickel Cadmium batteries that are used in the Roof Robot shown provided in the CAD renderings of the Ascender Sled. A wireless repeater may be comfortably mounted in the highlighted regions shown in

3.4 Robot

The robot required two main upgrades. A new camera was required, allowing the camera to operate out of line of sight, as well as record video for future use. The microcontroller to remote control link also needed to be changed so that two-way out of sight communication was possible. Both of these requirements are satisfied under the new system.

3.4.1 Out of Sight Two-Way Communication

The robot was required to complete the following specification:

> Design and create a laptop based two way communication system that must:
>
> - Include a live video feed
> - Be able to operate out of line of sight

The Linksys Wireless-G Internet Home Monitoring Camera combined with a Netgear router allows for out of sight communication. As mentioned earlier this camera can be accessed by any Wi-Fi capable device. Using the online software the user can both watch live video as well as record any video they wish. This camera did cause some issues because of its weight; however we were able to offset this by designing a new camera mount as you can see in the figure below, **Error! Reference source not found.**. The Wi-Fi camera also eliminated the issue of interference that previous cameras encountered. This is because unlike the previous camera the Linksys camera works on a different frequency then common security cameras.

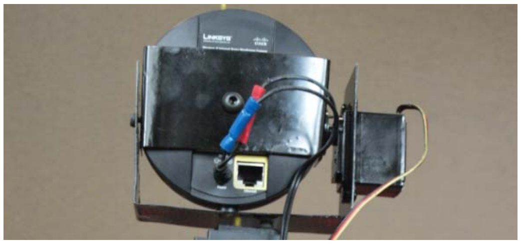


Figure 3-6: New Camera Mount

The range of this device is still not absolutely certain in all situations as wireless range depends on many factors including objects between the receiver and transmitter. In order to ensure that the robot would work in all situations we worked to maximize the range as well as performing multiple tests to ensure range. The average test results are displayed in Table 5 below.

Roof Robot Range Testing at Two Stories						
Dongle to Dongle (Feet)	0	10	20	30	40	50
Camera Signal Still (s lag)	0-1	0-1	0-1	2-5	5-7	5-15
Camera Signal Moving (s lag)	0-2	0-4	2-5	4-7 Lag	5-10	5-15
Vex Signal	Full Strength	Full Strength	Full Strength	Full Strength	Intermittent Signal	No Signal

Table 5: Range Testing

This test was performed in a typical three story Worcester building. The router, VEX wireless device, and the robot was placed at the two story level while the computer and remote control were placed on the ground level. The robot was then driven straight away from the router and placed at ten foot intervals. The camera signal and VEX Signal were then measured. The camera was measured in amount of lag while the VEX was measured ability to communicate. These results prove the robot has met the specifications laid out for it.

3.5 Single Charging Point

The robot was required to meet the following specification:

Create a universal charging point.

The single charging point requirement was met. This included installing a new switch, upgrading the charger, and rewiring the robot. The finished result can be seen in, Figure 3-7, below. These upgrades ensured that the batteries would not be accidently overcharged and explode, a big safety concern from previous iterations. The rewiring also eliminated several questionable wiring jobs leftover from previous groups.

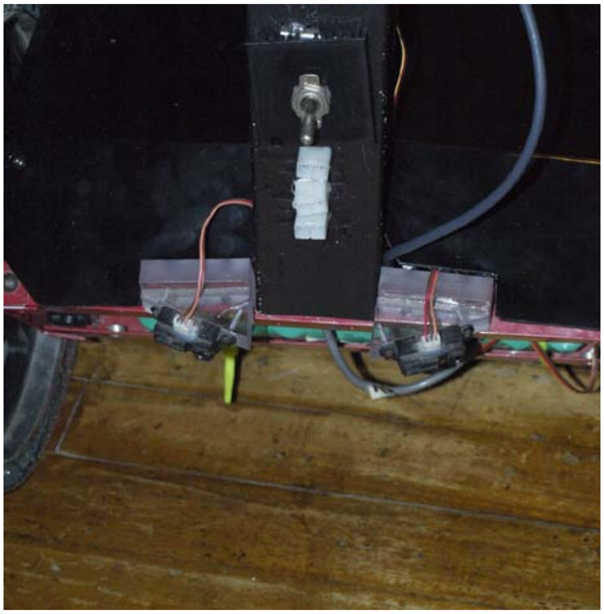


Figure 3-7: Charging Port

3.6 Improved Traction

We tested 4 different products as possible choices for contact materials. Below is a table illustrating the improvements in traction associated with the

After the testing that we were able to do with our scale model, we determined that the brown shelving material with two layers of scrub pad substrate material. This provides a better grip surface for 2 reasons. For one, the brown shelving material has a higher coefficient of friction on roofing material than was existing on the robot. The added substrate also allows for higher deformation of the wheel, which means that the contact surface is expanded.

3.6 Setup Guide

Up to this point there has been lots of information about the individual components however the overall operation may still not be clear. This section should explain from start to finish how the robot is managed and controlled.

1. Place the fully folded base approximately X ft. from the edge of the house.
2. Swing out both of the hinged sections of the base so that they are parallel to their counterparts on the static portion of othe base.
3. Place the two telescoping poles into the base at the bottom and pin them into place.
4. Attach the top nonconductive section of the poles.
5. Insert the first removable top section (the one without winches) into the top of the base on the side closest to the house.
6. Raise the poles to vertical using the other removable section to push them upward.
7. Insert the second removable section (the one with the winches attached) into the top of the base, locking the poles to vertical.
8. Run the required cables through the pulleys and attach them to the winches.
9. Once all cables are run, winch the cable that attaches to the poles in order to extend them to their full height.
10. At full height remove the section that doesn't contain the winches, and lean the poles toward the house slowly.
11. Once the poles make contact with the eve of the roof ensure that contact is made with the nonconductive spacer.
12. With the poles leaning against the house, attach the clamp at a location such that the sled does will not make contact with the base.
13. Attach the sled to the bottom of the poles letting it slide back down onto the clamp.
14. Once the slet is securely fastened to the ascender, power on the robot via the switch on the front.
15. Power on the remote control via switches on the front and back.
16. Power up the router by plugging in the attached battery.

17. Place the robot in the sled.
18. Power on the laptop.
19. Plug the controller into the laptop
20. Open the easyC terminal window.
    a. View this to see robot feedback
21. Connect to RR3
22. Open up a browser and type 192.168.1.4 into the address bar.
    a. View this to see the camera
23. Now winch the sled up toward the top of the roof.
24. With the robot at the top of the roof, begin roof inspection referring to the user guide for operation details.

4. Recommendations

4.1 New User Interface

With the group's limited coding experience the group decided not to tackle this task. However it is desperately needed. Currently the user must sit near the ascender, holding the remote control, while staring at the computer. Ideally the remote control and the video feed should be done on one device. With the ability to control the device over any Wi-Fi capable device this user interface could either be done on a traditional laptop, or for a more lightweight situation, this could be done on a mobile device such as an IPad.

4.2 Center Joint

The center joint of the robot needs a redesign. At the very least the center joint could stand to be remanufactured. The main issue is that all of the weight is being supported by a single 10-24 bolt which is tapped through aluminum. There is also an issue with the screws that hold the chain which attaches to the potentiometer. This would be noncritical since it isn’t load bearing, however the robot needs an accurate potentiometer reading in order to adjust the speed of the wheels properly. The issue here is that the holes are tapped into a lexan cylinder. Lexan does not have optimal hardness when compared with a steel screw and as such the holes have migrated over time. This makes the threads that have been tapped inaccurate and there is no more room in the part to tap new holes. We recommend re-machining the part out of a different material, whether aluminum or something harder.

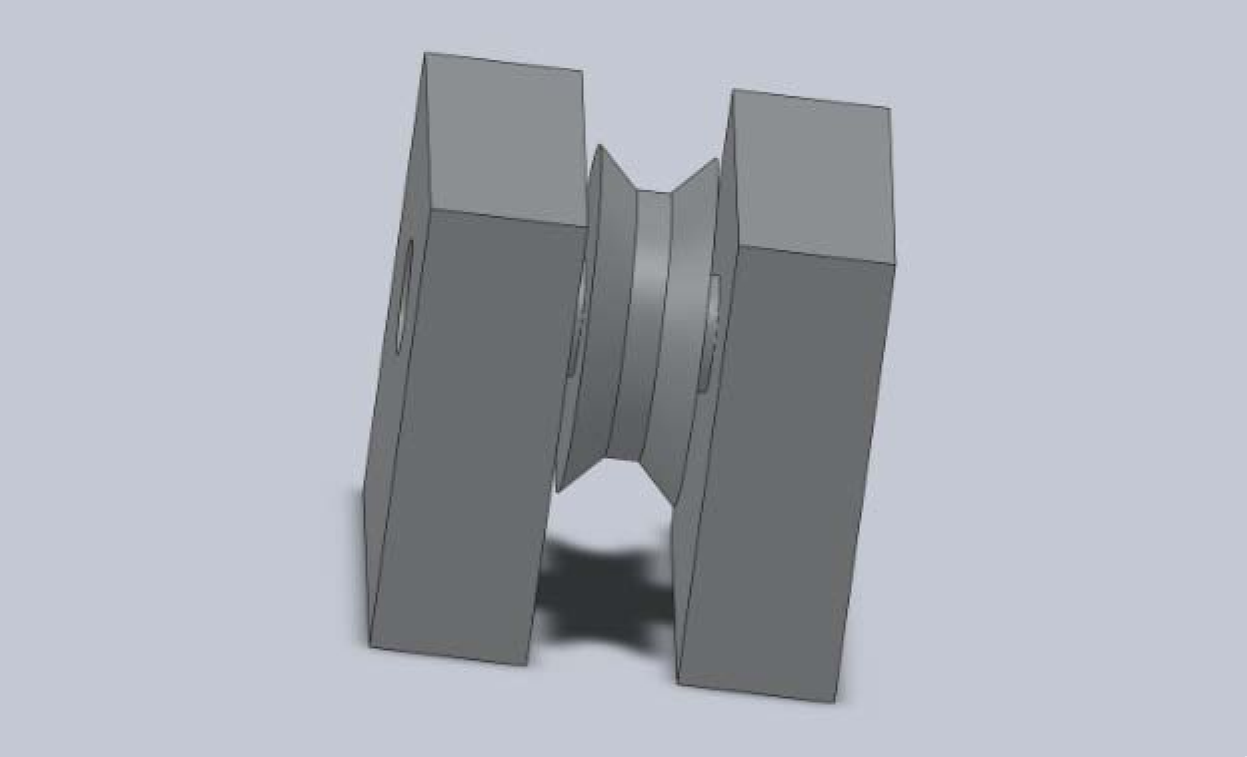
4.3 Traction Materials

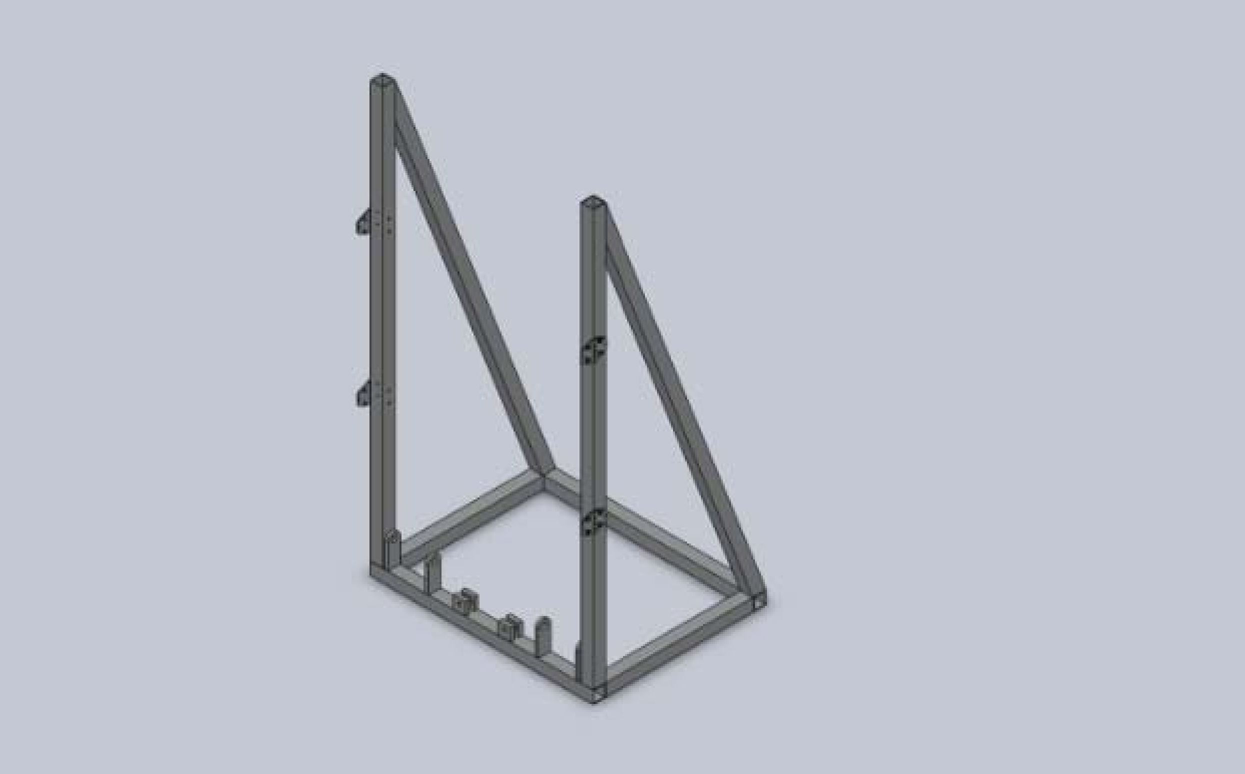
The material that we arrived at for this iteration of the wheels was the brown foam rubber material with two layers of scrub pad substrate. While this was an improvement over the material that had been on the robot, there is still much work to be done.

