

WPI AIAA Research Rocket for the Investigation and Observation of Recovery and Staging

WARRIORS III

A Major Qualifying Project
Submitted to the faculty of
Worcester Polytechnic Institute
in partial fulfillment of the requirements for the
Degree of Bachelor of Science

Submitted By:

Jared Buhler	_____
Troy Coverstone	_____
Nicholas Cummings	_____
Siobhan Fleming	_____
Thomas Huleatt	_____
Theodore McDonald	_____
Patrick Renaud	_____
Megan Yocom	_____

Approved:

Prof. John Blandino

Date: March 10, 2008

Authorship Page

Acknowledgements.....	Theodore McDonald
Executive Summary	Troy Coverstone, Siobhan Fleming, Theodore McDonald & Patrick Renaud
1 Introduction.....	Troy Coverstone
2 Background.....	Siobhan Fleming
2.1 Aerodynamics	Siobhan Fleming
2.1.1 Fin Flutter.....	Siobhan Fleming
2.1.2 Mathematical Modeling Equations	Megan Yocom
2.1.3 Data Acquisition System.....	Troy Coverstone
2.1.4 Dynamic Flight Stability	Troy Coverstone
2.2 Hybrid Rocket Propulsion.....	Patrick Renaud
2.2.1 History.....	Patrick Renaud
2.2.2 Benefits of Hybrid Propulsion	Nicholas Cummings
2.2.3 Current Hybrid Rocket Research	Jared Buhler
2.2.4 Previous Grain Studies.....	Jared Buhler
2.2.5 Inefficiencies and Problems with Hybrid Propulsion	Theodore McDonald
2.2.6 Previous Studies.....	Theodore McDonald and Jared Buhler
3 Methodology	Siobhan Fleming
3.1 Initial Test Vehicles	Patrick Renaud
3.2 Aerodynamics	Siobhan Fleming
3.2.1 Test Stand.....	Tom Huleatt
3.2.2 Fin Designs	Siobhan Fleming
3.2.3 Wind Tunnel Testing	Siobhan Fleming
3.2.4 MATLAB Modeling of Flutter Equations	Megan Yocom
3.2.5 Fin Simulations by FLUENT	Thomas Huleatt
3.2.6 Data acquisition System.....	Troy Coverstone
3.3 Hybrid Rocket Propulsion.....	Theodore McDonald
3.3.1 Motor Design Criteria.....	Nicholas Cummings
3.3.2 CFD Injector Modeling.....	Jared Buhler
3.3.3 Motor Hardware Design.....	Theodore McDonald
3.3.4 Oxidizer Delivery System.....	Theodore McDonald
3.3.5 Test Stand.....	Nicholas Cummings
3.3.6 Safety Measures.....	Patrick Renaud
3.3.7 Testing Procedures	Patrick Renaud
3.3.8 Data Collection	Nicholas Cummings
3.4 Final Test Vehicle	Patrick Renaud
3.4.1 Mission Criteria.....	Patrick Renaud
3.4.2 Design	Patrick Renaud
3.4.3 Data Acquisition System.....	Troy Coverstone
4 Results and Analysis	
4.1 Initial Test Vehicles	Siobhan Fleming
4.2 Aerodynamics	Siobhan Fleming
4.2.1 2D CFD Analysis.....	Thomas Huleatt
4.2.2 Mathematical Modeling of Flin Flutter.....	Megan Yocom

4.2.3	Wind Tunnel Testing	Siobhan Fleming, and Megan Yocom
4.3	Hybrid Rocket Propulsion.....	
4.3.1	Testing Results.....	Nicholas Cummings and Theodore McDonald
4.3.2	CFD Results and Analysis	Jared Buhler
5	Conclusions.....	
5.1	Initial Test Vehicle.....	Jared Buhler
5.2	Aerodynamics	Siobhan Fleming and Megan Yocom
5.3	Hybrid Rocket Propulsion.....	
5.3.1	Ignition Delay and Chamber Design.....	Patrick Renaud
5.3.2	Injector Flow Modeling	Jared Buhler
5.3.3	Motor Operation.....	Jared Buhler
5.4	Final Test Vehicle	Patrick Renaud

Acknowledgements

The WARRIORS III MQP Student group thanks:
Advisor Professor John J. Blandino
For his encouragement, dedication and support

David Stechmann
For constant support and assistance with numerical analysis and predictions of hybrid performance

Ryan Caron
For help with electronics associated with the control of the hybrid motor

John S. DeMar
For assistance with sensor acquisition and help with propulsion fundamentals

Todd Moore of Skyrippersystems
For aid in hybrid motor design and graciously allowing us to use his software

Mike Dabrowski of Genesis One
For help troubleshooting serial communication with the onboard flight computer

Adriana Hera
For much needed assistance with software

Randall K. Harris
For help and advice with the motor setup and operation as well as allowing us to test rocket motors in the Fire Science Laboratory

Professor John Hall
For continued help with equipment acquisition and for generous donation of strain gauges

Professor John Sullivan
For lending the PCI DAQ card

Mr. Dave Messier, Manager, Environment and Occupational Safety
For assisting with permissions to test on campus

Executive Summary

The 2008 WARRIORS III MQP focused on two broad areas: 1) hybrid rocket propulsion research which explored new injector methods and 2) a study of aerodynamics including fin flutter and flow over fins. Through complete development of a paraffin fuel hybrid motor, studies of nitrous oxide injector methods were performed. Support equipment, controls, and a sensor network were assembled to allow for ground testing. Flutter characteristics were observed and measured during wind tunnel testing of proportionally sized fins of multiple geometries and materials. Two initial flight vehicles were used as test platforms for existing electronics and allowed the students to gain valuable experience in the fabrication and design techniques of high power rockets. This experience was later applied in the design and construction of a final flight vehicle that supported the testing of fin flutter with a flight capable data acquisition system and allowed for a flight weight hybrid propulsion system.

Hybrid rocket propulsion systems have been researched since the 1930's. The term "hybrid" denotes a propulsion system that utilizes a liquid oxidizer and a solid fuel. This system is useful because it requires less plumbing than that of a bi-propellant liquid system and unlike solids, allows for throttling and shut-down. Hybrids can also be safer than solid propulsion systems. Hybrid systems can withstand small imperfections in the fuel grains, whereas in solids, small imperfections can cause catastrophic failures. After the Challenger accident, a hybrid strap-on booster was designed as a possible replacement for the solid boosters which caused the loss of the Shuttle and its crew. Another notable use of hybrid technology was the X-Prize winning SpaceShipOne.

WARRIORS III set out to design and construct a lab-scale hybrid rocket motor. This motor was designed to use nitrous oxide as the oxidizer and paraffin wax as the fuel. One of the goals of the project was to determine how different injector designs changed performance characteristics in small hybrid motors. Prior to testing motors, support equipment was designed and constructed. First a test stand capable of handling all loads present during testing was manufactured. Data acquisition was performed by two pressure transducers and a load cell. These sensors were connected to a National Instruments DAQ system which used a LabVIEW program to collect and store the data. By the end of the project, LabVIEW was also programmed to control the operation of the motor based on user inputs.

To test the support equipment, a small 5 lb PVC/ N₂O motor was designed and tested. Using this motor, all sensors and operation equipment were deemed ready for larger motors. After the initial testing of the support equipment was completed, testing of a 10 lb paraffin/ N₂O motor was conducted. During these tests, problems with ignition and flame holding were encountered. During this series of testing it was determined that the cause of these problems was excessive vaporization time for the nitrous oxide.

This meant that the nitrous oxide was not completely vaporized in the combustion chamber. To solve this problem, a four port injector was designed with the goal of facilitating complete vaporization in the combustion chamber. While this did not solve the problem, it resulted in the most successful test. This test was 1.7 second burn that produced close to 10 lbs of thrust.

It was determined that the vaporization delay was still the cause of the problem. A 25 lb motor was designed that utilized a longer fuel grain length. While this motor was not manufactured due to time constraints, it was believed that with an increase in fuel grain length, the problems with ignition and flame holding would no longer be present.

The term “flutter” refers to the vibration of a structure, in this case the fins or rocket body, when subject to an aerodynamic load. There are three simple modes of fin flutter that may be observed on a rocket: stall flutter, pitch-bending flutter, and fin bending-torsion flutter. For this project the bending-torsion flutter was investigated as it was determined that this type was most likely to occur on rocket fins.

The goals of the aerodynamic study were as follows:

- 1: To produce and record data from fin flutter on scaled fins using a wind tunnel and data acquisition system.
- 2: To compare data received from wind tunnel testing with projected figures from mathematical models.
- 3: To design and manufacture a flight capable data acquisition system that could collect and record strain gage data on model rocket fins.

These goals were satisfied in two ways: performing wind-tunnel testing, and designing and manufacturing a flight capable data acquisition system. This data acquisition system was designed to fly in the final flight vehicle.

Using a wind tunnel, different geometries of 0.25 inch plywood fins and 0.063 inch G-10 fiberglass fins were tested in order to produce visible fin flutter. The wind tunnel was operated until the fin appeared to be vibrating. At this point the data was recorded. For the plywood fins this meant running the wind tunnel near its maximum speed, for the G-10 fins the wind tunnel speed was operating at approximately half the maximum speed of the wind tunnel. In each case data was recorded, during testing of the plywood fins the amplifier in the Wheatstone bridge did not function properly. This problem was not identified until the team began testing the G-10 fiberglass fins. A new amplifier was obtained and the data for the G-10 fiberglass fins was recorded. This data proved that the wind tunnel tests and the FinSim model produced comparable result for the flutter frequency. Data from the G-10 fins was imported into MATLAB and graphs showing the Fourier transform of the signal were produced for each

set of test data. These graphs had peaks correlating to the natural frequency of the fins close to those predicted by the FinSim model. Additionally, screen shots of the LabVIEW program after it was run were taken. This program also performed a Fourier transform on the data with results that were consistent with the FinSim model. This empirical testing validated the reliability of FinSim for use in model rockets.

Finally the FCDAQ which was designed to be placed on the final rocket combines both analog and digital circuitry while using strain gages to measure the horizontal fin flutter of amateur rockets. Results were not obtained on this final rocket because of time constraints. Overall the success in the aerodynamics study was experimental validation of results produced by FinSim.