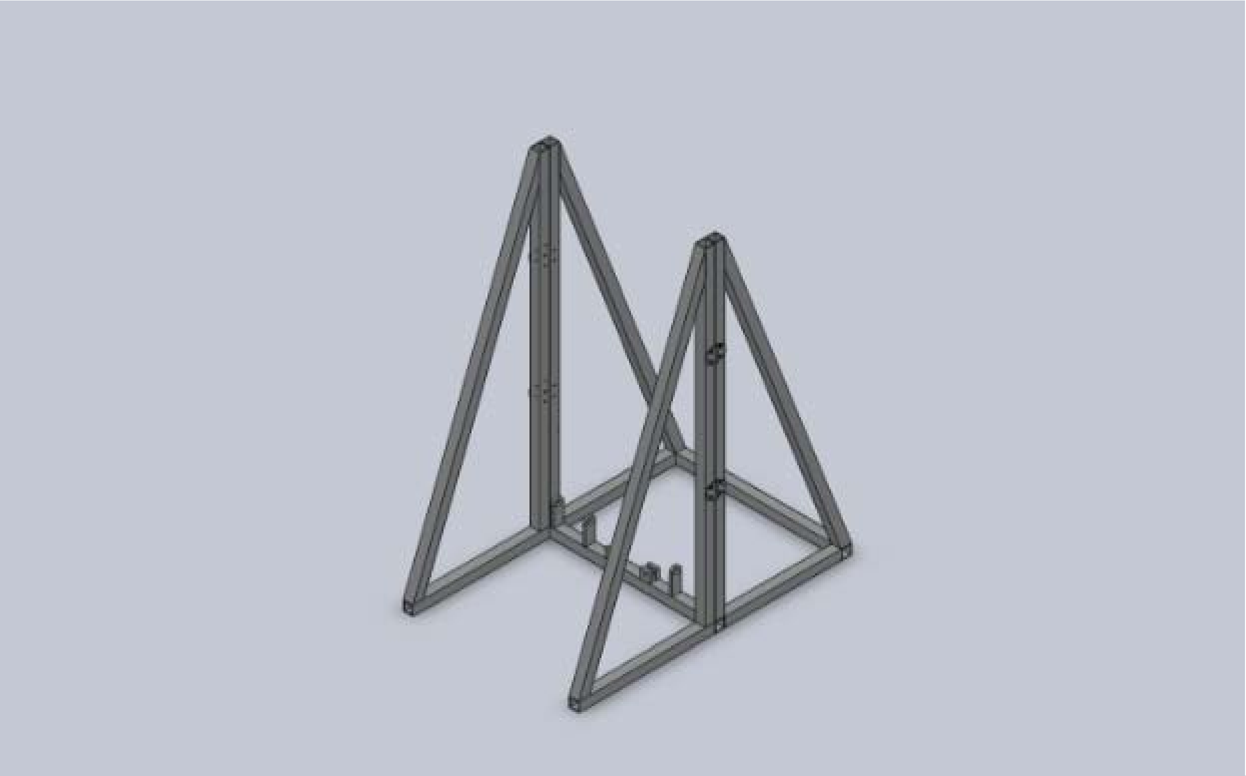
5. Conclusions

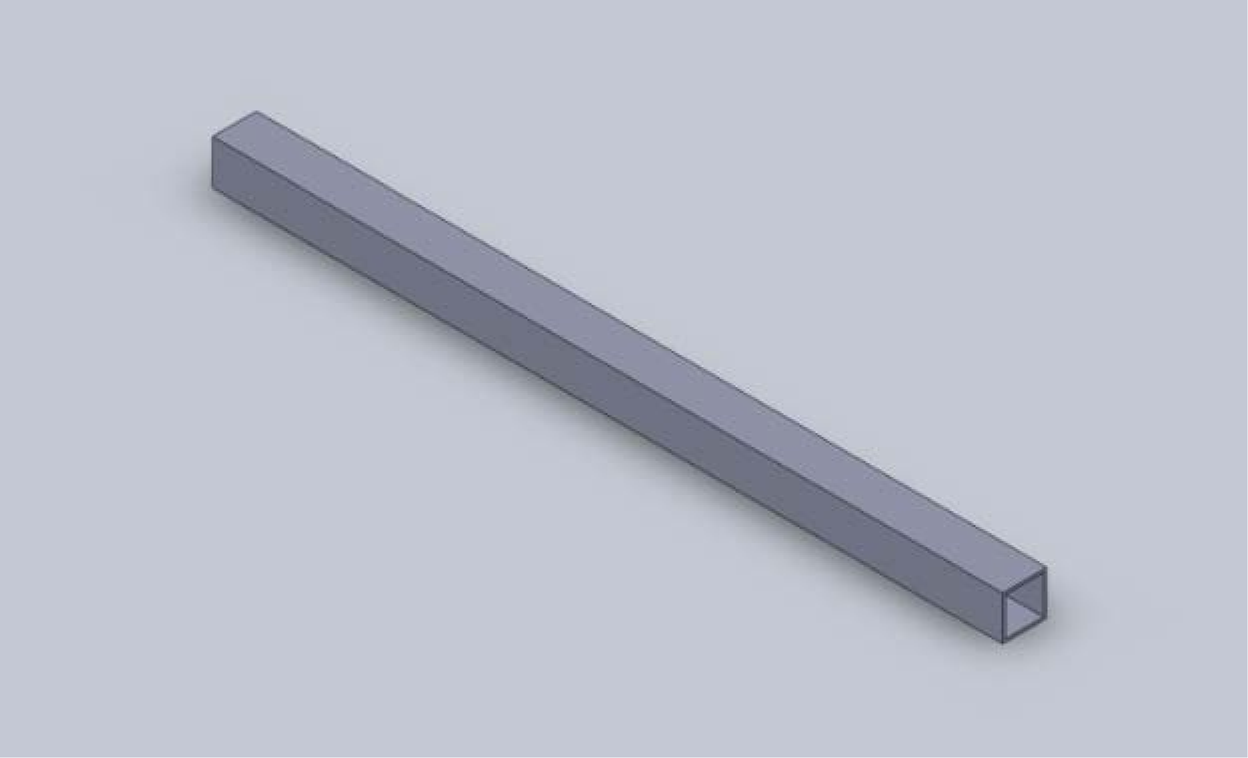
The Ascender System designed by the Roof Robot 3 MQP team should provide a low weight, cost effective solution in transporting the Roof Robot from ground level to its operating region on Roof Tops for inspections purposes. The folding base, telescoping poles, and revised ascender sled designs provide a platform of technologies that will allow future project teams to advance the Roof Robot platform to a production product. With the improvements on the Roof Robot’s electronic controls and communications systems the newer platform is now closer to being ready for real world application and use. Future MQP teams that will work on the Roof Robot platform will have a robust launching point for development.