Table of Contents

Authorship Page.....	i
Acknowledgements.....	iii
Executive Summary.....	iv
Table of Contents.....	vii
List of Tables.....	xiv
1 Introduction.....	1
2 Background.....	2
2.1 Aerodynamics.....	2
2.1.1 Fin Flutter.....	2
2.1.2 Mathematical Modeling Equations.....	3
2.1.2.1 Case 1 (Flexure –Torsion Flutter).....	3
2.1.2.2 Case 2 (Flexure-Aileron Flutter).....	4
2.1.2.3 Case 3 (Torsion-Aileron Flutter).....	4
2.1.3 Data Acquisition System.....	4
2.1.4 Dynamic Flight Stability.....	6
2.1.4.1 Longitudinal Moment of Inertia.....	7
2.1.4.2 Aerodynamic Corrective Moment Coefficient.....	7
2.1.4.3 Damping Moment Coefficient.....	8
2.1.4.4 Radial Moment of Inertia.....	9
2.1.4.5 Natural Frequency.....	9
2.2 Hybrid Rocket Propulsion.....	9
2.2.1 History.....	10
2.2.2 Benefits of Hybrid Propulsion.....	12
2.2.3 Current Hybrid Rocket Research.....	13
2.2.4 Previous Grain Studies.....	17
2.2.5 Inefficiencies and Problems with Hybrid Propulsion.....	18
2.2.6 Previous Studies.....	19
2.2.7 Current Injector Design Study.....	20
2.2.7.1 Axial Injectors.....	22
2.2.7.2 Radial Injectors.....	23
2.2.7.3 Swirl Injectors.....	23
3 Methodology.....	25
3.1 Initial Test Vehicles.....	25
3.1.1 Vehicle Design and Construction.....	25
3.1.2 Flight Computer.....	25
3.1.3 Vehicle Design Process.....	26
3.1.4 Construction of Vehicle Components.....	27
3.1.5 Vehicle Assembly.....	28
3.2 Aerodynamics.....	29

3.2.1	Test Stand.....	30
3.2.2	Fin Designs	31
3.2.3	Wind Tunnel Testing	34
3.2.4	MATLAB Modeling of Flutter Equations	35
3.2.5	Fin Simulations by FLUENT	36
3.2.6	Data Acquisition System.....	37
3.2.6.1	Overall goal.....	37
3.2.6.2	Shannon Sampling Theorem.....	39
3.2.6.3	Wheatstone Bridges	39
3.2.6.4	LabVIEW Program	41
3.2.6.5	Amplifier Board	45
3.3	Hybrid Rocket Propulsion.....	46
3.3.1	Motor Design Criteria	46
3.3.2	CFD Injector Modeling.....	48
3.3.2.1	Gambit.....	48
3.3.2.2	Modeling in FLUENT.....	50
3.3.2.3	Defining the Solver and Conditions.....	50
3.3.2.4	Material Definition and Solver Controls.....	51
3.3.2.5	Initial Results and Unsteady Modeling	52
3.3.2.6	Checking Model Consistency	52
3.3.3	Motor Hardware Design.....	53
3.3.4	Oxidizer Delivery System.....	61
3.3.5	Test Stand.....	62
3.3.6	Safety Measures	65
3.3.6.1	Fire lab safety compliance	66
3.3.6.2	Containment Shield/Sound Suppression	66
3.3.6.2.1	Size considerations.....	67
3.3.6.2.2	Cost analysis	67
3.3.6.3	Blast Deflector	68
3.3.6.3.1	Design	68
3.3.6.3.2	Manufacture	68
3.3.7	Testing Procedures.....	69
3.3.7.1	Complete Checklist of Operations	69
3.3.7.2	Complete Test Run.....	69
3.3.8	Data Collection	70
3.3.8.1	Sensors	70
3.3.8.2	Calibration.....	70
3.3.8.3	Interface Cable Connections	71

3.3.8.4	Data Acquisition (DAQ) Equipment.....	72
3.3.8.5	LabVIEW code	73
3.4	Final Test Vehicle	75
3.4.1	Mission Criteria.....	75
3.4.2	Design	76
3.4.2.1	Fin-Can	76
3.4.2.2	Electronics Bay	77
3.4.2.3	Main Body	79
3.4.2.4	Fin Geometries.....	80
3.4.3	Data Acquisition System.....	81
3.4.3.1	DAQ Hardware	81
3.4.3.2	FC-DAQ Code	83
4	Results and Analysis	84
4.1	Initial Test Vehicles	84
4.2	Aerodynamics	88
4.2.1	2D CFD Analysis.....	88
4.2.2	Mathematical Modeling of Fin Flutter.....	96
4.2.3	Wind Tunnel Testing	97
4.2.3.1	Testing of Wood Fins.....	97
4.2.3.2	Bar Test.....	101
4.2.3.3	G-10 Fins	103
4.2.4	Data Acquisition System.....	106
4.3	Hybrid Rocket Propulsion.....	110
4.3.1	Testing Results.....	110
4.3.2	CFD Results and Analysis	117
5	Conclusions.....	120
5.1	Initial Test Vehicle.....	120
5.2	Aerodynamics	121
5.3	Hybrid Rocket Propulsion.....	121
5.3.1	Ignition Delay and Chamber Design.....	121
5.3.2	Injector Flow Modeling	122
5.3.3	Motor Operation.....	122
5.4	Final Test Vehicle	123
6	Resources	124
7	Appendices.....	126
7.1	Appendix A: LabVIEW Setup for Fin Flutter.....	126
7.2	Appendix B: Pin out of 18F87J50 PIC microprocessor	128
7.3	Appendix C: Test Procedure Checklist	129
7.4	Appendix D: Fire Lab Safety Forms.....	131

7.5	Appendix E: FCDAQ Wiring Schematics.	135
7.6	Appendix F: FCDAQ Software Coding.....	137
7.7	Appendix G : Nitrious Oxide Flow Diagram.....	168
7.8	Appendix H : Manual Controler	169

List of Figures

Figure 1: Crystal oscillator [3]	5
Figure 2: Preliminary Schematic of Data acquisition system [4].....	6
Figure 3: Axis of Longitudinal Moment of Inertia about Alpha [6]	7
Figure 4: Hybrid Motor Schematic [9]	10
Figure 5: Schematic of NASA Hybrid [10]	14
Figure 6: Test firing of NASA Hybrid Rocket at NASA AMES in California [10].....	14
Figure 7: Design of Da Vinci Flight Vehicle [13]	17
Figure 8: Star Grain [14].....	17
Figure 9: Helically Fluted Grains [15].....	18
Figure 10: Model of a multi-port fuel grain after burning	19
Figure 11: Typical Injector [18].....	21
Figure 12: Boundary Layer Formation [21].....	22
Figure 13: Radial Injector [19].....	23
Figure 14: Two Types of Swirl Injectors [15]	24
Figure 15: RockSim with Rocket Design	26
Figure 16: Initial Flight Vehicle Fin-can CAD Design (Body Diameter = 3 in)	27
Figure 17: Flight Computer Enclosure.....	28
Figure 18: Assembled Test Rockets Rocket Height: Left -65.3 in., Right – 79.2 in.	29
Figure 19 Fin mounting device for wind tunnel testing	31
Figure 20: Clipped Delta Fin Design	32
Figure 21: Square Fin Design	32
Figure 22: Tapered Swept Fin Design	33
Figure 24: Trapezoid Fin Design	33
Figure 23: Swept Fin Design	34
Figure 25: Sample 1-axis Strain Gauge [5].....	38
Figure 26: Wheatstone Bridge [25].....	40
Figure 27: Quarter-Bridge Circuit [25].....	41
Figure 28: Half Bridge Circuit [25]	41
Figure 29: Full Bridge Circuit [25].....	41
Figure 30: Sample VI Visualizer	43
Figure 31: Oscilloscope, with recording function VI.....	43
Figure 32: Final Program Interface.....	44
Figure 33: LabVIEW code for FFT, and Oscilloscope	44
Figure 34: LabVIEW Code for Writing to a File	45
Figure 35: Single Port Injector Mesh.....	49
Figure 36: Four Port Injector Mesh.....	49
Figure 37: Mass Balance (Single Port)	52
Figure 38: Mass Imbalance (Four Port)	52
Figure 39: Velocity Contours of Single Port Injector	53
Figure 40: Velocity Contours of Four Port Injector.....	53
Figure 41: Cut away view of the 5 lb boilerplate motors with PVC fuel grain	58
Figure 42: Nozzle Holder and Injector after machining. Three inch OD Retention rings shown with each for scale.....	59
Figure 43: Nozzle holder with combustion chamber attached and nozzle insert.....	59
Figure 44: Injector assembly.....	60
Figure 45: 50 N motor assembly	60
Figure 46: Initial flow testing of the four port injector and CAD model showing four port arrangements	61
Figure 47: Cut Section of PVC Fuel Grain 3.5” in length	62

Figure 48: Final Test stand	63
Figure 49: Internal rails used: Internal rails used.....	64
Figure 50: Interim Motor Test Stand	64
Figure 51: CAD model of Test Stand and Support Equipment	66
Figure 52: CAD Design of the Blast Deflector.....	68
Figure 53: DAQ I/O Box	72
Figure 54: DAQ on connection plate, with connector box cable and digital controller	73
Figure 55: Hybrid DAQ 9.0 Front Panel.....	74
Figure 56: CAD Design of Fin-can for Final Flight Vehicle	77
Figure 57: Electronics Bay Sled Design	78
Figure 58: Assembled Electronics Bay Sled.....	79
Figure 59: CAD Model of the Final Flight Vehicle (Dimensions in Inches).....	80
Figure 60: PIC 18f452 Pin Layout [28]	82
Figure 61: Altitude for Initial Flight Vehicle "wicked awesome" flight 1.....	86
Figure 62: Altitude for Initial Flight Vehicle "wicked awesome" flight 2.....	86
Figure 63: Acceleration Graph of "wicked awesome" flight 1.....	87
Figure 64: Acceleration Graph of "wicked awesome" flight 2.....	88
Figure 65: Static Pressure LE.....	89
Figure 66: Static Pressure LETE.....	90
Figure 67: Dynamic Pressure LE	91
Figure 68: Dynamic Pressure LETE	91
Figure 69: Stream Function LE.....	92
Figure 70: Stream Function LETE.....	93
Figure 71: Velocity Magnitude LE	94
Figure 72: Velocity Magnitude LETE	94
Figure 73: Vorticity Magnitude LE	95
Figure 74: Vorticity Magnitude LETE.....	96
Figure 75: Stress analysis results displaying stresses on clipped delta rocket fin.....	98
Figure 76: FFT of the Clipped Delta fin with Horizontal Strain Gauges.....	100
Figure 77: Clipped Delta Fin with Vertical Strain Gauges	101
Figure 78: Aluminum bar test Fourier transform results with horizontal strain gauge placement.	102
Figure 79: Aluminum bar test Fourier transform results with vertical strain gauge placement.....	102
Figure 80: Fourier transform of G-10 fiberglass fin, tapered swept geometry at 25Hz.....	103
Figure 81: G-10 fiberglass fins Fourier transform results for the tapered swept fin with the wind tunnel at 25Hz.....	104
Figure 82: G-10 fiberglass square fin with vertical strain gauge orientation ran at 31Hz	105
Figure 83: G-10 Fiberglass; Clipped Delta fin geometry Fourier transform results. Wind tunnel speed of 31 Hz.....	106
Figure 84: Component Layout	108
Figure 85: Hyperterminal Download	109
Figure 86: Test 8, 12N PVC Grain, 5s burn.....	111
Figure 87: Test 11, 12N PVC Grain, 1s burn.....	112
Figure 88: Test 12, 12N PVC grain, 8s burn	113
Figure 89: Test 13K, 50N Paraffin motor, 3s burn	114
Figure 91: Exhaust Plume From 50N Paraffin Motor with 4 Port Injector, Test 14b.....	115
Figure 90: Test 14B, 50N Paraffin, 3s burn.....	114
Figure 92: Just after ignition of test 15 c, which failed to continue to burn	116
Figure 93: Test 15G, 50N Paraffin motor, 3s burn	116
Figure 94: Contours of Vorticity for Single Port Injector.....	118
Figure 95: Contours of Vorticity for four Port Injector	118
Figure 96: Turbulent Intensity in Single Port Injector	119

Figure 97: Turbulent Intensity in 4-port Injector	120
Figure 98: Sample VI Visualizer	126
Figure 99 : Oscilloscope, with recording function VI.....	127
Figure 100: Hybrid Motor Manual Controller	169
Figure 101: Hybrid Motor Controller wiring diagram.....	169

List of Tables

Table 1: Average impulse of a motor with a total impulse	47
Table 2: Propellant Properties	54
Table 3: Flight Summary Information for First Flight of “Wicked Awesome”	85
Table 4: Flight Summary Information for Second Flight of “Wicked Awesome”	85
Table 5: Square Fin, Test 1	99
Table 6: Square Fin, Test 2	99
Table 7: Swept Fin, Test 1	99
Table 8: Swept Fin, Test 2	99
Table 9: Clipped Delta Fin, Test 1	99
Table 10: Clipped Delta Fin, Test2	99
Table 11: Tapered Swept Fin, Test 1	100
Table 12: Tapered Swept Fin, Test 2	100
Table 13: Trapezoid Fin, Test 1	100
Table 14: Trapezoid Fin, Test 2	100
Table 15: Fin Geometry and corresponding frequency results from FinSim and wind tunnel testing.	106
Table 16: Hybrid Motor Testing overview	110

1 Introduction

The Warriors III major qualifying project (MQP) was an ambitious model rocketry project undertaken by eight Worcester Polytechnic Institute (WPI) students. This project group designed, built, and tested hybrid rocket motors, multiple data acquisitions systems and several high power rockets. The project was completed in three terms at WPI with a budget of \$1200. The goals were as follows:

Aerodynamic Fin Flutter goals:

- Develop a “simple” model to characterize fin flutter
- Use computer models to generate estimates of flutter characteristics (frequency and amplitude) over a range of flight conditions and fin designs (materials and geometries).
- Use model results to design fins for a final flight vehicle (material, geometry, and placement) to provide stable flight and a maximized measurable signal.
- Design and build FCDAQ compatible with payload capability for the flight vehicle.

Hybrid Propulsion System Goals:

- Design and build prototype N₂O/Paraffin hybrid rocket motor which meets following requirements;
 - Maximum impulse of 640 N•sec
 - Size and mounting compatible with flight vehicle
 - Incorporate redundant safety features where possible
- Test prototype HRM in laboratory to collect data on chamber pressure and temperature to compare with analytical model

Final Flight Vehicle:

- Design and build a flight vehicle which meets following requirements;
 - Capable of flying an operational powered FCDAQ
 - Motor installation compatible with either standard solid motor or HRM developed for project
 - Allows for interchangeable test fin(s)
 - Capable of attaining minimum altitude of 1500 feet.
- Maximum installed impulse will not exceed 640 N•sec

Because of the various goals, the team split into two groups. One group concentrated on the hybrid rocket motor aspect of the project, the second group concentrated on fin flutter. The group designed and built all model rockets together.

2 Background

The WARRIORS III MQP was broken down into several subsections: Aerodynamics, a Flight Compatible Data Acquisition System (FCDAQ), Hybrid Propulsion System, and a Flight Vehicle. In the following sections, background information for each of these topics, with the exception of the Flight Vehicle, will be provided. The focus of the aerodynamics section is on using MATLAB to model equations used to generate estimates of fin flutter; therefore, the background will detail information on the equations used and past research that relates to this fin flutter study. The FCDAQ will be part of the payload on the final flight vehicle; the information in the background section includes basic background information on Data Acquisition Systems (DAQs), previous work done with DAQs and fin flutter studies. The Hybrid Propulsion section focuses on designing and building a prototype hybrid rocket motor, and this section will provide background information about hybrid propulsion: past research, firing hybrid motors, and current research.

2.1 Aerodynamics

This section presents background information about fin flutter, modeling equations, dynamic flight stability, and data acquisition systems. The term “flutter” refers to the vibration of a structure, in this case the fins or rocket body, when subject to an aerodynamic load.

2.1.1 Fin Flutter

There are three simple modes of fin flutter that may be observed on a rocket: stall flutter, pitch-bending flutter, and fin bending-torsion flutter. The first, stall flutter, transpires at a high angle of attack. This is unlikely to be observed in the testing of fins for Warriors III because there will be no angle of attack. Thus, this type of flutter was not investigated, instead the focus was on the other two more common types: pitch-bending and fin bending-torsion. Pitch-bending flutter is an oscillation of the rocket body interacting with bending fins. If this oscillation proves to be un-damped this may result in the loss of the rocket’s fins. Pitch-bending flutter is most likely to occur in high moment of inertia rockets with rectangular fins where the ratio of the distance of the center of gravity (measured from the fins leading edge) over the chord length is high. The most likely type of flutter would be fin bending-torsion flutter. This happens when two natural modes of a fin’s oscillations interact through another shared resonance, for instance, fin bending or twisting [1].

All mechanical structures, including rockets, have a natural frequency of oscillation. However, it is unlikely for these structures to oscillate or vibrate at their exact natural frequency. There are two

degrees of freedom characterized by a natural frequency, bending and torsion (twisting). If a fin does not have the ability to bend and twist simultaneously, then any oscillations in either degree of freedom will be dampened. However, if the fin bends, it is possible for the energy repelled by the impact to cause the fin to twist. The energy from this effect is deflected back into the bending of the fin. This fluctuation may continue and amplify if the fin is un-damped, which may cause the fin to fail.

Turbulence in airflow is the cause of flutter; therefore, fins on a model rocket that are designed with a material that is too thin, might flex. Fins that are improperly sized for the rocket or for the material of fin, meaning too thick or too thin, too big or too small, also contribute to fin flutter. This is usually the case when the semi span of the fin is too large.

Fin flutter on a rocket can generate lift, thereby causing the rocket to become unstable. This instability has the potential to be dangerous and must be taken into serious consideration. In the most extreme of cases, fin structural failure can lead to loss of the vehicle. In a less severe situation, the rocket will have minimal flutter but will lose significant altitude and velocity. This also is likely to throw the rocket off of its expected path and should still be considered dangerous.

In order to effectively investigate and analyze fin flutter, the phenomenon needed to be evaluated quantitatively so that any findings could be easily traced back to design variables.

2.1.2 Mathematical Modeling Equations

Several modeling methods were used to test the maximum velocity in which the rocket can travel before fin flutter occurs. It is possible to use these methods to calculate the frequency at which the fin flutters. In order to generate an original model of fin flutter, a study of a National Advisory Committee for Aeronautics (N.A.C.A.) report 685 related to the study of fin flutter was investigated.

Theordorsen and Garrick derived the following expression for the velocity at which an aileron will flutter. In order to use these equations for a rocket fin consider the chord length as the average of the tip chord and root chord.

2.1.2.1 Case 1 (Flexure –Torsion Flutter)

$$v = \frac{r_{\alpha} \omega_{\alpha} b}{\sqrt{\kappa}} \frac{1}{k} \frac{1}{X} \quad (1)$$

Where the radius of gyration of a wing aileron, $r_{\alpha} = \sqrt{(I_{\alpha}/(mb^2))}$, m is the mass of the wing per unit span length I_{α} is the moment of inertia of the wing aileron about the elastic axis (the mid-chord to the center of gravity), $\kappa = (\pi\rho b^2)/m$. $\omega_{\alpha} = \sqrt{(C_{\alpha}/I_{\alpha})}$ is the natural frequency of torsional vibration in a vacuum, and C_{α} represents the torsional stiffness of a wing per unit span length. The reduced wave

length is defined by k^{-1} , and X is a fixed value of this length that is plotted until a continuous curve represents the two variables. The variable represents the half chord of the airfoil [2].

2.1.2.2 Case 2 (Flexure-Aileron Flutter)

$$v = \frac{\omega_k b}{\sqrt{\kappa}} \frac{1}{k} \frac{1}{\sqrt{X}} \quad (2)$$

In this expression for flutter velocity, or critical speed to which the fin-can travel before flutter occurs, ω_k represents the natural frequency of the wing in deflection. It can be calculated by the equation, $\omega_k = \sqrt{(C_k/m)}$ where C_k is the stiffness of the wing in deflection per unit span length. The other parameters are defined above in Case 1 (Flexure –Torsion Flutter)[2].

2.1.2.3 Case 3 (Torsion-Aileron Flutter)

$$v = -\frac{r_\alpha \omega_\alpha b}{\sqrt{\kappa}} \frac{1}{k} \frac{1}{\sqrt{X}} \quad (3)$$

All of the variables represented in this equation are defined in Case 1 (Flexure –Torsion Flutter) [2].

Theodorsen and Garrick also note that it is assumed that the pressure on the airfoil was about two and a half pounds per square foot per chord length. The stiffness-axis location is located 30% of the chord length from the leading edge and the center of gravity location is 40% of the chord from the leading edge. The aileron length is considered to be one fifth of the total chord. Torsional frequency is four times that of the bending frequency, and the aileron frequency is 1.22 times that of the bending frequency. These assumptions are necessary when doing additional work on this subject and understanding their conclusions.

2.1.3 *Data Acquisition System*

A data acquisition system (DAQ) is a system of measuring and recording data. This set up consists of both hardware and software. The hardware is the physical electronics that create the sensors, amplifiers, interface, logic controllers and onboard memory. The software is the logic based instruction set that will convert the raw data into a digital signal which will then store the data into the physical memory for later analysis.

The overall design of a data acquisition system consists of some basic components. The first major component is a microprocessor, otherwise known as an embedded controller. This acts as the central nervous system of the data acquisition system, to control where the information goes and at what time. The next major component is the crystal oscillator that is required to maintain a steady timing signal so that everything can happen at the correct time. The memory chip will store all the collected information for the sensors, and later read this back to a CPU for analysis. The signal conditioning hardware, consisting of operational amplifiers and the resistors that control their gain will allow a usable signal to be gathered from the strain gauge sensors.

The final processor will be similar to an 18F series PIC microprocessor. An example of a timing agent is seen in Figure 1. The crystal will act as the primary timer for the board. Its base frequency will be able to be divided into standard units of time.



Figure 1: Crystal oscillator [3]
© 2008 Kyocera Corporation

This will become the base frequency for the processor; which is a vital component to guarantee all operations happen at the right time. The remaining items are a power source such as a 9-volt battery, and the supporting electronics that make the processor function, such as resistors and capacitors. The connectors that allow the system to communicate with the CPU to download the final data must also be included. Figure 2 is a DAQ that was designed by Steve Hagemen will be used as a preliminary example to base the flight capable DAQ. The information regarding the design and use of this system can be found on Steve Hagemen's website.