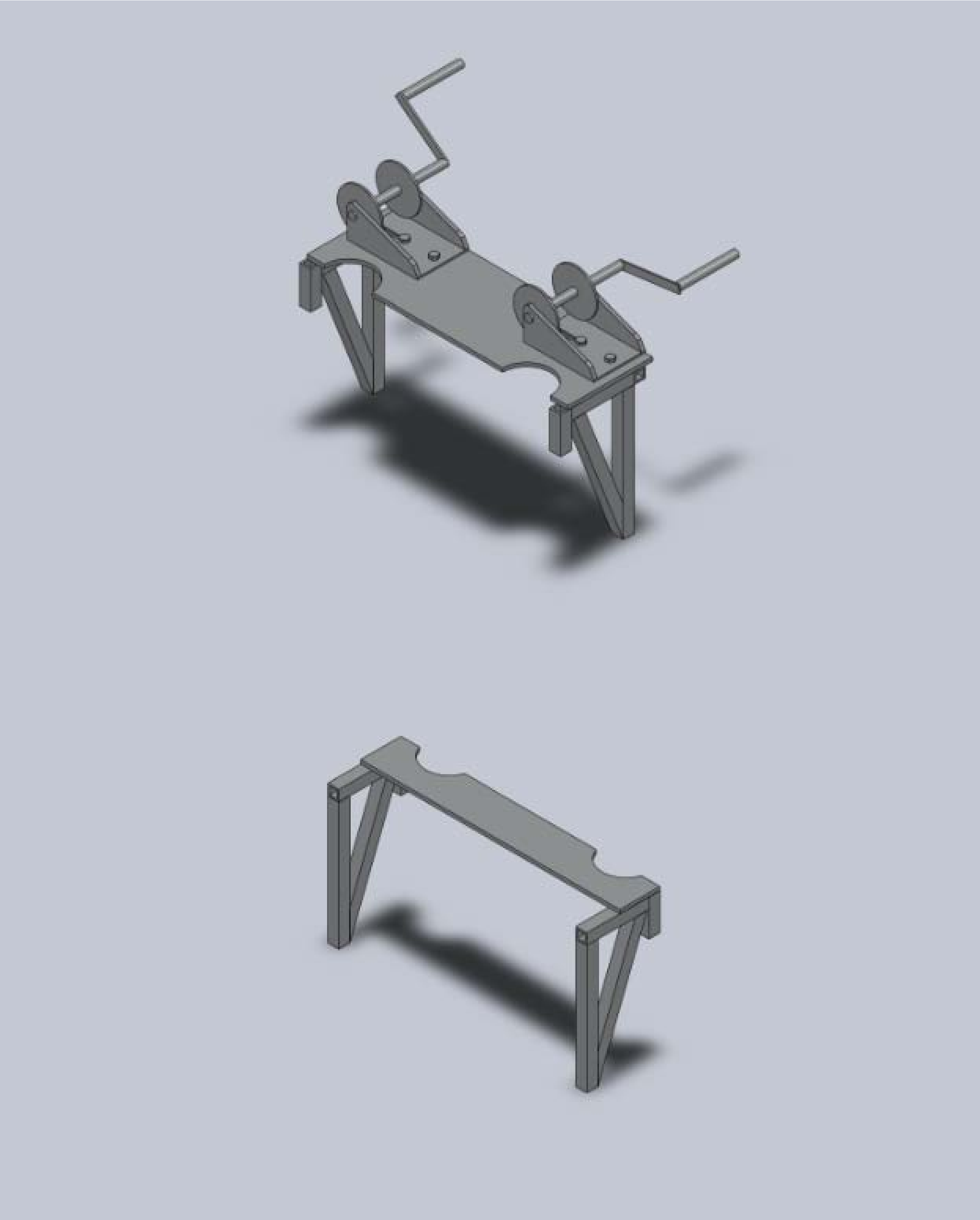
Appendix A- Base CAD Drawings

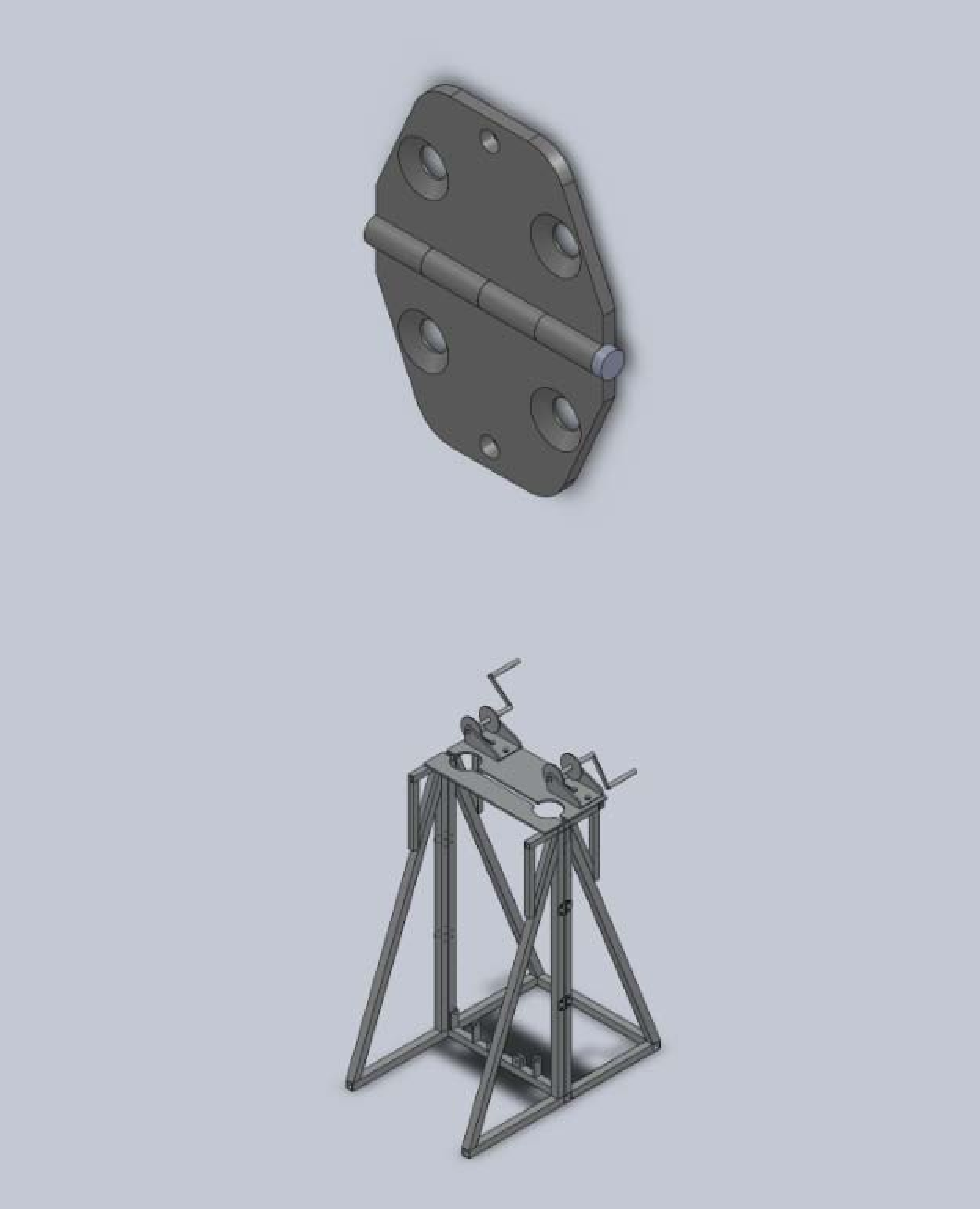


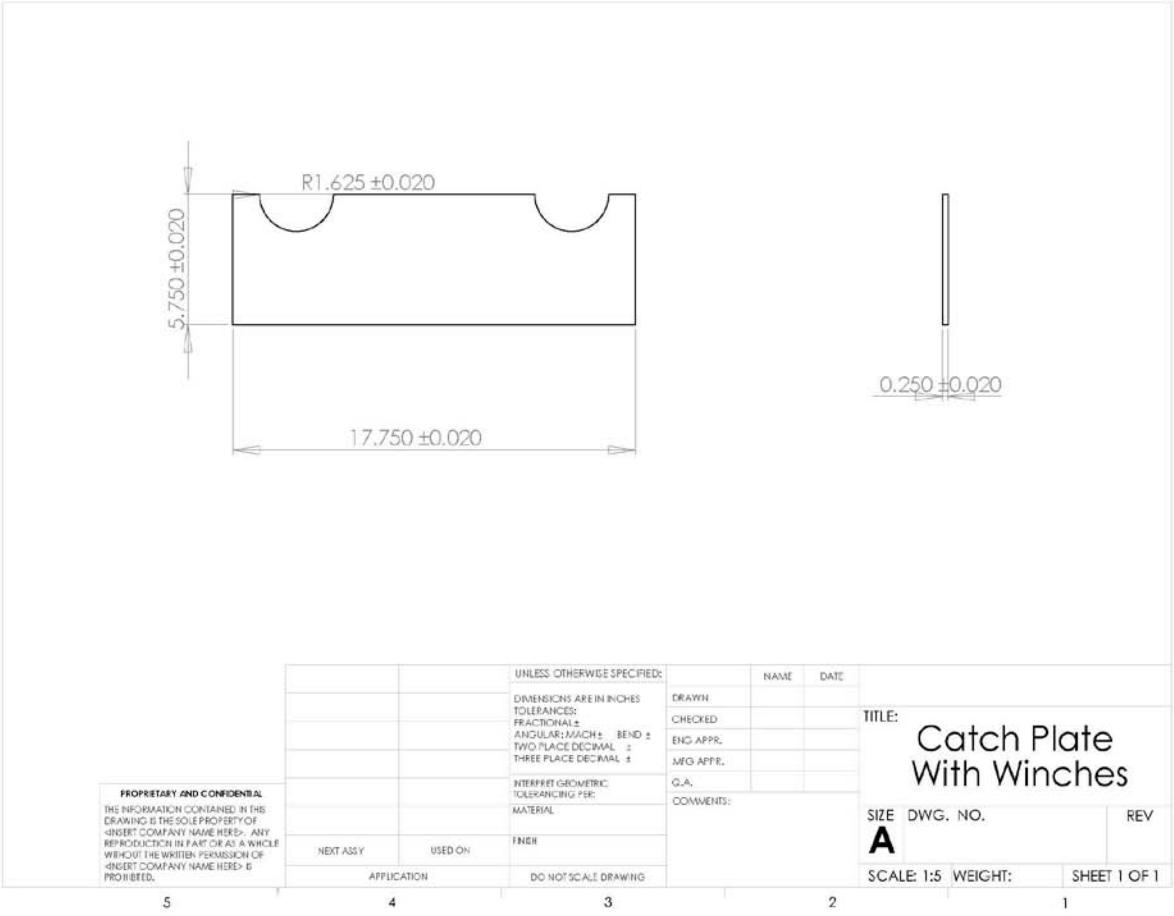


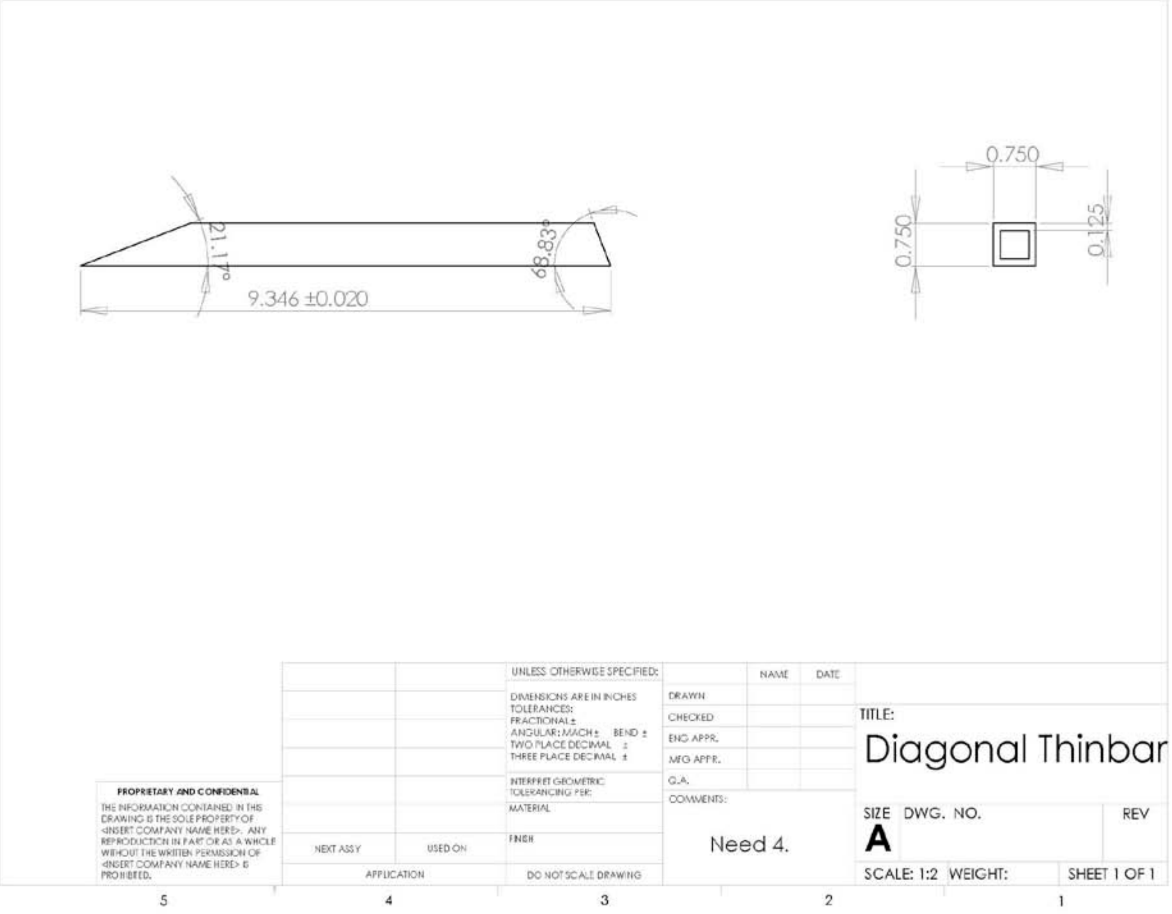


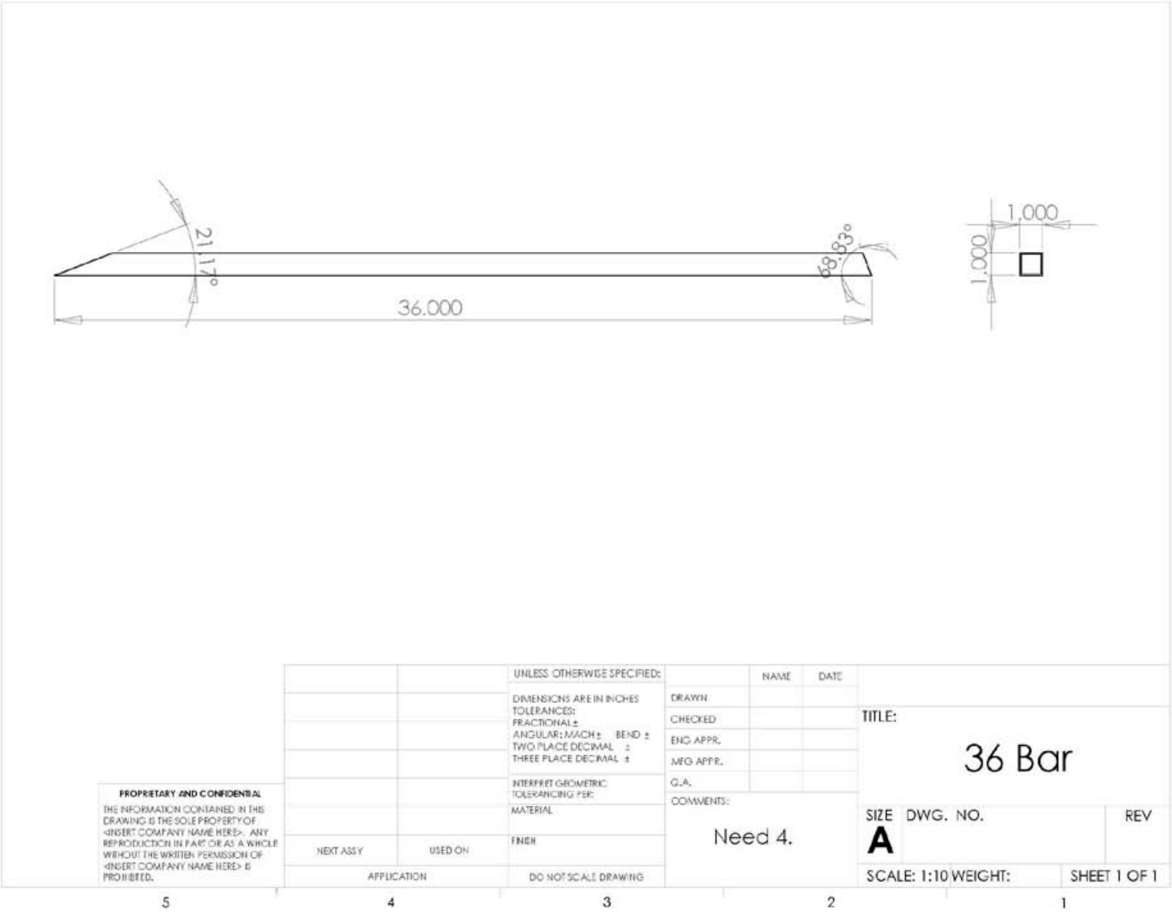







R1.625 ±0.020
5.750 ±0.020
17.750 ±0.020
0.250 ±0.020
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Catch Plate With Winches
SIZE
A
DWG. NO.
REV
SCALE: 1:5
WEIGHT:
SHEET 1 OF 1
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
5
4
3
2
1


21.17°
68.83°
9.346 ±0.020
0.750
0.750
0.125
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
Need 4.
TITLE:
Diagonal Thinbar
SIZE
A
DWG. NO.
REV
SCALE: 1:2
WEIGHT:
SHEET 1 OF 1
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
5
4
3
2
1



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY
USED ON
APPLICATION

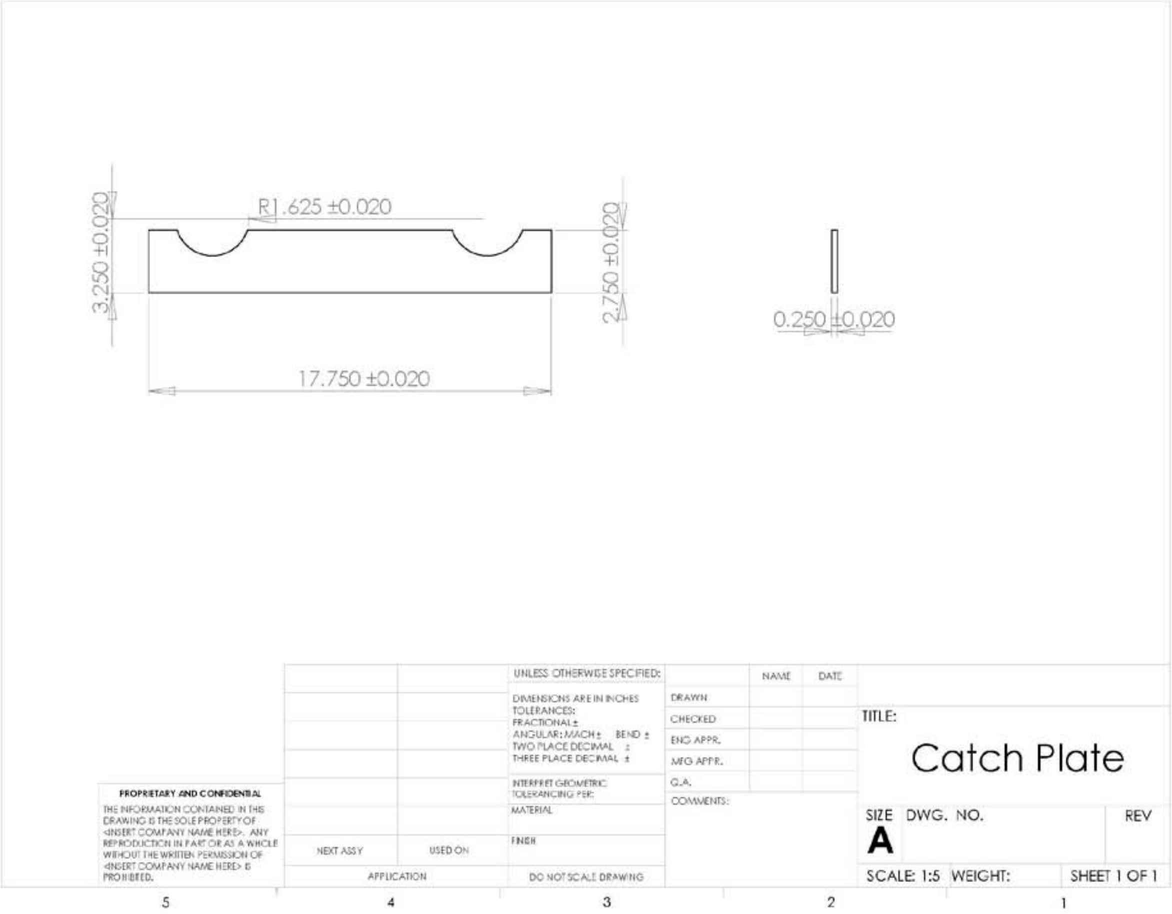
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING