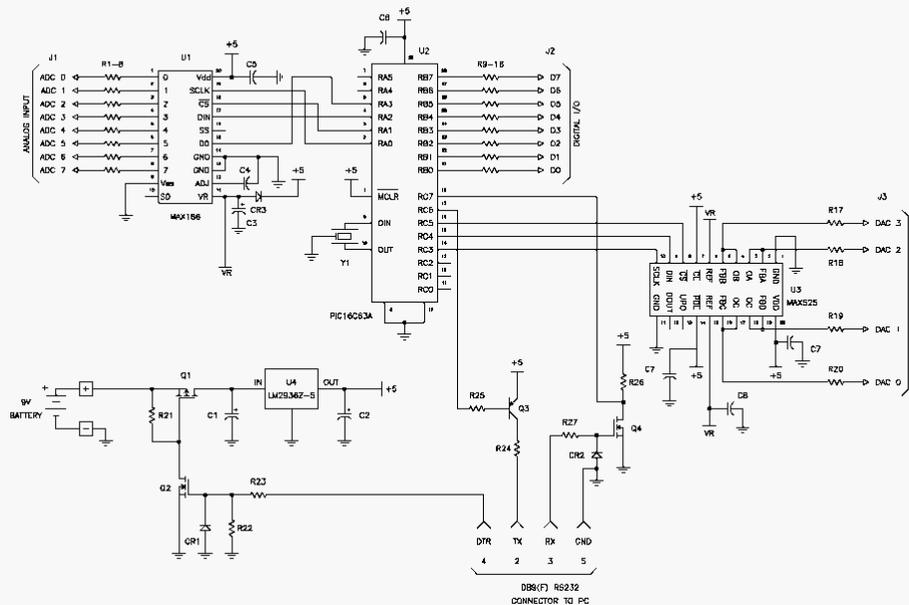


Figure 2: Preliminary Schematic of Data acquisition system [4]
 ©2000 Steve Hagman

2.1.4 Dynamic Flight Stability

The goal in designing a stable rocket is to create an object that when launched will not stray from its intended flight path. If this rocket does veer off course due to any outside forces, such as the wind, it will return to its desired path and compensate for this disturbance. This principal can be broken down into two parts, static stability and dynamic stability. Static stability is the initial response of the rocket to outside forces. Dynamic stability is how the rocket acts over a period of time to these disturbances. In order to compute the static stability, divide the tube diameter by the distance between the rocket's center of pressure (C_p), and its center of gravity (C_g). If this distance is between one and two the rocket is stable, if it is greater than two the rocket is over-stable, meaning that the rocket has a great tendency to weathercock. Over time the flight path will eventually turn into the wind instead of maintaining a straight flight path. In dynamic stability, there are five critically important parameters. These are of concern because safety is the most important factor, and a non-dynamically stable flight corresponds to very irregular and erratic flight which is anything but safe. The five major parameters that control dynamic stability are:

I_L = Longitudinal Moment of Inertia

C_1 = Corrective Moment Coefficient

C_2 = Damping Moment Coefficient

I_R = Radial Moment of Inertia

ω_n = Natural Frequency [5]

2.1.4.1 Longitudinal Moment of Inertia

There are three axes of rotation in a rocket: pitch, yaw and roll. The pitching moment can be neglected based on the direction of measurement. Any force that makes the rocket yaw and pitch about this axis is important to the rocket's overall stability.

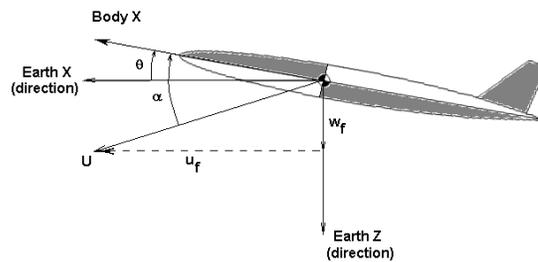


Figure 3: Axis of Longitudinal Moment of Inertia About the Center of Gravity [6]
©1987 McGraw-Hill Book Company

The longitudinal moment of inertia is proportional to the mass of the rocket and that mass is distributed. A rocket that is heavier at the front has a higher longitudinal moment of inertia than a rocket where the weight is concentrated in the middle [5]. The higher this proportion, the more difficult it is to have the rocket rotate around its center of gravity. The rocket will constantly have a different center of gravity due to fuel burning, this value constantly changes throughout the flight, until engine burnout.

2.1.4.2 Aerodynamic Corrective Moment Coefficient

The aerodynamic corrective moment coefficient is based on aerodynamic performance characteristics and used to determine how responsive a rocket is to sudden disturbances that occur in flight. This is important in determining to what extent a rocket will lead into the wind on a windy day and its ability to fly straight. The larger this coefficient, the more responsive the rocket will be to sudden disturbances. As this coefficient is affected by aerodynamic forces interacting with the rocket, it is important to note that the fins have the most significant impact on this parameter.

$$C_1 = \frac{\rho}{2} \cdot V_2 \cdot A_r \cdot C_{na} (Z - W) \quad (4)$$

This can be seen in the above equation by defining some of the variables. C_1 is the aerodynamic corrective moment coefficient; this is result of how sensitive the rocket will be to disturbances. The variable ρ is the density of air and the given altitude V is the velocity of the rocket at that moment in time. The constant A_r is known as the reference area and is usually just the maximum frontal area of the rockets body tube. The coefficient C_{na} is the normal force, and finally the difference between the center of gravity (Z) and the center of pressure (W) is given as (Z-W) [3].

2.1.4.3 Damping Moment Coefficient

$$C_{zr} = \dot{m}(L_{ne} - w)^2 \quad (5)$$

The variable definitions, are as follows, C_{zr} is known as the propulsive damping moment coefficient. The variable \dot{m} is the rate of mass expulsion from the nozzle given in grams per second, L_{ne} is the distance the nozzle is from the tip of the nose cone. Finally, W is the distance from the C_G to the tip of the nose cone [7].

This equation shows how fast the rocket will dampen any oscillations. The higher this coefficient the faster the rocket will return to an equilibrium point. Note that the distance term is squared; this illustrates the dependence the damping moment coefficient has on the length of the rocket. The further the nozzle is placed from the center of gravity the higher the value, and the quicker it will be able to disperse oscillations.

As with the other variables, anything that is a function of the center of gravity cannot be a static number. As the rocket burns its fuel, this value will change. It is also important to note that the engine does play the most important factor in determining this coefficient, but the other components, such as payload, nose cones, and fins along with their location relative to the rocket's center of gravity, will also play a role.

$$C_{2A} = \frac{\rho \cdot v \cdot A_r}{2} \sum (c_{na_component} (Z_{component} - W)^2) \quad (6)$$

C_{2A} corresponds to the aerodynamic damping moment coefficient. The variable ρ is the density of air at the rockets altitude, V is the velocity of the rocket. The constant A_r is the reference area; this is usually just the area of a circle at the maximum diameter of the rocket. The coefficient $c_{na_component}$ is the normal force of the individual components such as the nose cone, body tube, and fins. The constant

$Z_{component}$ is the distance from the nose tip to the CP of the specific component and W is the distance from the nose tip to the CG of the rocket.

This equation represents the rocket's ability to correct itself due to the contribution of a normal force, and the forces induced by aerodynamic drag on the overall structure of the rocket.

2.1.4.4 Radial Moment of Inertia

The radial moment of inertia is almost exactly the same as the longitudinal moment of inertia, the only difference is that it is for the pitching moment about the horizontal axis of the rocket. This axis is known as the pitching axis and was conveniently neglected in determining the longitudinal moment. This moment is the measure of a model rocket's ability to correct any spinning that may occur in flight. The greater this number the greater chance the rocket does not spin in flight.

2.1.4.5 Natural Frequency

$$\omega_n = \sqrt{\frac{C_l}{I_l}} \quad (7)$$

The variable ω_n is the natural frequency of oscillation. The coefficient C_l is the corrective moment which is described in section 2.1.4.3. The variable I_l is the longitudinal moment of inertia [4] which was described in 2.1.4.1.

Typically engines with larger thrusts produce higher flight oscillations. George K. Mandell [5] recommends that the most desirable natural frequencies for a model rocket lie between 0.2V and 1.0V, where "V" is the velocity of the rocket in meters per second. According to his book, when the natural frequency is closer to 0.2V, it means the longitudinal moment of inertia is too great for the corrective moment coefficient to overcome. Therefore, the model cannot respond to the moments applied by its fins with sufficient rapidity for safe and stable flight. The book goes on to say that "a higher value of Natural Frequency means the Longitudinal Moment of Inertia is too low, so the model is easily disturbed, or the corrective moment is too large, usually indicating too-large fins that cause excessive drag." [5]

2.2 *Hybrid Rocket Propulsion*

There are three main propulsion systems used in modern launch vehicle propulsion systems; the liquid rocket, the solid rocket, and the hybrid rocket. In liquid propulsion systems, an oxidizer and fuel

are mixed via injectors that form a combustible liquid mixture. In solid rocket systems, the fuel and oxidizer are mixed into a solid that is ignited when the surface of the solid reaches its ignition temperature. A hybrid system encompasses aspects found in both solid and liquid systems; in most cases the oxidizer is in a liquid state and the fuel is in a solid state. The liquid oxidizer is injected into the solid fuel grain and combustion is usually started by injecting a hypergolic¹ fluid into the combustion chamber [8, 9] (see Figure 4).

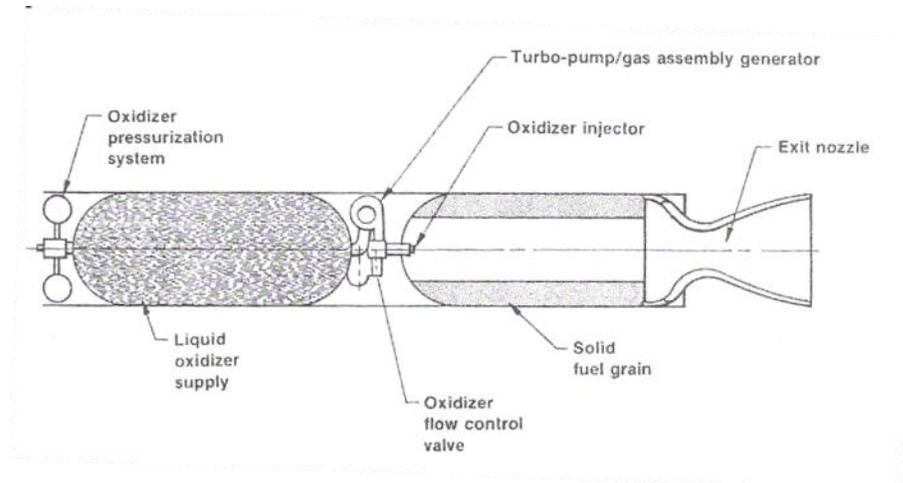


Figure 4: Hybrid Motor Schematic [9]
©2007 American Institute of Aeronautics and Astronautics, Inc.

2.2.1 History

Hybrid rocket development can be traced back to the 1930's; around the same time as many basic developments in solid and liquid rockets were taking place. Solid rocket motors were the first to be studied due to the early development of gunpowder. These solid motors used powder charges to achieve combustion. Robert Goddard, commonly referred to as the father of model rocketry, started out by trying to inject gunpowder into a combustion chamber instead of igniting it all at once. The dangers of igniting large amounts of gunpowder were brought to light when, in 1933, Reinhold Tiling and three assistants died when a 40lb powder charge exploded. Due to the complexity of injecting small amounts of gunpowder into the combustion chamber, Goddard soon shifted his research to liquid rockets [9].

The first recorded effort of a hybrid motor was designed by Sergei Korolev and Mikhail K. Tikhonravov. They were working on the Russian GIRD program; GIRD-09 was successfully flown on

¹ Hypergolic propellants are a fuel and an oxidizer that ignite when they come in contact with each other. Ignition occurs without an external aid.

August 17, 1933, to an altitude of 1500m. The motor utilized a solid fuel comprised of gelled gasoline with Liquid Oxygen (LOX) as an oxidizer [9]. Although this was not a true hybrid by today's standards (because the fuel was gelled, not solid) it marked the beginning of research of hybrid propulsion.

In 1937, serious research in the hybrid field began. The first major tests were performed at I.G. Farben (a group of German chemical companies) by Leonid Andrussow, O. Lutz, and W. Noeggerath [9]. Their first tests involved a 10 kN motor utilizing a coal fuel and Nitrous Oxide (NOS) as an oxidizer. These tests ultimately failed due to carbon's high heat of sublimation, which resulted in a negligible burning rate. The low burning rate produced very poor performance characteristics.

The first research for a flight worthy hybrid motor was not conducted until the mid 1940's. This research was carried out by the Pacific Rocket Society (PRS), which is a private sector consortium, who used LOX and a wide variety of fuels. Some of these fuels included; wood, wax injected with carbon black, and many types of rubber based fuels. The PRS started by making nozzles out of Douglas fir soaked in zinc and ammonium chlorides; this design was quickly dropped because the nozzle would disintegrate after only 15 seconds of burning. On their next attempt they once again used Douglas fir nozzle, this time coated in calcium sulfate. This proved more successful than the first tests; however, no

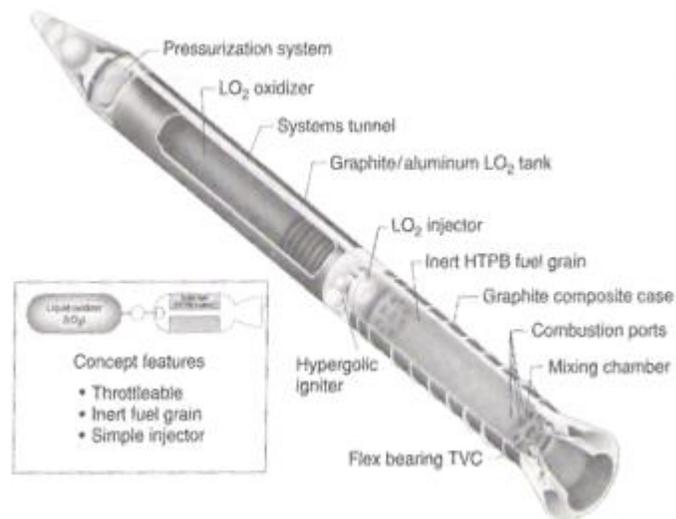


Figure 5: Hybrid Strap-On Booster Design for the Space Shuttle [8]
©2001 John Wiley and Sons.

relevant data could be collected, as the motor flew off its static test stand (rated for 40-60 lbs of thrust). After many tests, they finally designed a motor that used LOX, a rubber based fuel, and an aluminum alloy nozzle. This motor was successfully launched in June 1951 to an altitude of 30,000 ft [9].

Since these early tests, organizations such as the National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory, General Electric, and Thiokol have done research on hybrid

technology. In the 1970's, hybrid technology was once again a major research topic in the rocket industry. During these years hybrid developments were focused on target oriented missiles and tactical missile applications. High power upper stage motors were another area of significant research on hybrid motors. After the *Challenger* disaster in 1986, a hybrid strap-on booster was designed (see Figure 5) as a potential replacement for the solid rocket boosters that caused that particular Space Shuttle to explode a mere 73 seconds into flight. Several high thrust hybrid motors have since been developed; the two most recent being a 250,000 lbf motor tested by the American Rocket Company from 1993-1995 [8] and 60,000 lbf motor tested by the Lockheed Martin Corporation [9]. These large hybrids use a LOX oxidizer and hydroxyl terminated polybutadiene (HTPB) as a fuel. This combination is preferred because it provides a less toxic and virtually smoke-free exhaust. During these tests, it was also found that this combination almost produced performance equivalent to a LOX-kerosene liquid motor [8].

2.2.2 Benefits of Hybrid Propulsion

Hybrid propulsion is worth investigating for several reasons. As a blend of both solid and liquid propulsion systems, hybrids share in some of the advantages of each system. In addition, hybrid propulsion has many of its own unique benefits.

Some unique benefits of a hybrid system are related to safety, due to the dual propellant nature of hybrid propulsion. Many types of fuels used in hybrid motors are not significantly reactive or easily ignitable when separated, and thus are safer to handle before firing. This is a significant difference from solid propellants and some liquid propellants, which must always be considered combustible. In addition, because many fuel combinations may not ignite when separate, there is very little risk of explosion or damage during storage or manufacture; this can be a significant safety hazard in liquid and especially solid rockets.

There are several advantages to hybrid rockets over other conventional propulsion systems. Compared to bipropellant rockets, for example, a hybrid with a single liquid propellant would only need half the related feed system of a liquid-fuelled rocket with two liquid propellants. This also means a significant boost in reliability – with fewer rotating parts, there is a decreased chance for problems. The use of a solid fuel section compared to liquid rockets also increases the average fuel density, meaning that the same amount of thrust can be delivered in a smaller vehicle [9].

Compared to solid boosters, hybrids have many benefits as well. In terms of simple performance characteristics, the lighter exhaust gases increases specific impulse. Additionally, by regulating oxidizer flow rates, the hybrid motor can be throttled or shut off entirely and restarted. On the contrary, solid

motors will continue to burn until all the fuel is consumed leaving no method of throttling or shut-down. Hybrid fuel grains are typically safer than solid fuel grains; cracks in solid grains can be devastating due to over pressurization in a small area from burning, which can lead to motor explosion or motor failure. Most hybrids, however, cannot burn without an oxidizer, and as such most cracks are harmless unless they threaten the structural integrity of the motor tube [9].

Finally, these rockets can utilize many common items as fuels; the testing for this MQP utilizes polyvinylchloride (PVC) or paraffin wax as a solid fuel. These two fuels are both very common and inexpensive chemicals; in addition, many of the other fuels that can be used for larger scale testing are common inert chemicals. This means that per launch, hybrids can be significantly cheaper than solid rocket motors. All these benefits combined with their relative simplicity in operation and benefits over other chemical propulsion systems, makes hybrid propulsion a worthy topic of investigation.

2.2.3 Current Hybrid Rocket Research

Some of the most important avenues of research in rocketry today, in both model and large scale applications, are finding new forms of abundant fuel that are environmentally friendly. Many researchers have turned to hybrid rocket propulsion in order to try and fill this void. NASA is currently looking into development of a large scale hybrid motor to power some of its new launch vehicles. A hybrid-paraffin motor is appealing to NASA because it is both environmentally friendly and largely dependent upon resources that are easily found in our environment.

The by-products of combustion from typical hybrid propellants are carbon dioxide and water; unlike conventional rocket fuel that produces aluminum oxide and acidic gasses, such as hydrogen chloride. [10]

This means that there would be far less danger to people on the ground from the exhaust produced by the main engines of a launch vehicle.

Engineers at the NASA Ames research center in California have conducted feasibility studies and tests to see if large scale paraffin motors are both possible and easier to make and use than the traditional boosters rockets used on the Space Shuttle. These tests have thoroughly concluded that not only are large scale paraffin motors attainable but that they show performance characteristic trends similar to the smaller lab scale motors which they are based upon. [10] An actual full scale paraffin motor being tested by NASA can be seen in Figure 6 and Figure 7.

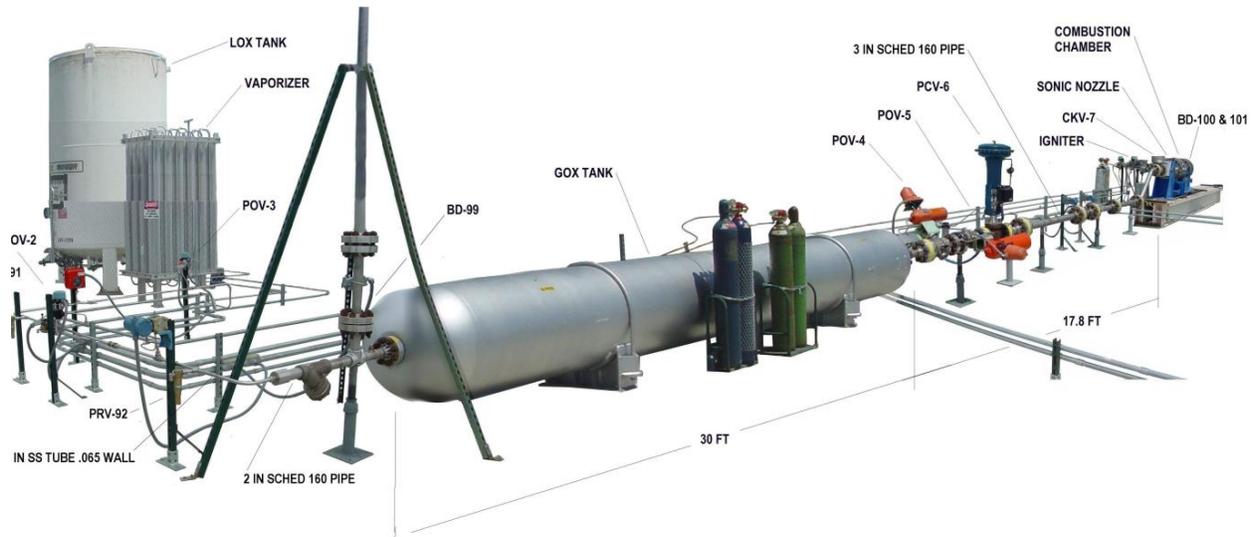


Figure 6: Schematic of NASA Hybrid [10]

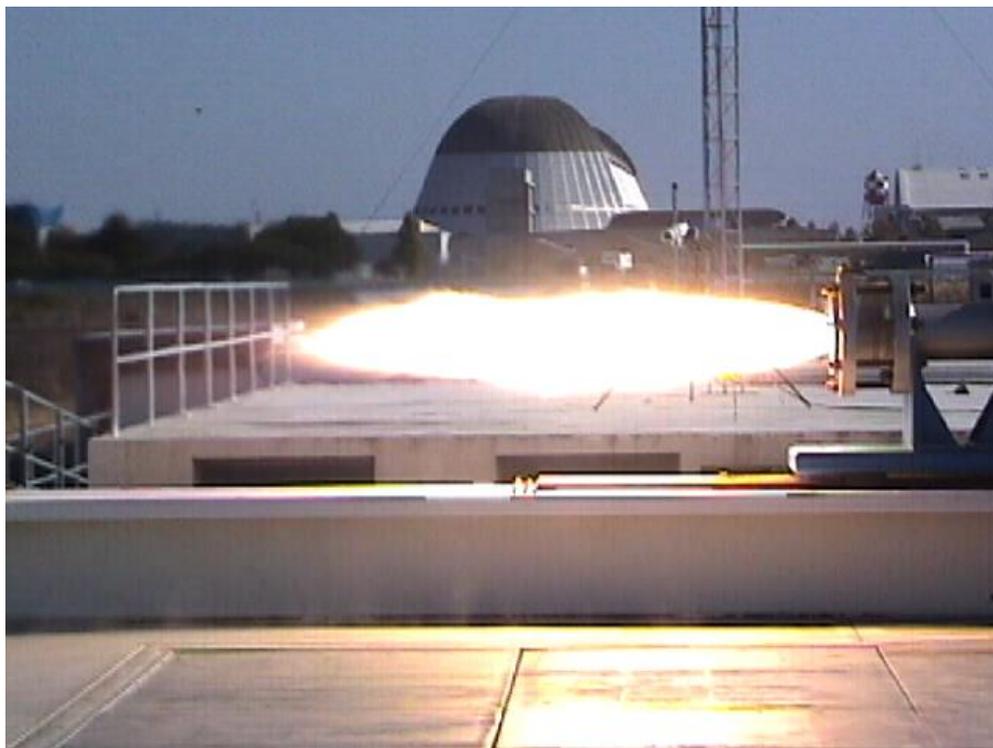


Figure 7: Test firing of NASA Hybrid Rocket at NASA AMES in California [10]

Besides government institutions like NASA, there are a large number of academic institutions that are pursuing their own research into hybrid propulsion methods. Here are just a few examples of research that has been or is currently being conducted in the United States.