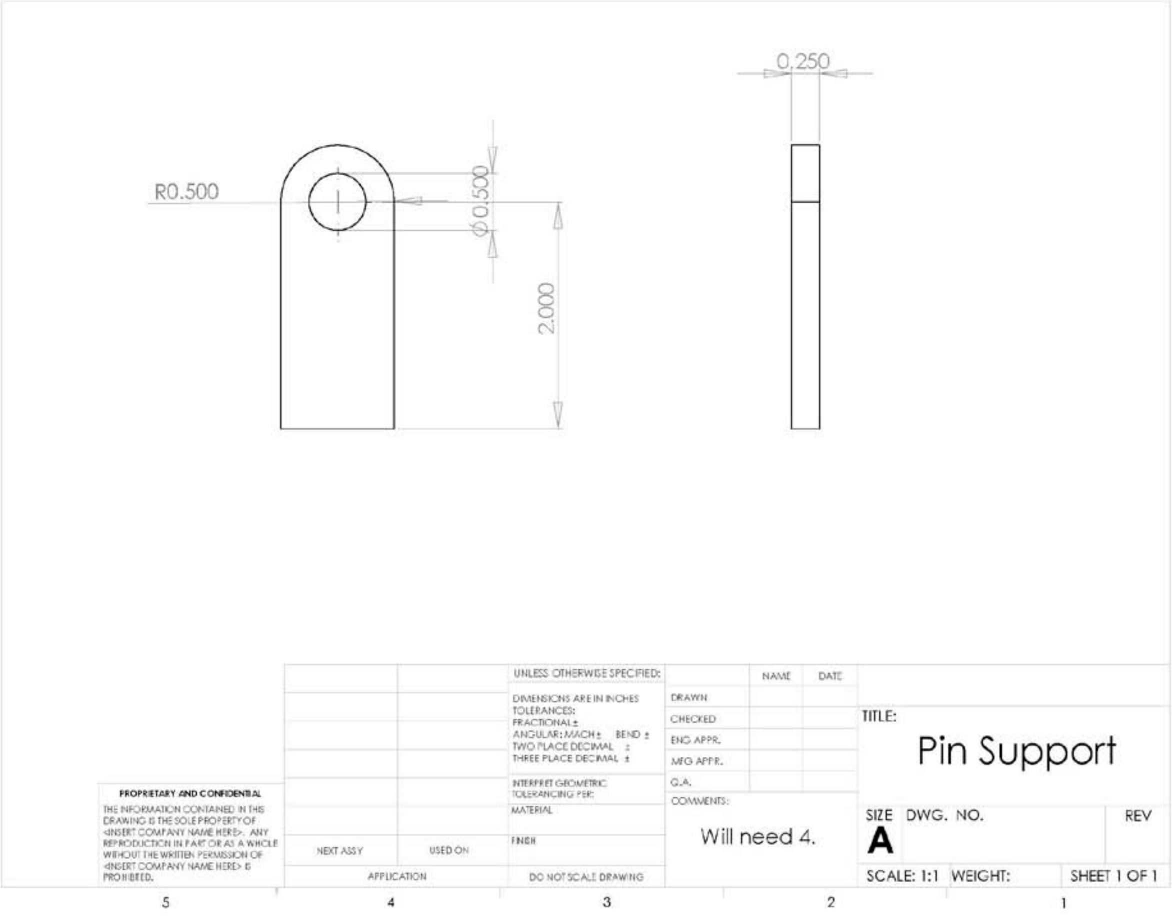
	NAME	DATE
DRAWN		
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:
Need 4.

TITLE:
36 Bar

SIZE A
DWG. NO.
REV
SCALE: 1:10
WEIGHT:
SHEET 1 OF 1


R1.625 ±0.020
3.250 ±0.020
2.750 ±0.020
17.750 ±0.020
0.250 ±0.020
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
NEXT ASSY
USED ON
APPLICATION
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Catch Plate
SIZE
A
DWG. NO.
REV
SCALE: 1:5
WEIGHT:
SHEET 1 OF 1
5
4
3
2
1



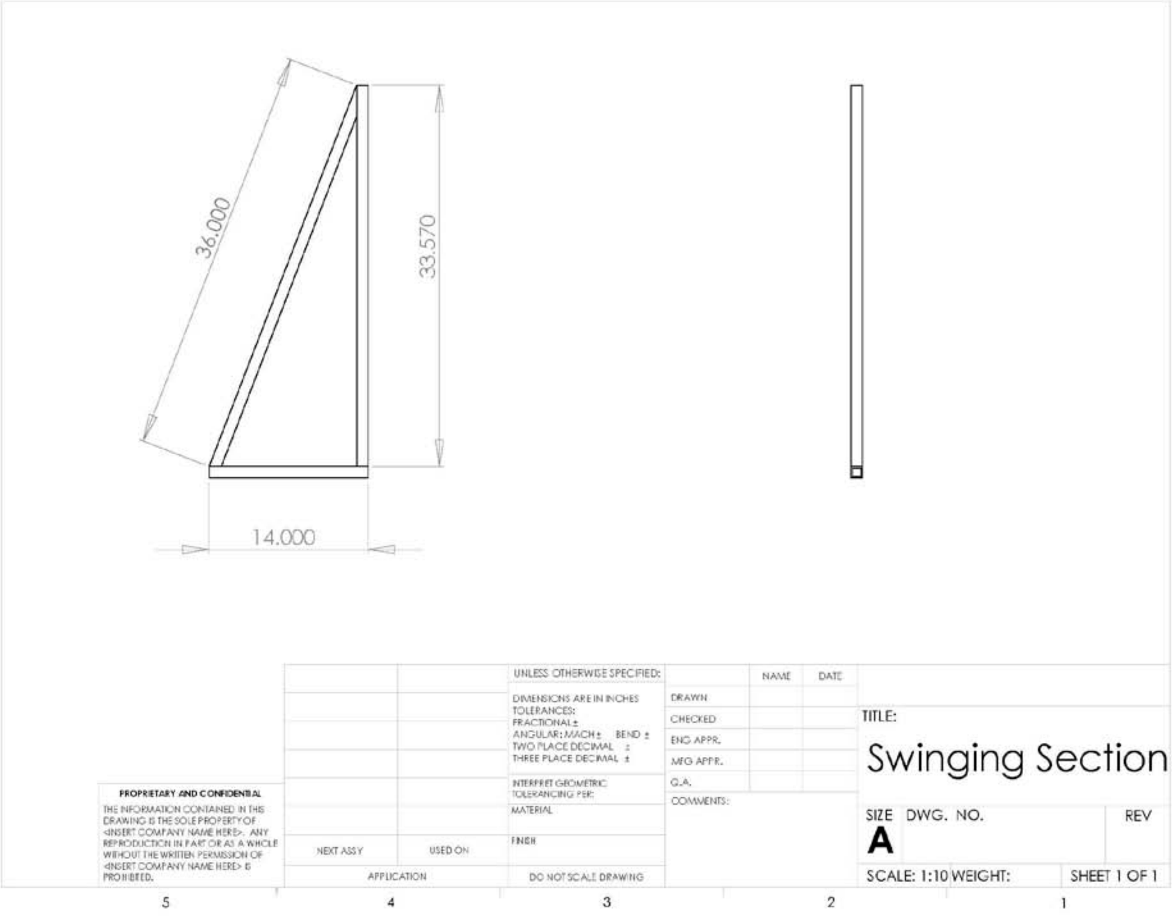
TITLE: Pin Support

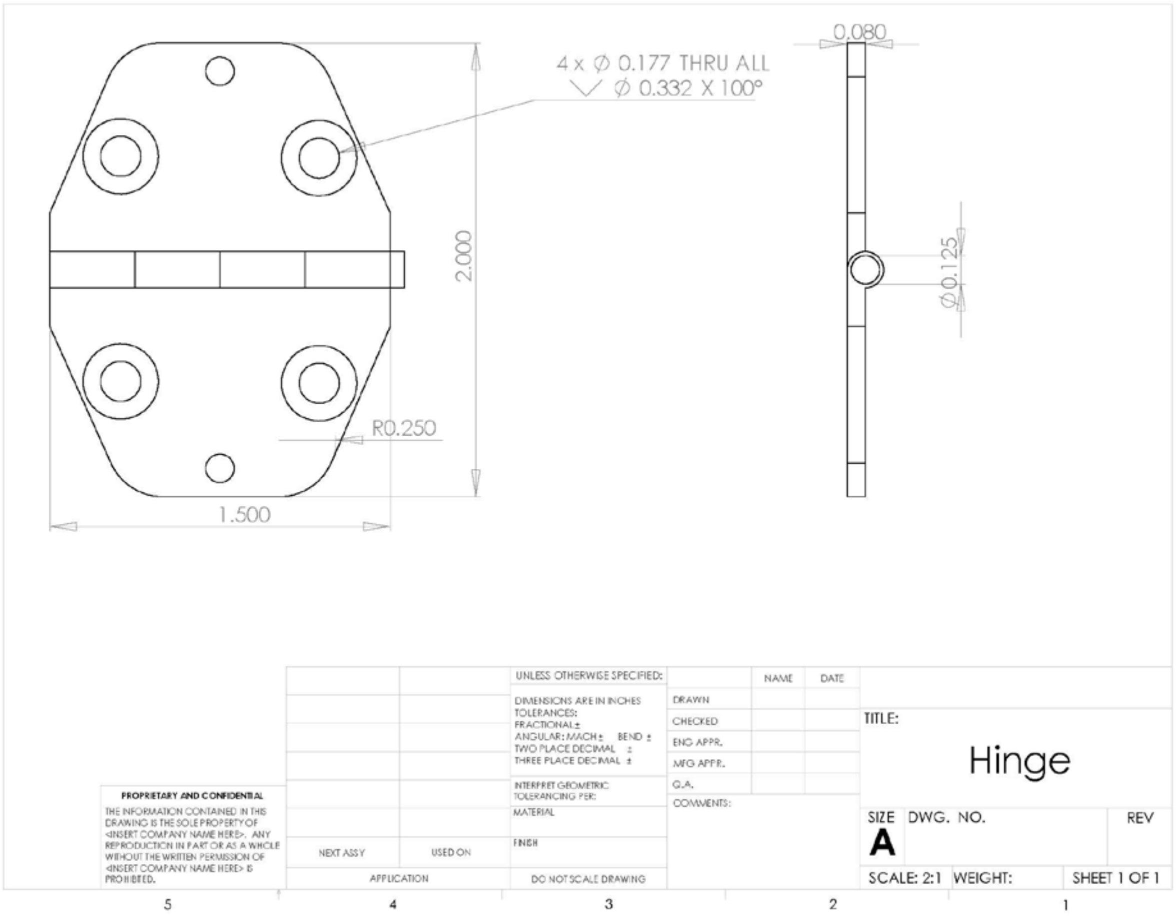
COMMENTS: Will need 4.