Work done at the University of New Orleans includes mathematical modeling of regression rate ablating hybrid solid fuels and flame propagation in hybrid motors. [8] There have been papers published at the University of Tennessee on inviscid modeling of classical and vortex injection in hybrid rocket motors. [8] Testing at Arizona State has been done in areas such as the swirl injector design and its effects on increasing regression rate and combustion stability. [8]

The Aeronautics and Astronautics Department at Stanford University has conducted research on high regression rate fuels by way of liquid entrainment, or more simply the absorption of solid particles into a liquid flow. This is significant because the principal fuel grain used in their study was paraffin based. They conducted three individual flights at Black Rock in Nevada with a commercially available hybrid engine using both a cellulose fuel that came with the motor and a paraffin fuel. They also used Nitrous-oxide as the oxidizer, which is what we will be attempting. They actually demonstrated that the paraffin fueled motor could propel a rocket higher than the cellulose fueled motor. [8]

International Universities have also been closely following the development of hybrid motor design. In the Netherlands at the TNO Prins Maurits Laboratory they have developed system analysis models of complete hybrid systems with optimized grain designs with the associated liquid components. [8] At the Indian Institute of Technology a group has conducted a systematic experimental investigation on ways to enhance the regression rate in GOX/HTPB hybrid rocket motors. They discovered that adding Ammonium Perchlorate (AP) or aluminum in the fuel along with the reduction of grain port diameter enhanced the regression rate. [8]

The commercial market has not ignored the presence of this new hybrid technology, in fact it has already helped propel one company on a \$10 million dollar endeavor. The X Prize was created in 1996 as a joint venture between several different companies which challenged contestants to build and fly a vehicle capable of taking a crew of two up to an altitude of 100 km. This task had to be completed twice within a window of two weeks. Eventually the contest was won by Scaled Composites led by Burt Rutan. This event was significant in hybrid rocket propulsion because the craft was propelled by a hybrid rocket motor using nitrous oxide and HTPB fuel. [8] The motor that propelled SpaceShipOne was developed by the Environmental Aeroscience Corporation (eAc).

eAc was incorporated in 1994 and specializes in the development, fabrication, and testing of innovative Nitrous Oxide and Liquid Oxygen hybrid rocket motors for sounding rockets, tactical missiles, small orbital launch vehicles, and space tourism applications. [11]

Another hybrid project that eAc is currently in development of is the MuLV launcher system. “MuLV is designed to be the world's smallest orbital launch vehicle, capable of placing 350-1200 lb payloads into low earth orbit.” [11] The MuLV will incorporate a 30,000 lb thrust hybrid motor with a easily transportable launch platform and minimal ground crew to open a new low cost alternative to many prospective companies.

Since the X prize was won, new contests have been developed which include the new X Prize Cup, an annual event which challenges rocket enthusiasts to enter different space oriented contests from creating a lunar-lander to a multistage launch vehicle. A Successor to The X Prize, America's Space Prize offers \$50 million to the team which can accomplish certain tasks such as; taking a crew of at least five people to a minimum of 400 kilometers and maintain a sufficient velocity to complete two full orbits. [12]

One entry that will be competing for this prize will be the Da Vinci Space Project which is taking an all new approach to this problem. This vehicle will be lifted to an altitude of 80,000 feet with reusable helium balloon and then detach and ignite the engines on the vehicle. The vehicle itself is fairly simply designed as a cylinder with a spherical tip. This project is interesting, because the engine system which was selected to propel this new vehicle is a Nitrous-oxide hybrid motor system, very similar to the one described in this report, just on a much larger scale. Instead of paraffin as the fuel grain, the Da Vinci engine uses HTPB, “The engine is in the 68,181 N thrust class (15,000 lbs st) with a total impulse of 4,545,454 N·s (1,000,000 lb·sec).” [13] The basic idea is that once a sufficient height has been reached the engine block will separate from the rest of the vehicle and descend back to earth under its own automatic guidance system. The reusability of the vehicle (including the engine) is paramount for the company's success, so a parachute as well as landing airbags will deploy in order to soften the landing of the engine block. Even though all of the previously stated vehicles are competing to win various prizes the final goal of each company is to break into the commercial market by offering low-cost transportation to zero-gravity space.

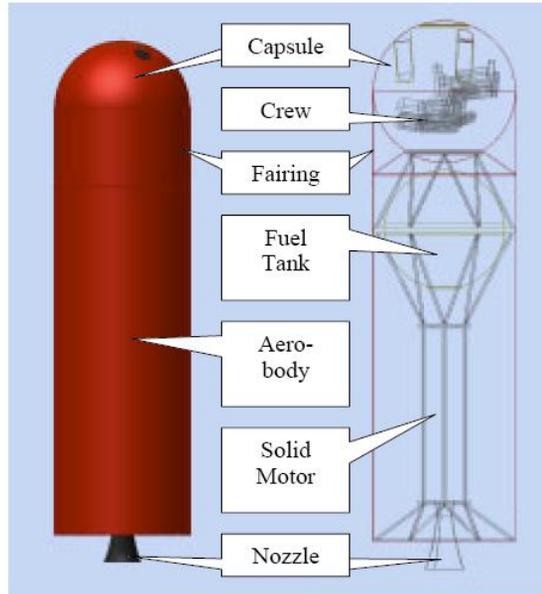


Figure 8: Design of Da Vinci Flight Vehicle [13]
 © 2003 Environmental Aerospace Corporation

2.2.4 Previous Grain Studies

When designing an injector one must also take into account the fuel grain configuration within the motor. One classic example is the star configuration, which allows for more initial surface area during the burn when compared to a Bates grain allowing for a higher level of thrust. A star grain configuration can be seen in Figure 9:



Figure 9: Star Grain [14]
 ©2003 ERG Aerospace Corp

When using a swirl injector as described above to create a vortex flow inside the combustion chamber it is desirable to have a fuel grain which also helps to facilitate this flow pattern. One such pattern is known as a fluted grain. Here spirals are cut into a grain in order to create a rotation of the

oxidizer as it passes through the combustion chamber. When combined with a swirl injector, a significant increase in thrust can be observed. Some examples of these grains are shown in Figure 10:

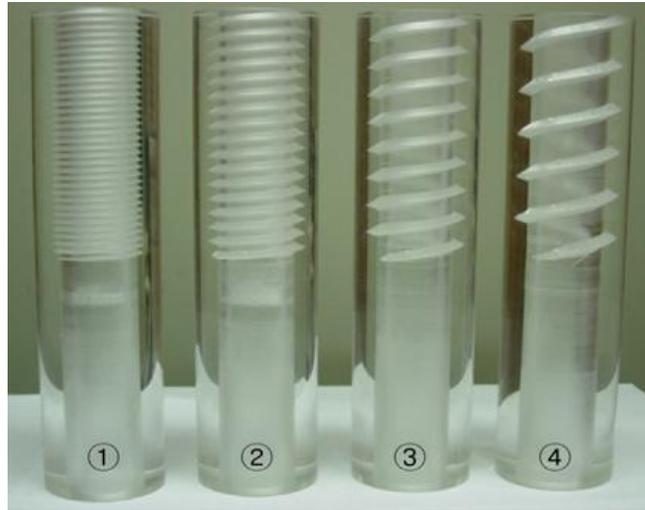


Figure 10: Helically Fluted Grains [15]
© 2006 Elsevier Masson SAS

2.2.5 Inefficiencies and Problems with Hybrid Propulsion

Hybrid motors present a high level of safety and low cost of operation, however low performance and thrust levels often keep hybrid motors from being the propulsion choice for flight vehicles. Until recently, the regression rates, the speed at which the fuel can burn from the surface and expose un-burnt fuel, of typical hybrids were nearly three times slower than that of ammonium perchlorate composite propellants. Due to this slow burning rate, hybrid motors needed complex fuel grain configurations to operate at similar thrust levels. [9]

A previous study was completed by the American Rocket Corporation's (AMROC) predecessor in the 1980's to demonstrate the possibility for hybrids to replace solids in the space shuttle. In this study, a high thrust motor was developed, which forced designers to configure the grain geometry to have a large surface area. Of the high area grain designs, the only feasible configuration was found to be a multi-port grain.

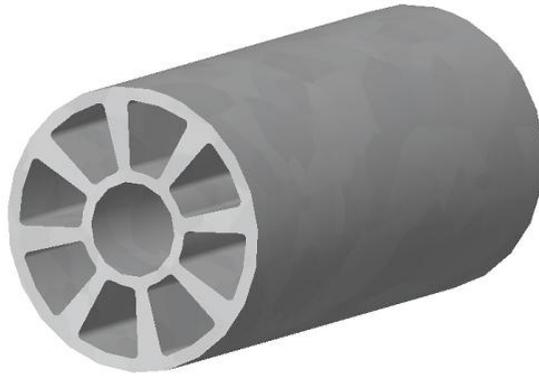


Figure 11: Model of a multi-port fuel grain after burning

This is extremely inefficient due to the amount of un-burnt fuel at the end of the burn, where these un-burnt fuel slivers were ejected from the motor at the end of the burn. [8, p19]

Unlike liquids and solids the control of the oxidizer-to-fuel ratio (OF), can only be controlled by the flow rate of the oxidizer as the fuel grain surface area is purely a function of time and burn rate. In order to operate at high specific impulse, the motor must be constantly throttled to maintain a constant OF ratio or a grain must be designed to have a constant surface area as it burns which often forces a lower than optimal grain geometry.

The challenges with maintaining OF ratio are also coupled with combustion instabilities that are often associated with hybrid motors that operate with combustion pressures near the delivery pressures.[9, p599 -601] This is frequently seen in amateur motors where the small scale coupled with a delivery pressure that decreases dramatically during operation. This produces a high amplitude sinusoidal oscillation in the combustion chamber pressure which is further magnified in the thrust. This combustion chamber pressure combined with injector pressure can be avoided by highly turbulent injectors as well as longer combustion chambers. [8, p362]

2.2.6 Previous Studies

The bulk of hybrid testing has been accomplished using HTPB. This is often chosen because so much is known about this synthetic rubber since it is the binder and fuel for most solid ammonium perchlorate composite propellants (APCP) motors. Fuel choices are usually hydrocarbon based, as such, HTPB, Coal, Wood and Plastics were the initial choices for fuels in hybrids [8].

Within the last decade high regression rate propellants have been studied, where different formulations of organic waxes provided high performance. Paraffin wax is the choice of many research institutes due to its high accessibility, low cost, and casting ability. [8] Though the wax fuels have high

regression rates, they also have low melting temperatures, which can lead to mechanical stability issues of the fuel grain itself.

Additives used in the fuels or oxidizers in hybrid rocket motors are extremely common. Lithium and fluorine have been added to liquid oxidizers to increase burning performance and mass density. Fuel additives are more common, and include aluminum, carbon black, as well as thickening agents to raise melting temperatures [8, p113-115].

Metal additives such as aluminum are often added in small percentages, usually not greater than 20%. These additives often support higher combustion temperatures leading to an increase in Isp; however with paraffin wax, the additive can have the reverse effect in hybrid motors compared to APCP. [16]

Additives such as carbon black or oximide are used for inhibiting the burning rate of the motor. Carbon or other opacifiers are used to inhibit radiative heating of the grain past the flame front. Other chemicals can be used in parallel to inhibit the burning rate of the fuel itself. However, this is less common in hybrid motors, because a change in oxidizer flow rate can be made to achieve a similar effect with only a small decrease in specific impulse [8, p413].

Injector designs play an extremely important role in combustion of rocket engines. In liquid motors, the injector is responsible for the proper mixing of oxidizer and fuel; it is also responsible for the spread and combustion formation in the chamber. With hybrid motors, the injector is the only mechanism for spreading the oxidizer evenly over the fuel grain in order to achieve a nearly constant OF ratio.

There are a few different types of injectors that are the concentration of this report: axial, impinging, and swirl injection. Axial injection is most often used in amateur rocket motors, as well as many research institutes as the oxidizer flow characteristics are easy to calculate. This however does not always present an optimized flow as the regression rate in the center of the grain is often higher than at the nozzle end or the injector end [17]. Swirl injectors as well as impinging injectors are designed to distribute the oxidizer evenly across the fuel grain face. This can be done by increasing turbulence in the boundary layer, which gives the oxidizer and fuel more time to mix properly and burn inside the combustion chamber [9, p603].

2.2.7 Current Injector Design Study

Using results from a single axial orifice injector as a control, various injectors were designed to determine if optimization of the regression rate of the fuel grain within the motor was possible. In order to optimize the regression rate of a fuel grain there are several factors that one can take into account. Specifically, this study was conducted with the goal to increase the regression rate while still maintaining

stable combustion within the motor. Classical hybrid regression rates are determined using the following equation:

$$\dot{r} = aG_{ox}^n \quad (8)$$

Where \dot{r} is the regression rate of the surface of the grain, while a and n are predetermined constants specific to the propulsion system configuration, operating pressure, propellants, OF ratio and grain shape. G in this instance is the mass flux of the oxidizer. Judging from this equation only, it may seem possible to maximize the regression rate simply by increasing the mass flux of the oxidizer. Recent studies stipulate that there are more factors to take into account than simply the oxidizer mass flux. Two important aspects that will be discussed are boundary layer disturbance and heat transfer.

The cross section of a typical, simple axial injector is shown in Figure 12.

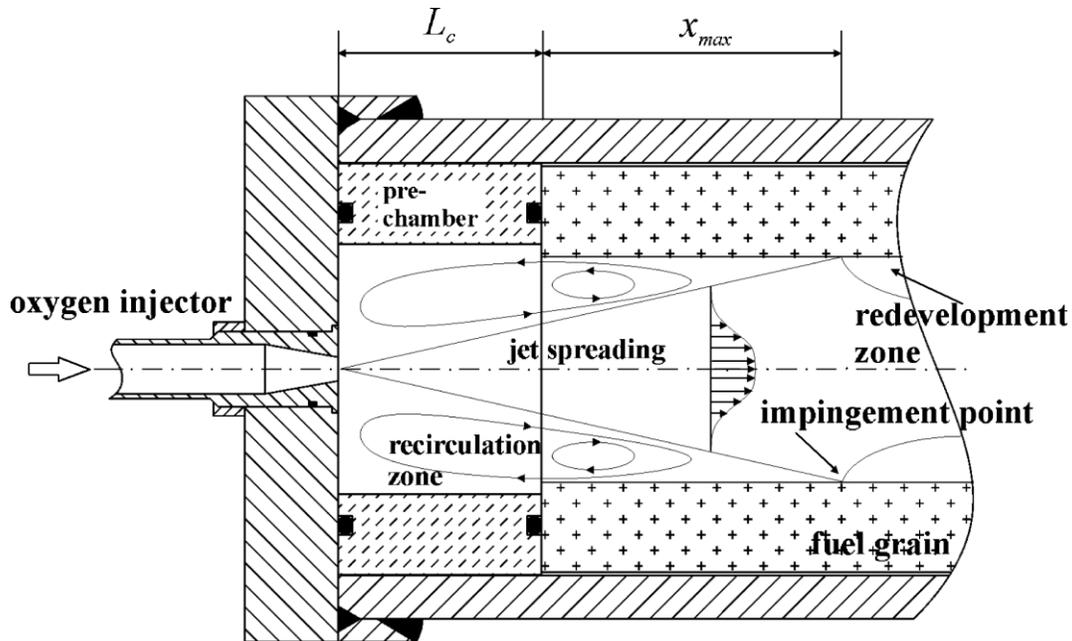


Figure 12: Typical Injector [18]
© 2006 Elsevier Masson SAS

As one can see from this image there are three distinct zones inside a motor with a simple, single-orifice injector. The first is called the recirculation zone, where the oxidizer, which is in contact with the walls, is recirculated back into the main flow zone. This facilitates combustion because the oxidizer becomes saturated with fuel from the grain walls which is then introduced into the pure oxidizer stream so that it can react. The second zone is called the impingement zone where the main stream of oxidizer first comes in contact with the fuel grain wall.

The last and largest section is the zone downstream of the impingement point where a turbulent boundary layer begins to form [18]. This zone is of particular interest to this study because it is the

interaction of this boundary layer with both the surface of the grain and the main oxidizer flow which can determine the regression rate of the surface. As the boundary layer forms there is less contact between the solid fuel in the walls of the grain and the pure oxidizer flow. This boundary layer consists of a high amount of fuel from the grain surface and it would be desirable to introduce this fuel into the main oxidizer flow in order to enhance combustion. This phenomenon is illustrated in Figure 13:

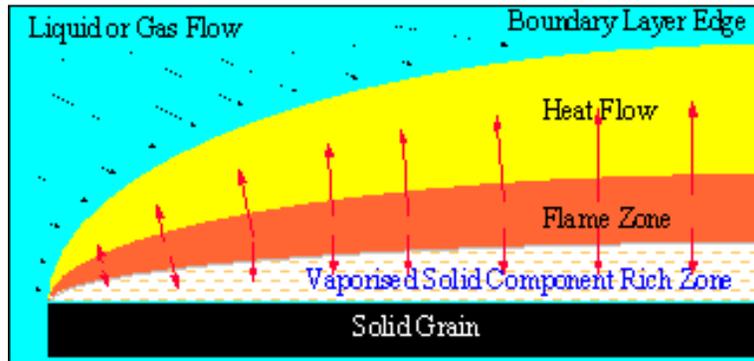


Figure 13: Boundary Layer Formation [21]
© 2002Tsinghua University

The boundary layer is also a non-ideal case for heat transfer from the middle of the tube to the walls, due to a thermal layer which is formed within it. This hinders the melting of the grain, and ultimately slows combustion. Ideally small vortices would exist within the boundary layer. These would help to facilitate heat transfer and proper mixing. It is this aspect of each injector that was studied and evaluated.

The first task set was to decide what types of injector should be modeled. After careful research, three main categories were decided upon: axial injectors, radial injectors, and swirl injectors.

2.2.7.1 Axial Injectors

As seen in Figure 12, axial injectors are the easiest to produce and also have some of the best performance with respect to regression rate. The typical injector consists of a single hole in the center of the injector face. The flow is then passed through this hole under high pressure in order to atomize the oxidizer. Other configurations of this type of injector can include a radial distribution of smaller holes in addition to impinging streams of oxidizer flow which helps to disrupt the boundary layer formation downstream. One ability of the axial injector type is to pass high volumes of oxidizer flow making it ideal for rocket motors which require a large initial impulse with a relatively short burn time. One disadvantage of this injector is the uneven burn of the fuel grain, which resembles a cone after the burn.

2.2.7.2 Radial Injectors

Radial injectors consist of a hole just like axial injectors, however aft of that there is usually a cap which is attached to the injector orifice that directs the flow radially outward to the fuel grain. An illustration of this is seen in Figure 14:

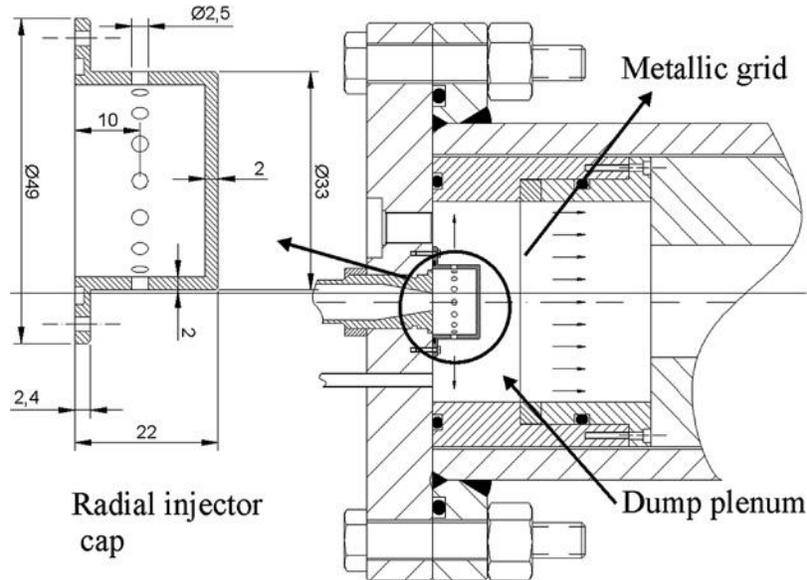


Figure 14: Radial Injector [19]
© 2006 Elsevier Masson SAS

The radial injector does not support as high a flow rate through the motor as the axial injector does; however, it creates a steadier and more even regression of the fuel grain. Normal configurations of radial injectors have somewhere around 16-20 equally spaced holes of a certain diameter. This type of injector is ideal for long sustained burns which require a steady amount of thrust. Radial injectors are more complicated to produce and wide spread use has not yet occurred.[8] One possible application for radial injectors could be for a long duration satellite mission where initial thrust is not as important as a gradual and sustained burn.