SIZE A

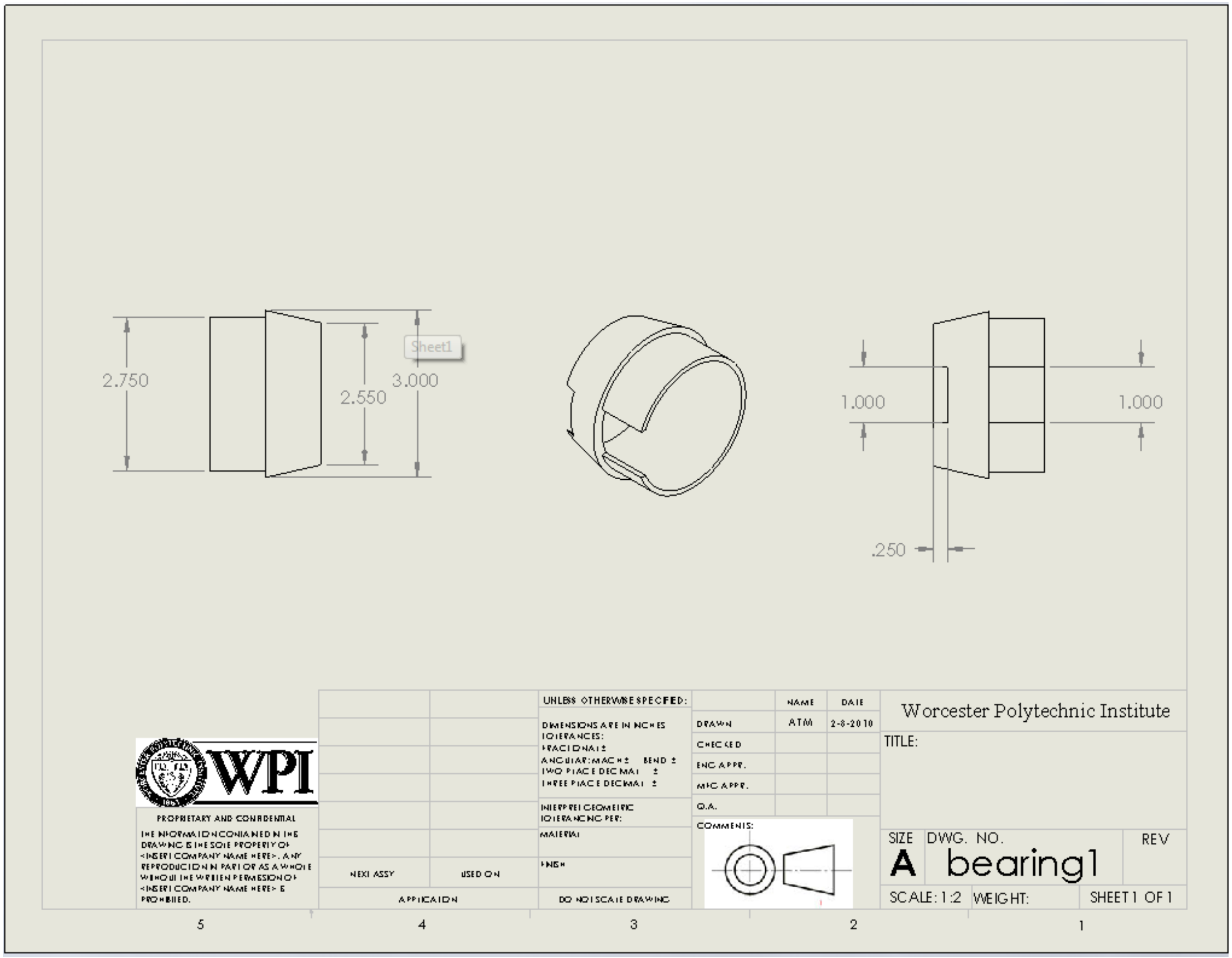
SCALE: 1:1

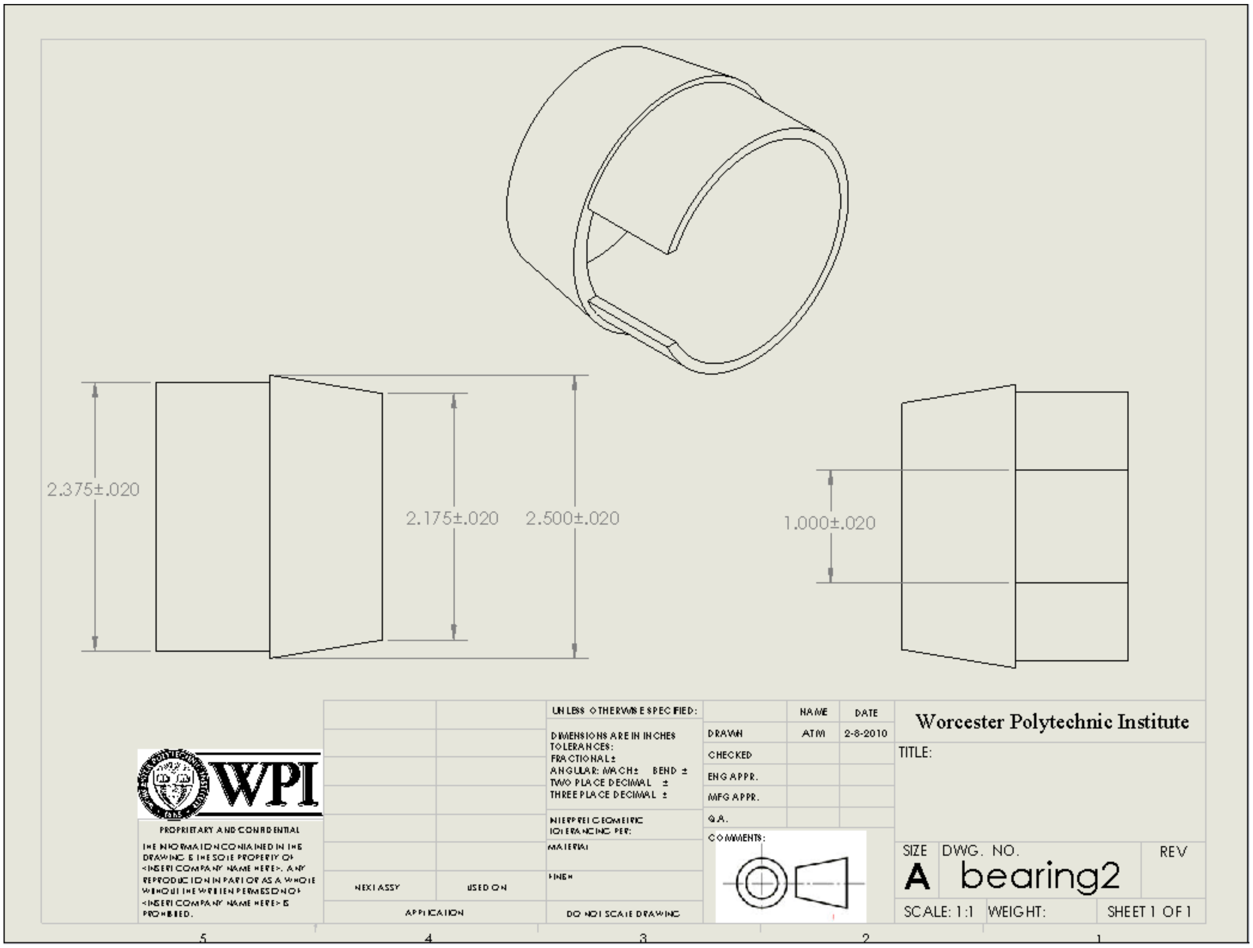
SHEET 1 OF 1


36.000
33.570
14.000
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Swinging Section
SIZE
A
DWG. NO.
REV
SCALE: 1:10
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
NEXT ASSY
USED ON
APPLICATION
5
4
3
2
1


4 x ∅ 0.177 THRU ALL
∨ ∅ 0.332 X 100°
0.080
2.000
∅0.125
R0.250
1.500
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Hinge
SIZE
A
DWG. NO.
REV
SCALE: 2:1
WEIGHT:
SHEET 1 OF 1
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
5
4
3
2
1

Appendix B- Pole CAD Drawings





Worcester Polytechnic Institute

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY | USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

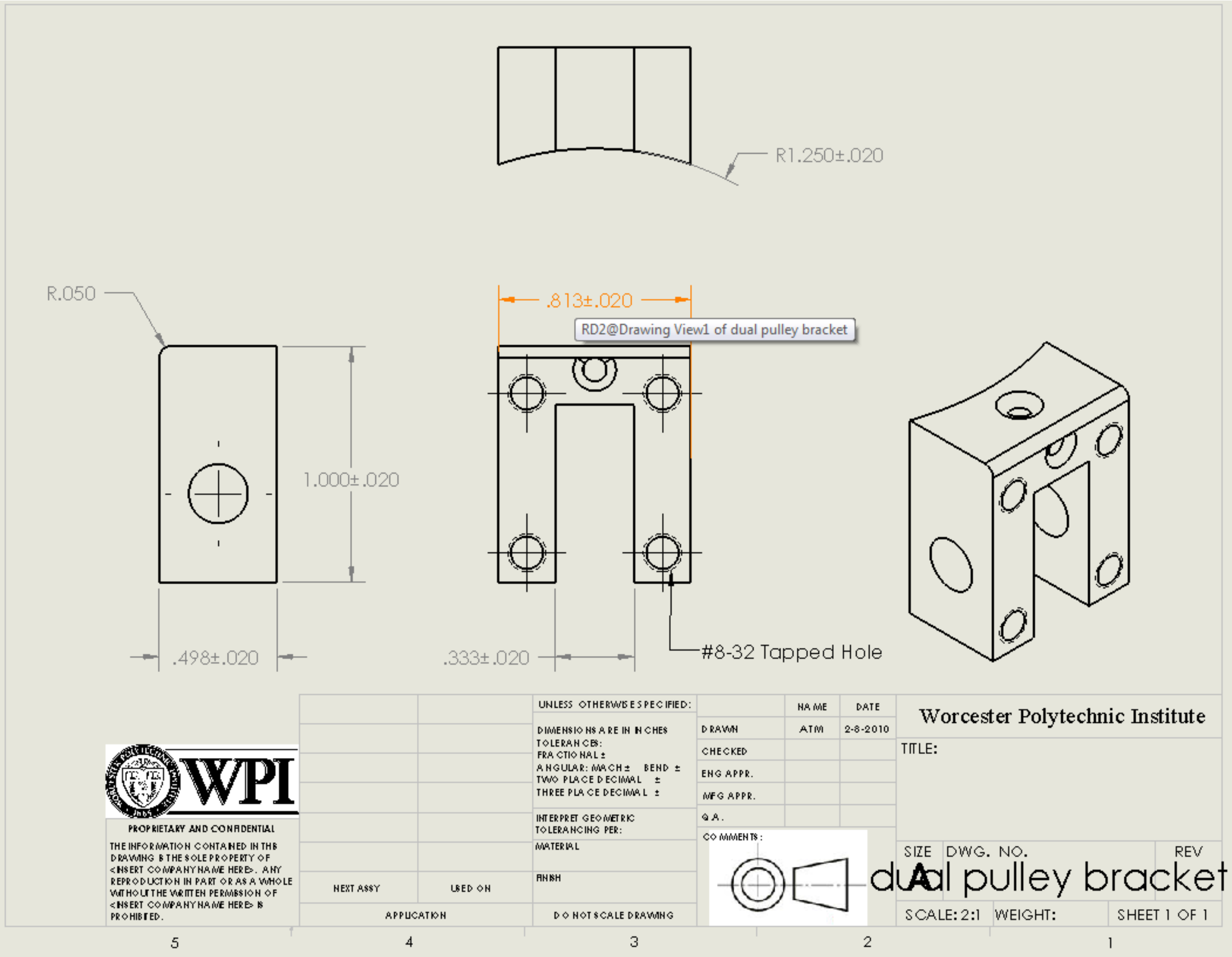
	NAME	DATE
DRAWN	ATM	2-8-2010
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

TITLE:

SIZE A
DWG. NO. bearing2
REV

SCALE: 1:1
WEIGHT:
SHEET 1 OF 1



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY | USED ON
APPLICATION

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	ATM	2-8-2010
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

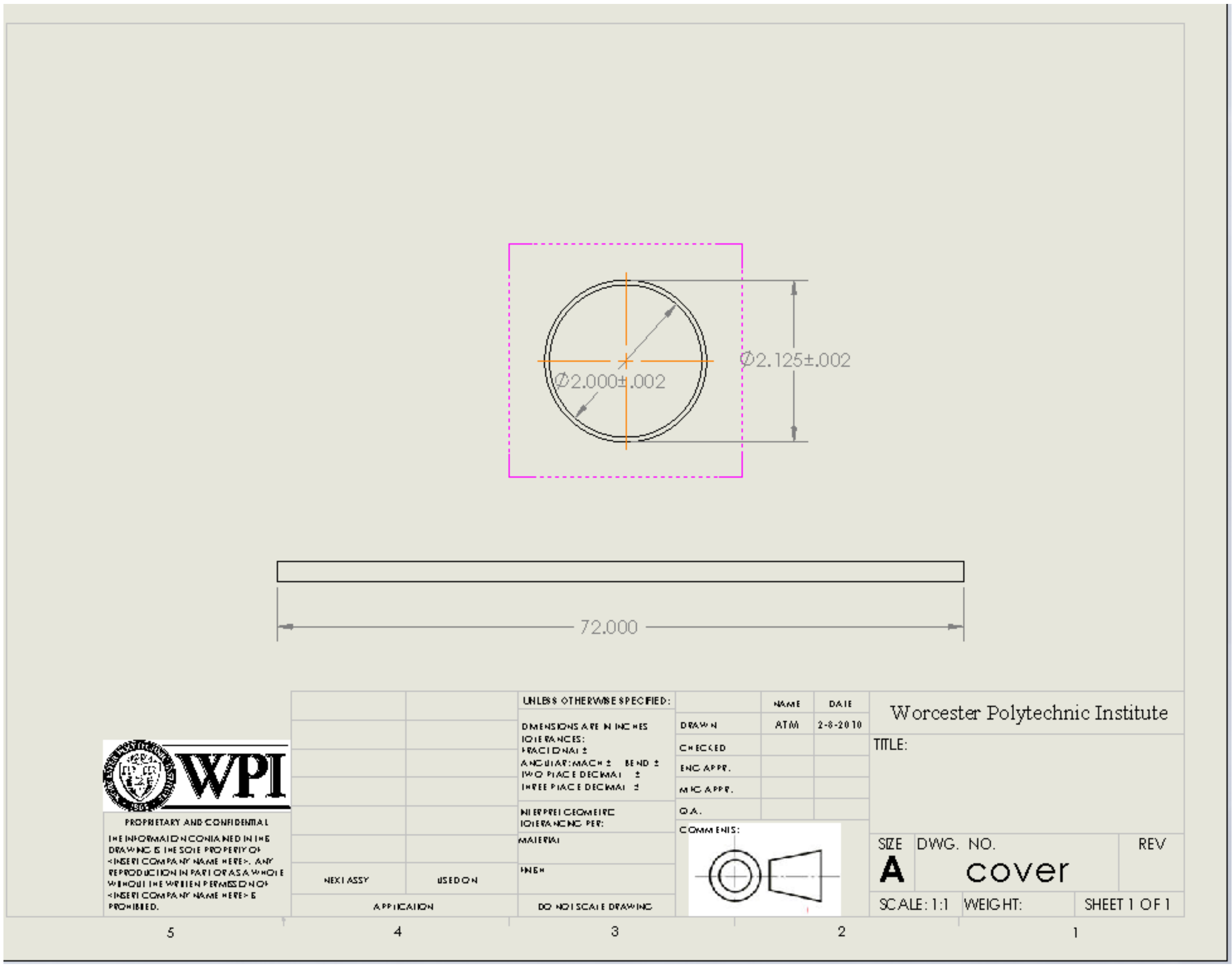
COMMENTS:

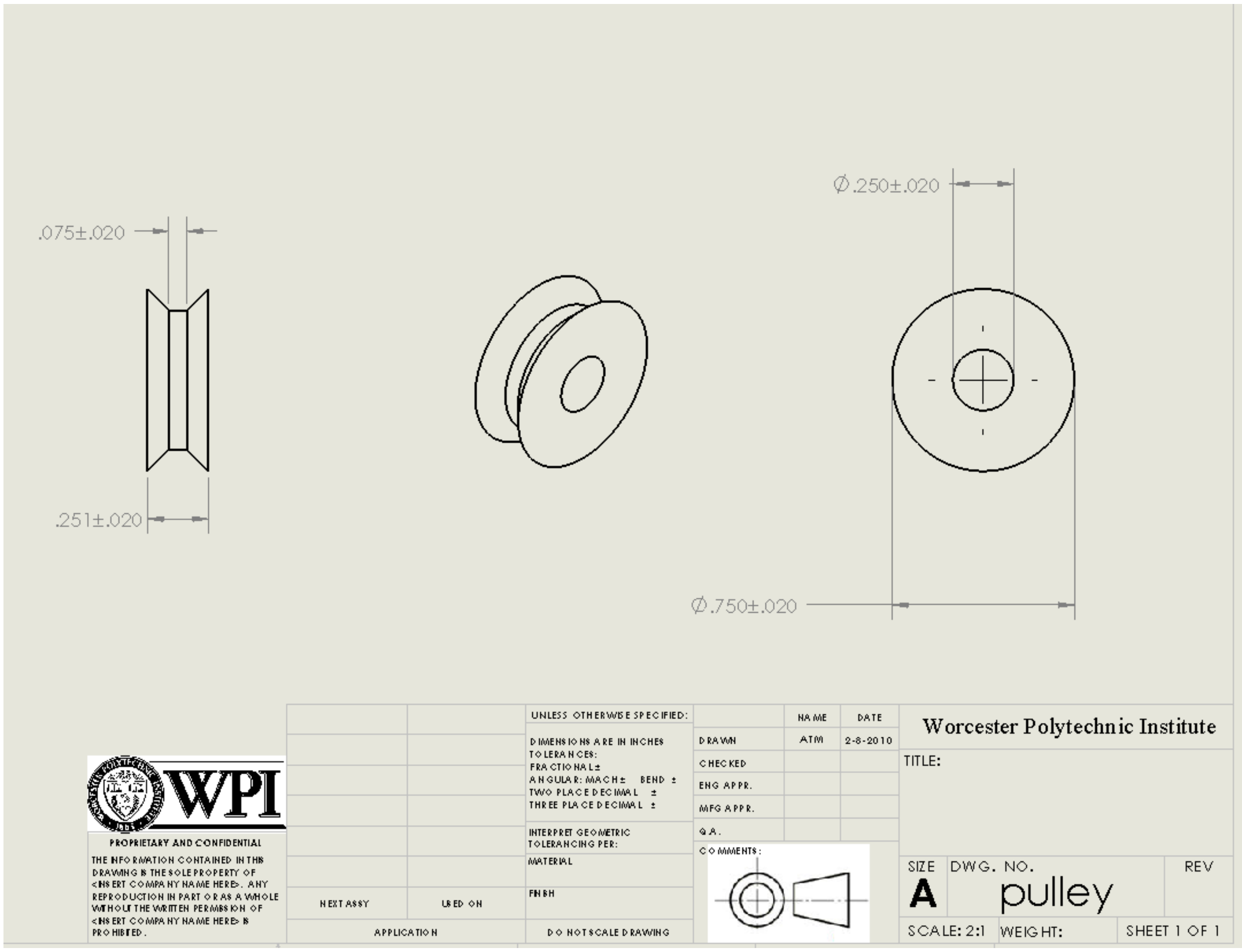
Worcester Polytechnic Institute

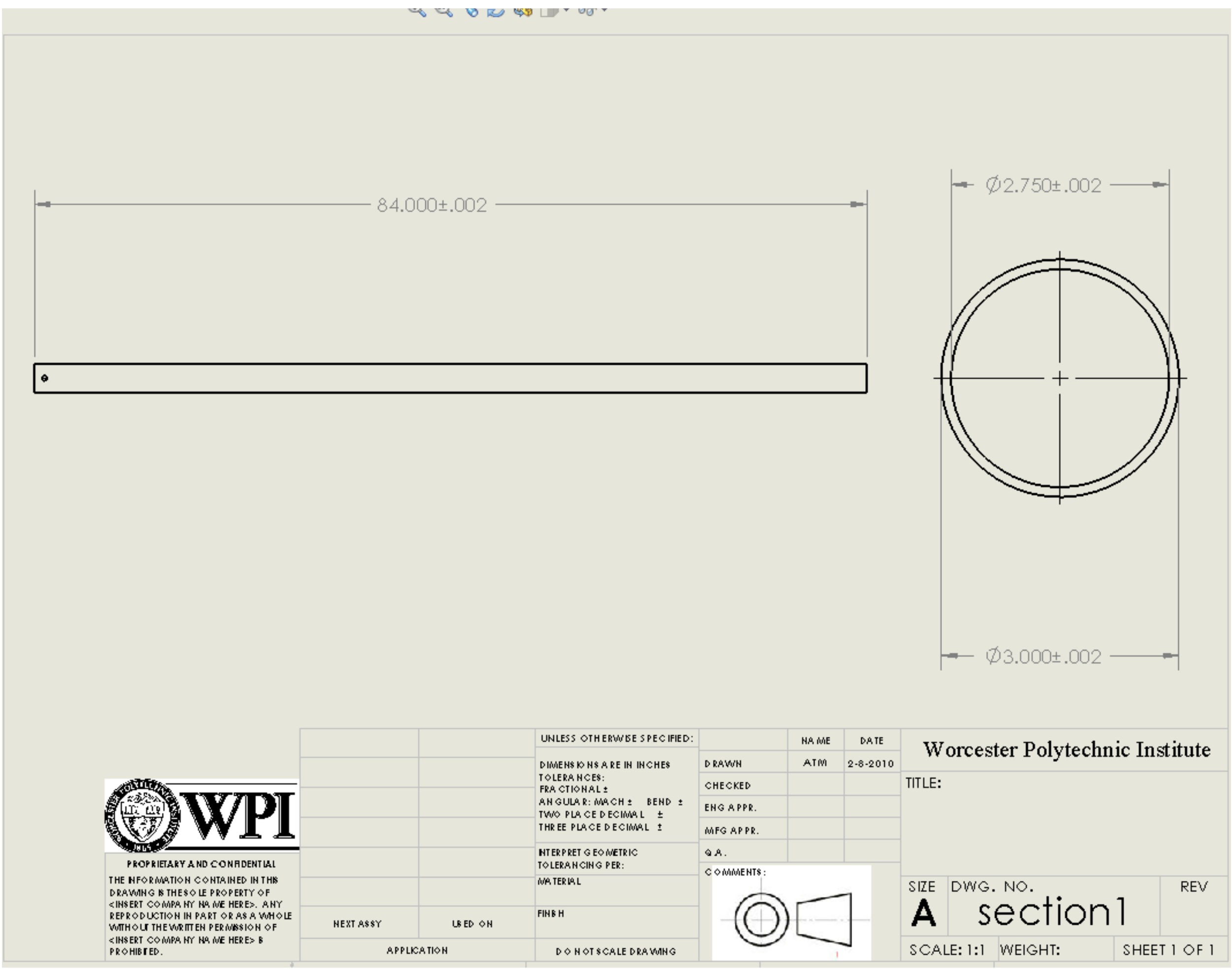
TITLE:

SIZE A DWG. NO. dual pulley bracket REV

SCALE: 2:1 WEIGHT: SHEET 1 OF 1


∅2.125±.002
∅2.000±.002
72.000
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
ATM
2-8-2010
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
Worcester Polytechnic Institute
TITLE:
SIZE
A
DWG. NO.
cover
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
5
4
3
2
1


.075±.020
.251±.020
Ø.250±.020
Ø.750±.020
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
ATM
2-8-2010
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
Worcester Polytechnic Institute
TITLE:
SIZE
A
DWG. NO.
pulley
REV
SCALE: 2:1
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.



PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

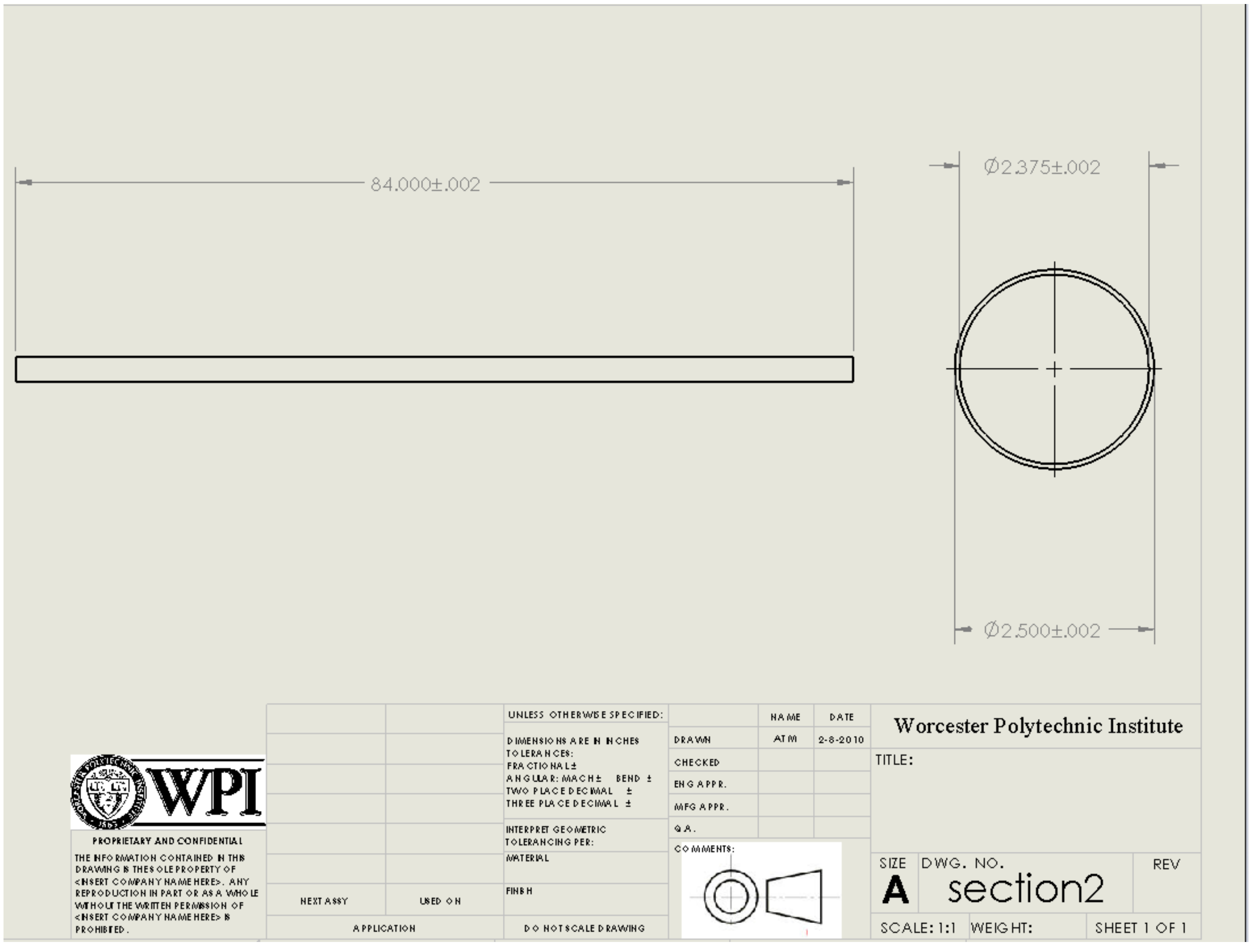
NEXT ASSY
USED ON
APPLICATION

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	ATM	2-8-2010
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

Worcester Polytechnic Institute
TITLE:
SIZE A
DWG. NO. section1
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1



Worcester Polytechnic Institute

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY
USED ON
APPLICATION

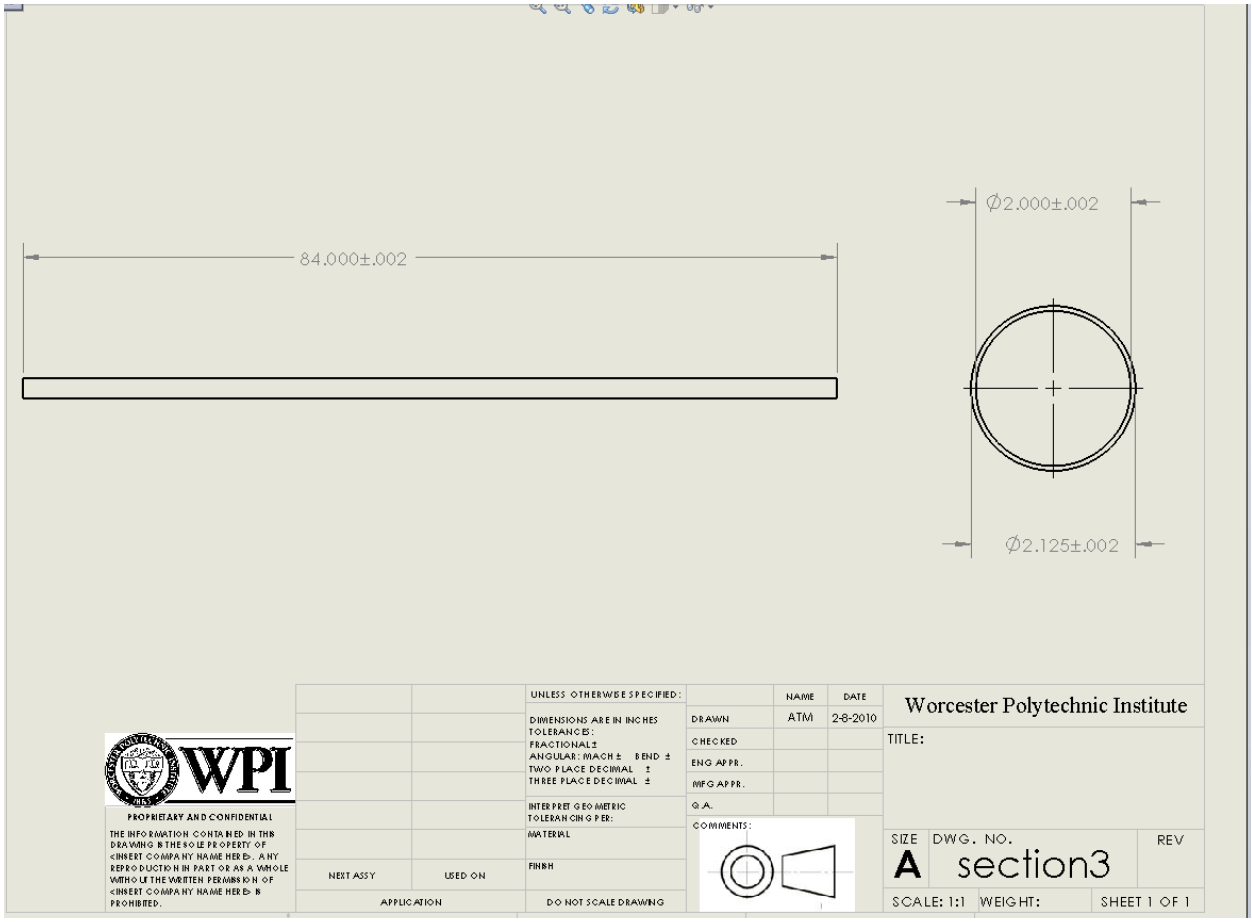
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	ATM	2-8-2010
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

TITLE:

SIZE A
DWG. NO. section2
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1



Worcester Polytechnic Institute

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	ATM	2-8-2010
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

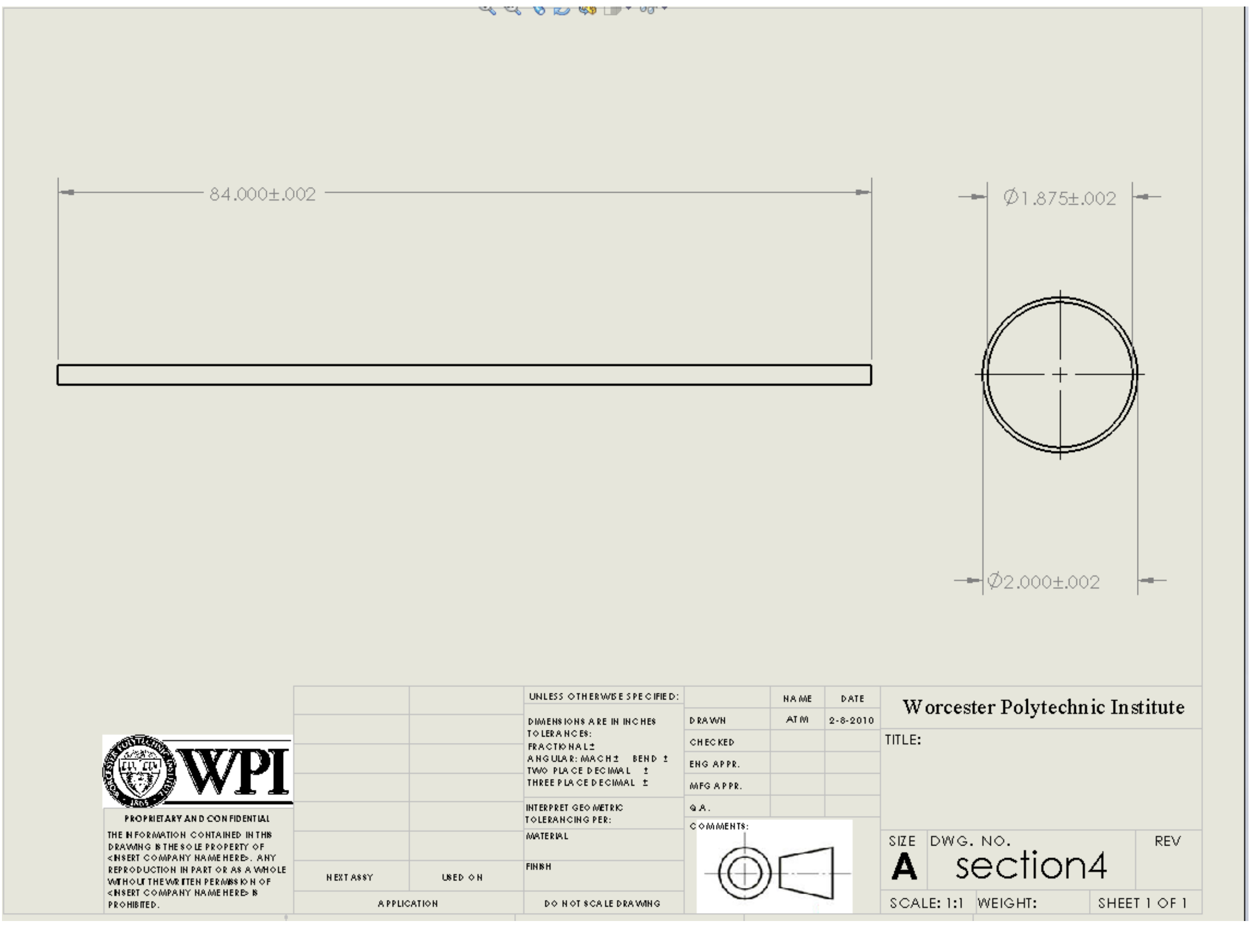
COMMENTS:

TITLE:

SIZE A
DWG. NO. section3
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY
USED ON
APPLICATION

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING

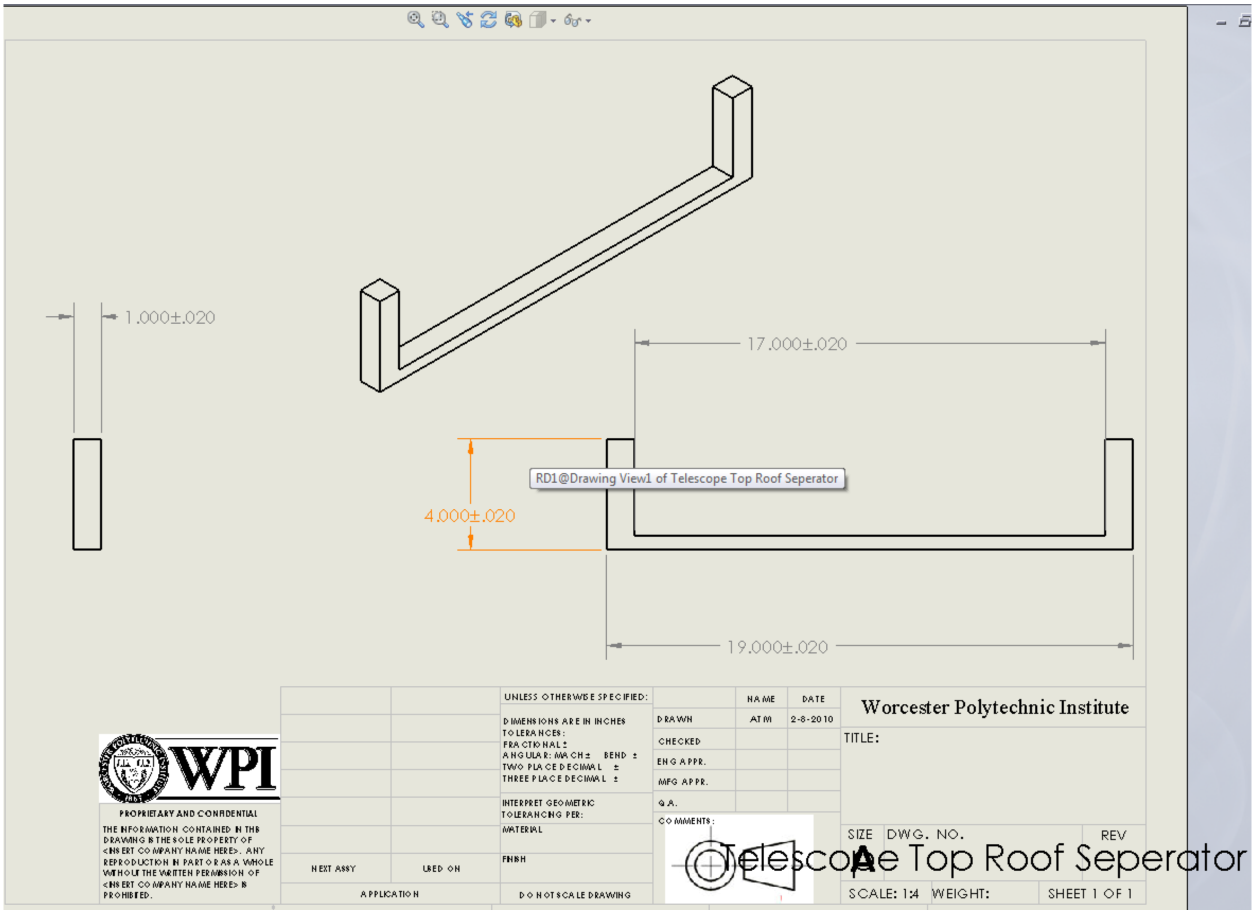
	NAME	DATE
DRAWN	ATM	2-8-2010
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

Worcester Polytechnic Institute

TITLE:

SIZE A
DWG. NO. section4
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1



WPI

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	ATM	2-8-2010
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

Worcester Polytechnic Institute

TITLE:

SIZE A DWG. NO. Telescope Top Roof Seperator REV

SCALE: 1:4 WEIGHT: SHEET 1 OF 1

Appendix C- Sled CAD Drawings

Please see attached folder

Appendix B- Robot Code

Main //This is the main section of code it calls all of the other functions.

```
#include "Main.h"
void main ( void )
{
        Intialize ( ) ; //This resets all the sensors to neutral
        while ( 1 )
        {
                get_Inputs ( ) ;  // Reads the joysticks on the remote controll
                get_Feedback ( ) ; // Reads the center joint and encoders
                updateTargets ( ) ;  // Does the math to determine desired motor outputs
                target[0]=targetFL ; // Assigns desired speed to individual wheels
                target[1]=targetFR ; //
                target[2]=targetBL ; //
                target[3]=targetBR ; //
                calcSpeed ( ) ;        //Traction control, adjusts wheel speed accordingly
                Check_Battery ( ) ; // Display Battery Voltage if 5 is pressed
                RangeFinders ( ) ;  //Ensures robot does not slips off the roof
                Drive ( ) ;             //Outputs desired speed to motors
                Look ( ) ;              //Outputs desired position to servos
        }
}
```

adjustforError //Aids in traction control and ensures motors do not exceed max output

```
#include "Main.h"
void adjustforError ( int count )
{
        int speedBehind;
        float percentError;
        int comps;
        delay[i]++;
        if ( delay[i]<2 ) //Checks the error found in calcSpeed()
        {
                return ;
        }
        Else   //assigns each wheel its opposite
        {
                delay[i]=0 ;
        }
        if ( i==0 )
        {
                comps=2 ;
        }
        if ( i==1 )
        {
                comps=3 ;
        }
        if ( i==2 )
        {
                comps=0 ;
        }
```

```
        if ( i==3 )
        {
                comps=1 ;
        }
        error=clicksPerSec[i]-expectedClicksPerSec[i] ;  //Compares the error on opposite wheels
        percentError = error / expectedClicksPerSec[i] ;
        speedBehind = (int)(clicksPerSec[i] - clicksPerSec[comps]) ;
        if ( percentError > .1 ) //If the wheels are out of sync change the output
        {
                pWheel = (-percentError * 2) - 1 ;
        }
        else if ( percentError < -.1 )
        {
                pWheel = (-percentError * 2) + 1 ;
        }
        else
        {
                pWheel = 0 ;
        }
        if ( percentError < -.1 && speedBehind < -12 ) //Compares the errors to an accepted rate
        {
                pWheel = pWheel+10 ;
        }
        if ( target[i] > 0 ) //Determines if the wheels are too fast or too slow
        {
                correction=pWheel ;
        }
        if ( target[i] < 0 )
        {
                correction=-pWheel ;
        }
        output[i] = output[i] + correction ; // adjusts the output
        if ( target[i] == 0 )
        {
                output[i] = 0 ;
        }
        if ( target[i] > 0 && output[i] < 12 )
        {
                output[i] = 12 ;
        }
        if ( target[i] < 0 && output[i] > -12 )
        {
                output[i] =-12 ;
        }
        if ( output[i] > 127 )  //ensures the outputs don’t max out
        {
                output[i] = 127 ;
        }
        if ( output[i] < -127 )
        {
                output[i] = -127 ;
        }
}
```

```
calcSpeed  //Traction Control
#include "Main.h"
void calcSpeed ( void )
{
    for ( i=0 ; i<4 ; i++ )  //Compares timers to encoders
    {
        expectedClicksPerSec[i]=(float)(target[i])*.85 ;
        if ( target[i]<0 )
        {
        expectedClicksPerSec[i] = -1 * expectedClicksPerSec[i] ;
        }
        expectedTimeBetweenClicks = (1200 / expectedClicksPerSec[i]) ;
        if ( expectedTimeBetweenClicks > 120 )
        {
        expectedTimeBetweenClicks = 120 ;
        }
        if ( newCount[i] > oldCount1[i] )
        {
            timeInterval = newClickTime[i] - oldClickTime2[i] ;
            numClicks = newCount[i] - oldCount2[i] ;
            oldClickTime2[i] = oldClickTime1[i] ;
            oldClickTime1[i] = newClickTime[i] ;
            stuckWaiting[i] = newClickTime[i] ;
            oldCount2[i] = oldCount1[i] ;
            oldCount1[i] = newCount[i] ;
            clicksPerSec[i] = 1000 / (float)(timeInterval / numClicks) ;
            adjustforError ( i ) ;
        }
        if ( (newClickTime[i] - oldClickTime1[i]) > expectedTimeBetweenClicks )//indicates no
    //slippage
        {
            clicksPerSec[i] = 0 ;
        }
        if ( (newClickTime[i] - stuckWaiting[i]) > expectedTimeBetweenClicks )//if slippage
        //occurs adjust for error
        {
            stuckWaiting[i] = newClickTime[i] ;
            delay[i] = 2 ;
            adjustforError ( i ) ;
        }
    }
}
```

```
Check_Battery  //Provides for battery feedback
#include "Main.h"
void Check_Battery ( void )
{
    Button5 = GetRxInput ( 1 , BTN5 ) ;
    Battery = GetMainBattery ( ) ;
    if ( Button5==FullFwd )
    {
    PrintToScreen("The voltage remaining is: %.2f Volts\n\n\n",Battery);
    }
```

```
}
```

Drive //outputs final speed to motors

```
#include "Main.h"
void Drive ( void )
{
        for ( i=0 ; i<= 3 ; i++ ) //ensures the motors don’t max out
        {
                if ( output[i] > 127 )
                {
                        output[i] = 127 ;
                }
                if ( output[i] < -127 )
                {
                        output[i] = -127 ;
                }
        }
        SetMotor ( FL_Motor , output[0]+servoNeutral ) ; //outputs speed plus neutral
        SetMotor ( FR_Motor , output[1]+servoNeutral ) ;
        SetMotor ( BL_Motor , output[2]+servoNeutral ) ;
        SetMotor ( BR_Motor , output[3]+servoNeutral ) ;
        if ( targetAngPWM>-5 && targetAngPWM<5 )  //Prevents a dead band
        {
                targetAngPWM=0 ;
        }
        if ( targetAngPWM>127 ) //ensures center motor doesn’t max out
        {
                targetAngPWM=127 ;
        }
        if ( targetAngPWM<-127 )
        {
                targetAngPWM=-127 ;
        }
        targetAngPWM=targetAngPWM+centerNeutral ;  //creates speed plus neutral
        SetMotor ( Center_Motor , targetAngPWM ) ;      //outputs speed plus neutral
}
```

---

get_Feedback //gets feedback from components

```
#include "Main.h"
void get_Feedback ( void )
{
        int countDiff1;
        int countDiff2;
        long timeDiff1;
        long timeDiff2;
        double halfpot;
        potReading = GetAnalogInput ( POT ) ;//Read the center potentiometer
        potAngle = ((float) potReading - potNeutral)*0.087890625 ; // this magic number takes the
        //1024 bit pot and puts converts it into 90 degrees of freedom (MN = 1024/90)
        // To change the degrees of freedom replace 90 with the desired degrees of freedom.
        halfpot = potAngle / 2 ;
        if ( (potAngle <1) && (potAngle>-1) ) //Prevents a dead zone
```

```
    {
        halfTangent = 0.0001 ;
    }
    else
    {
        if ( halfpot<0 ) //math.h did not work for this code.  A lookup table was needed with
        //the tangent of half the angle of the potentiometer.  It contains values for integer
        //ranges of degrees from zero to fifty.
        {
            halfTangent=lookup[-(int)halfpot] ;
            halfTangent=0-halfTangent ;
        }
        else
        {
            halfTangent=lookup[(int)halfpot] ;
        }
    }
    for ( i=0 ; i<=3 ; i++ )  //Traction control comparing timers to encoders
    {
        newCount[i] = GetEncoder(i+6);
        newClickTime[i] = GetTimer(i+1);
        if ( (newCount[i] > 32000) )
        {
            countDiff1 = newCount[i] - oldCount1[i];
            countDiff2 = oldCount2[i] - oldCount1[i];
            oldCount2[i] = 0;
            oldCount1[i] = countDiff2 ;
            newCount[i] = countDiff1 + countDiff2;
            PresetEncoder ( i+6 , newCount[i] ) ;
        }
        if ( newClickTime[i] > 1000000000 )
        {
            timeDiff1 = newClickTime[i] - oldClickTime1[i];
            timeDiff2 = oldClickTime1[i] - oldClickTime2[i];
            oldClickTime2[i] = 0;
            oldClickTime1[i] = timeDiff2 ;
            newClickTime[i] = timeDiff1 + timeDiff2;
            stuckWaiting[i] = newClickTime[i];
            PresetTimer ( i+1 , newClickTime[i] ) ;
        }
    }
}
```

---

get_Inputs //Reads the values of the joysticks

```
#include "Main.h"
void get_Inputs ( void )
{
    Rx1 = GetRxInput ( 1 , 1 ) ;
    Rx2 = GetRxInput ( 1 , 2 ) ;
    Rx3 = GetRxInput ( 1 , 3 ) ;
    Rx4 = GetRxInput ( 1 , 4 ) ;
}
```

Intialize

```
#include "Main.h" //Initializes the robot
void Intialize ( void )
{
	SetMotor ( FL_Motor , Stop ) ; //Sets all motors to zero output
	SetMotor ( FR_Motor , Stop ) ;
	SetMotor ( BL_Motor , Stop ) ;
	SetMotor ( BR_Motor , Stop ) ;
	get_Feedback ( ) ;  //Gets the Center Joint Information
	targetAngle=potAngle ;
	D=potReading ;
	panTiltDelayCount=0 ;
	SetMotor ( Center_Motor , centerNeutral ) ; //Sets the center joint to neutral
	SetMotor ( Tilt , -127 ) ; //Sets the servos to neutral
	SetMotor ( Pan , Stop ) ;
	StartEncoder ( FL_Encoder ) ; //Starts the timers and resets the encoders
	PresetEncoder ( FL_Encoder , 0 ) ;
	PresetTimer ( 1 , 0 ) ;
	StartTimer ( 1 ) ;
	StartEncoder ( FR_Encoder ) ;
	PresetEncoder ( FR_Encoder , 0 ) ;
	PresetTimer ( 2 , 0 ) ;
	StartTimer ( 2 ) ;
	StartEncoder ( BL_Encoder ) ;
	PresetEncoder ( BL_Encoder , 0 ) ;
	PresetTimer ( 3 , 0 ) ;
	StartTimer ( 3 ) ;
	StartEncoder ( BR_Encoder ) ;
	PresetEncoder ( BR_Encoder , 0 ) ;
	PresetTimer ( 4 , 0 ) ;
	StartTimer ( 4 ) ;
}
```

Lockout //Stops the robot

```
#include "Main.h"
void Lockout ( void )
{
	SetMotor ( 2 , 0 ) ;
	SetMotor ( 3 , 0 ) ;
	SetMotor ( 4 , 0 ) ;
	SetMotor ( 6 , 0 ) ;
	SetMotor ( 7 , 0 ) ;
}
```

Look //Sets the servo outputs

```
#include "Main.h"
void Look ( void )
{
	targetTilt = GetRxInput ( 1 , 3 ) ;
	targetTilt=-targetTilt+117 ; //Sets neutral
	targetPan=targetPan+0 ;
	if ( targetTilt>127 )  //ensures the servos do not max out
	{
```

```
			targetTilt = 127 ;
		}
		if ( targetTilt<-127 )
		{
			targetTilt = -127 ;
		}
		if ( targetPan>127 )
		{
		targetPan = 127 ;
		}
			if ( targetPan<-127 )
		{
			targetPan = -127 ;
		}
		SetServo ( Tilt , targetTilt ) ; //Outputs to the servos
		SetServo ( Pan , targetPan ) ;
}
```

<u>RangeFinders</u> // Makes sure the robot does not fall off the roof

```
#include "Main.h"
void RangeFinders ( void )
{
		Button6 = GetRxInput ( Rx0 , BTN6 ) ;
		FL_IR = GetAnalogInput ( FL_Rangefinder ) ;
		FR_IR = GetAnalogInput ( FR_Rangefinder ) ;
		BL_IR = GetAnalogInput ( BL_Rangefinder ) ;
		BR_IR = GetAnalogInput ( BR_Rangefinder ) ;
		while ( (FL_IR<DangerZone||FL_IR<DangerZone||BL_IR<DangerZone
		||BR_IR<DangerZone)&&Button6!=FullFwd )
		// Determines if the range finders are triped and if the override is pressed

		{
			Lockout ( ) ;
			PrintToScreen ( "You are near an edge, the \n" ) ;  //Lets the operator know which
			//direction to drive
			if ( FL_IR<DangerZone||FR_IR<DangerZone )
			{
				PrintToScreen ( "front sensors are tripped\n\n\n" ) ;
			}
			else
			{
				PrintToScreen ( "back sensors are tripped\n\n\n" ) ;
			}
			Wait ( 100 ) ; // in msec
			Button6 = GetRxInput ( 0 , 6 ) ;
			if ( Button6== FullFwd )
			{
				PrintToScreen ( "Override Pressed. Please drive safe.\n\n\n" ) ;
			}
		}
}
```

<u>setTargetsForDriving</u> //Determines the correct motor speed

```
#include "Main.h"
void setTargetsForDriving ( void )
{
    float maxPower = 50;  //A higher number increases speed
    float powerScaleToJoystick;
    if ( rightToLeftRatio > 1 || rightToLeftRatio < -1 ) //Determines left or right turning
    {
        powerScaleToJoystick = maxPower / rightToLeftRatio ;
    }
    else
    {
        powerScaleToJoystick = maxPower * rightToLeftRatio ;
    }
    slowSide=((float)(Rx2)/128)*powerScaleToJoystick ; //prevents a dead band
    if ( slowSide > -15 && slowSide < 0 )
    {
    slowSide=-15 ;
    }
    if ( slowSide >= 0 && slowSide < 15 )
    {
        slowSide=15 ;
    }
    if ( rightToLeftRatio>1 || rightToLeftRatio<-1 )  //Gives wheel speeds based on turn
    {
        fastSide = slowSide*rightToLeftRatio ;
        targetFR = (int)(slowSide) ;
        targetBR = (int)(slowSide) ;
        targetFL = (int)(fastSide) ;
        targetBL = (int)(fastSide) ;
    }
    else
    {
        fastSide = slowSide/rightToLeftRatio ;
        targetFR = (int)(fastSide) ;
        targetBR = (int)(fastSide) ;
        targetFL = (int)(slowSide) ;
        targetBL = (int)(slowSide) ;
    }
}
```

setTargetsforPan //Legacy Code. Not included in current iteration

```
#include "Main.h"
void setTargetsforPan ( void )
{
    int turnOffset;
    int scaledPan;
    float scale = 1;
    if ( panTiltDelayCount < panTiltDelay )
    {
        if ( (Rx3 < -15) && Rx3 > -127 )
        {
            //targetTilt-- ;
        }
```

```
        if ( (Rx3 > 15) && Rx3 < 127 )
        {
            //targetTilt++ ;
        }
        panTiltDelayCount++ ;
    }
    else
    {
        panTiltDelayCount=0 ;
    }
    turnOffset = (Rx1) / 1.5 ;
    if ( (Rx4) >= 15 )
    {
        scale = (float)(127 - turnOffset) / 127 ;
    }
    if ( (Rx4) <= 15 )
    {
        scale = (float)(turnOffset-(-127)) / 127 ;
    }
    scaledPan = (Rx4) * scale ;
    targetPan = (scaledPan + turnOffset) ;
}
```

<u>setTargetsForTurning </u> //Determines wheel speed and center pivot for turning while stationary

```
#include "Main.h"
void setTargetsForTurning ( float joystickInD )
{
    targetAngle=joystickInD ;
    if ( (potAngle - targetAngle) > -2 && (potAngle - targetAngle) < 2 )//prevents dead zone
    {
        slowSide=0 ;
        I=0 ;
    }
    else
    {
        if ( potAngle< targetAngle ) //Turn left or turn right
        {
            slowSide=-12 ;
            if ( I>-12 )
            {
                I=-12 ;
            }
            else
            {
                I-- ;
            }
        }
        if ( potAngle> targetAngle )
        {
            slowSide=12 ;
            if ( I<12 )
            {
                I=12 ;
```

```
            }
            else
            {
                I++ ;
            }
        }
    }
    D=potReading ;
    targetAngPWM=(I) ;
    if ( rightToLeftRatio>1 || rightToLeftRatio<-1 )//Sets wheel speed
    {
        fastSide = slowSide*rightToLeftRatio ;
        targetFR = (int)(slowSide) ;
        targetBR = (int)(-slowSide) ;
        targetFL = (int)(-fastSide) ;
        targetBL = (int)(fastSide) ;
    }
    else
    {
        fastSide = slowSide/rightToLeftRatio ;
        targetFR = (int)(slowSide) ;
        targetBR = (int)(-slowSide) ;
        targetFL = (int)(-fastSide) ;
        targetBL = (int)(fastSide) ;
    }
}
```

updateTargets //Sets outputs based on joystick positions

```
#include "Main.h"
void updateTargets ( void )
{
    joystickInRadians=(float)(Rx1)/3.96875 ; // conversion factor so that a full turn ==32deg
    rightToLeftRatio =(halfWidth+(halfWidth*halfTangent))/(halfWidth-(halfWidth*halfTangent)) ;
    setTargetsForTurning ( joystickInRadians ) ;
    if ( (Rx2-servoNeutral)<-15||(Rx2-servoNeutral)>15 ) //Drive if not in dead band
    {
        setTargetsForDriving ( ) ;
    }
    setTargetsforPan ( ) ;//Move the camera
    if ( panTiltDelayCount < panTiltDelay ) //Delays Camera Movements
    {
        if ( (Rx3 < -15 || Rx3 > 15) && (Rx4 < -15 || Rx4 > 15) ) // ** && OR ||
        {
            panTiltDelayCount++ ;
        }
    }
    else
    {
        panTiltDelayCount=0 ;
    }
}
```