2.2.7.3 Swirl Injectors

Swirl Injectors are one of the cutting edge technologies being implemented by hybrid motors to advance performance past solid rocket engines. Swirl injectors are a combination of the radial and axial injectors. A front and section view of two different types of swirl injectors can be seen in Figure 15:

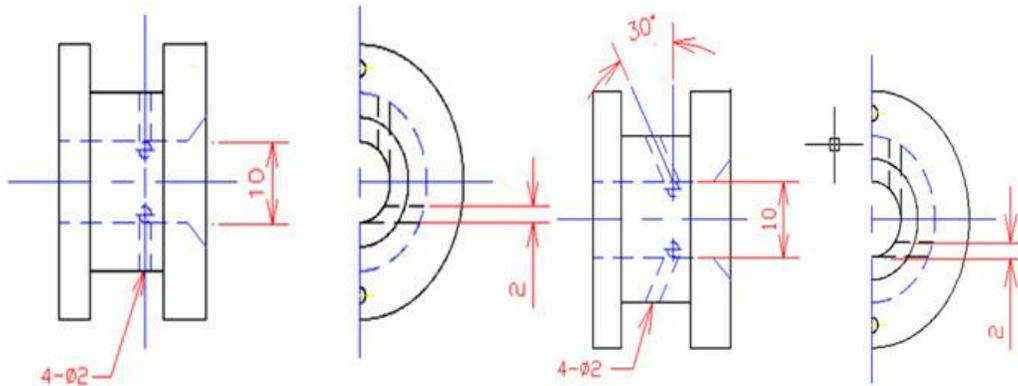


Figure 15: Two Types of Swirl Injectors [15]
© 2006 Elsevier Masson SAS

Each injector each has a single hole at the end just like an axial injector. However in the case of the swirl injectors, each has 4 bores coming out of the main inlet to expel some of the oxidizer in a 90 degree direction as in the first case, or a 60 degree direction as in the second case. This flow pattern greatly helps to disrupt the formation of the boundary layer downstream by inducing a swirl effect inside the chamber. Some tests have reported seeing increases of up to 700 percent in regression rate while using a swirl injector compared to an axial injector with similar flow properties. [15] A rocket using this injector can provide a tremendous amount of thrust with a more even burn of the fuel grain when compared to one using an axial injector. The only disadvantage of using a swirl injector is that they are generally more difficult to produce for the model rocket enthusiast.

Hybrid motors are seldom operated at the stoichiometric OF ratio. This is due to the high temperatures of combustion, as well as the configurations necessary to achieve complete burning inside the combustion chamber. Published data trends of and Isp values show how the change in OF from stoichiometric mixture will only have a small effect on Isp, while also significantly lowering the combustion flame temperature [8]. Often hybrids in the amateur community are run at extremely high OF ratios which operate using N_2O and PVC or HTPB. Small motors often run OF ratios of 10-20:1 with a set oxidizer flow rate. This drives the necessary surface area down dramatically allowing for much shorter fuel grain. However, these motors operate using much of the nitrous oxide as a monopropellant, where the excess nitrous oxide not part of the combustion process decomposes from the surrounding heat. This leads to a decrease in specific impulse, though the specific impulse of N_2O as a monopropellant is roughly 170 seconds^{-1} , slightly less than matched combustion with plastic fuels.

3 Methodology

The overall goal of the WARRIORS III MQP was to design a flight vehicle which can carry an operational Flight Compatible Data Acquisition System (FCDAQ), allows for interchangeable fins, and can attain a minimum altitude of 1500ft, with a maximum installed impulse not exceeding 640 N·sec. In order to design this vehicle, research and testing was conducted on the three areas mentioned in the background (Aerodynamics, FCDAQ and Hybrid Propulsion). The following section explains the work done in each area and how they integrate into the final flight capable vehicle. The aerodynamics section includes information about how modeling was conducted and then used in wind tunnel testing. The FCDAQ section explains design, manufacturing, and testing in order to fulfill the payload requirements of the flight vehicle. The hybrid propulsion system section details how the motor was built and how testing was conducted. Finally, there is a description of the final flight vehicle, and how it was designed.

3.1 Initial Test Vehicles

The purpose of the initial flight vehicles was to provide members of the team with experience in how to design, build and launch a model rocket. The design was completed in RockSim (Apogee Components, Inc., Colorado Springs, CO), then the parts were ordered and over a term the rockets were built. Two different designs, with different fin geometries were built.

3.1.1 Vehicle Design and Construction

With the purpose for the initial flight vehicles as described in the previous section, the next step was to design and construct the rockets. Prior to the design process, the decision to build two separate rockets was made. Another reason for building two rockets was to test the two flight computers. Each rocket however had a similar design and construction process.

3.1.2 Flight Computer

Two flight computers were obtained from the WARRIORS II MQP. These flight computers are both model MC-2 and manufactured by G-Wiz partners. By using these flight computers, the team can receive acceleration, velocity, and altitude data. During the WARRIORS II MQP, the flight computers did not get the opportunity to be flight tested. To ensure that both flight computers could collect data, they were flight tested using the initial test vehicles.

3.1.3 Vehicle Design Process

The initial rocket designs were made using RockSim, a computer program that allows the user to design and run simulations on different rocket designs. The program contains motor data (total impulse, average thrust, burn time, etc.) for a large number of model rocket motor manufacturers. When running a simulation, the user can choose which motor they want to run the simulation with. The simulation will return information on the simulated flight such as; max velocity, max altitude, and max acceleration. Using these simulations the user can determine which motor to use to achieve certain mission parameters. The goals for this project were to achieve an altitude greater than 1,500 ft with a motor impulse limit of less than 640 N-s. With these parameters, the team set out to design a rocket using a pre-determined body tube size, motor mount tube size, and nose cone design.

Both rockets utilized a split body tube design. The two body tubes were connected by a tube coupler. This tube coupler also served as the enclosure for the flight computers. Once a rough length of the rocket body was determined, the fin sets had to be designed. For both rockets, the team selected a 4 fin configuration. One of the benefits of using RockSim was that the program would show the user if the rocket would be stable during flight. Using this feature, the individual fin sets and the rocket lengths were adjusted until RockSim showed that the rocket would be stable during flight. A screenshot of RockSim displaying the design of one of the rockets can be seen in Figure 16.

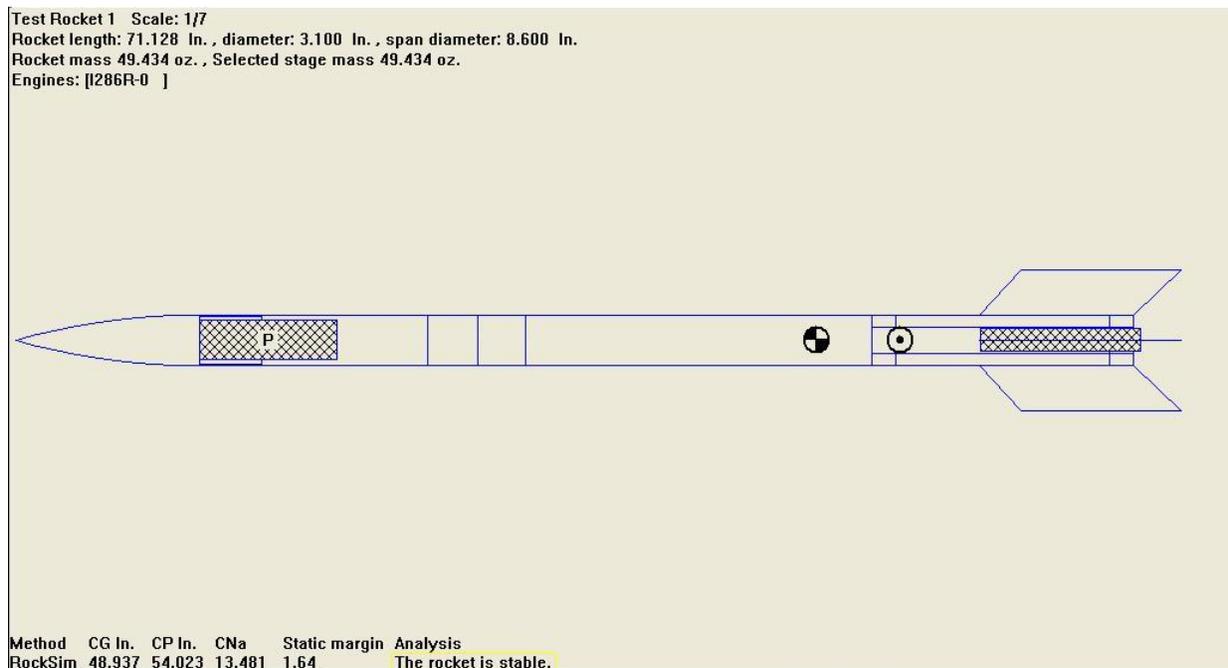


Figure 16: RockSim with Rocket Design

3.1.4 Construction of Vehicle Components

To simplify construction, many of the rocket components were ordered from an online vendor. These parts included the body tubes, motor mount tubes, tube couplers, and the nose cones. The first step of the construction process was to manufacture the centering rings, bulkheads, and fins from a sheet of 0.25 in. thick plywood. This was completed using a CNC milling machine. Once these parts were finished, the fin-can was constructed.

The fin-can is a structure that holds the fins and the motor, attaching them to the rest of the rocket. To build the fin-can, three centering rings were attached to the motor mount tube using high strength epoxy. Next, the fins were mounted to the motor mount tube and the centering rings. The fin-can was then slid into the lower body tube so that the fins fit into four slits that were cut into the body tube. A model of the fin-can can be seen in Figure 17. These slits were then sealed with epoxy. Once the epoxy set, the excess epoxy was sanded off leaving a smooth finish between the body tube and the fins.

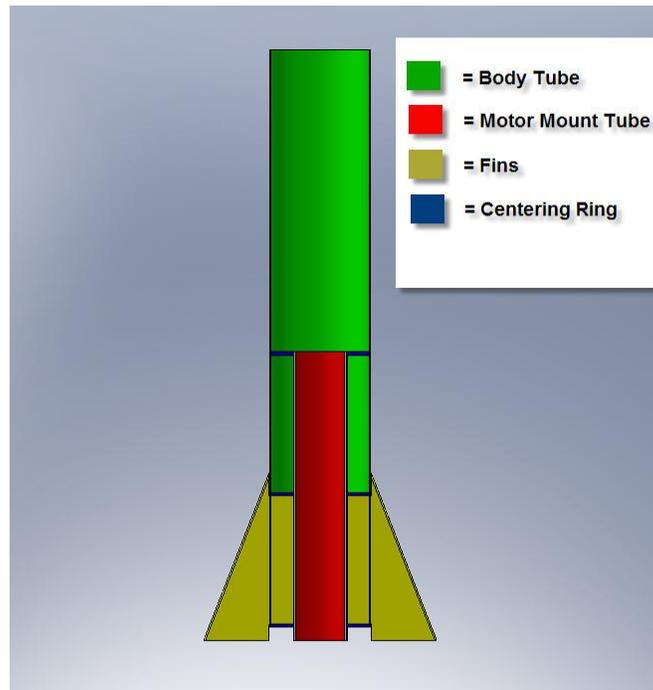


Figure 17: Initial Flight Vehicle Fin-can CAD Design (Body Diameter = 3 in)

The next part of the rocket that was constructed was the flight computer enclosure. This consisted of a plywood sled that would fit inside the coupler. This sled served as a mounting point for the

flight computer. The coupler was sealed on each end with a plywood bulkhead. The bulkheads were held on by a threaded rod that went through the center of the coupler and had bolts on each end. This allowed the enclosure to be opened for access to the flight computer. On each bulkhead an eye hook was bolted in, which were used as mounting points for shock chords that would keep the rocket segments attached after parachute deployment. The completed coupler/ flight computer enclosure can be seen in Figure 18.

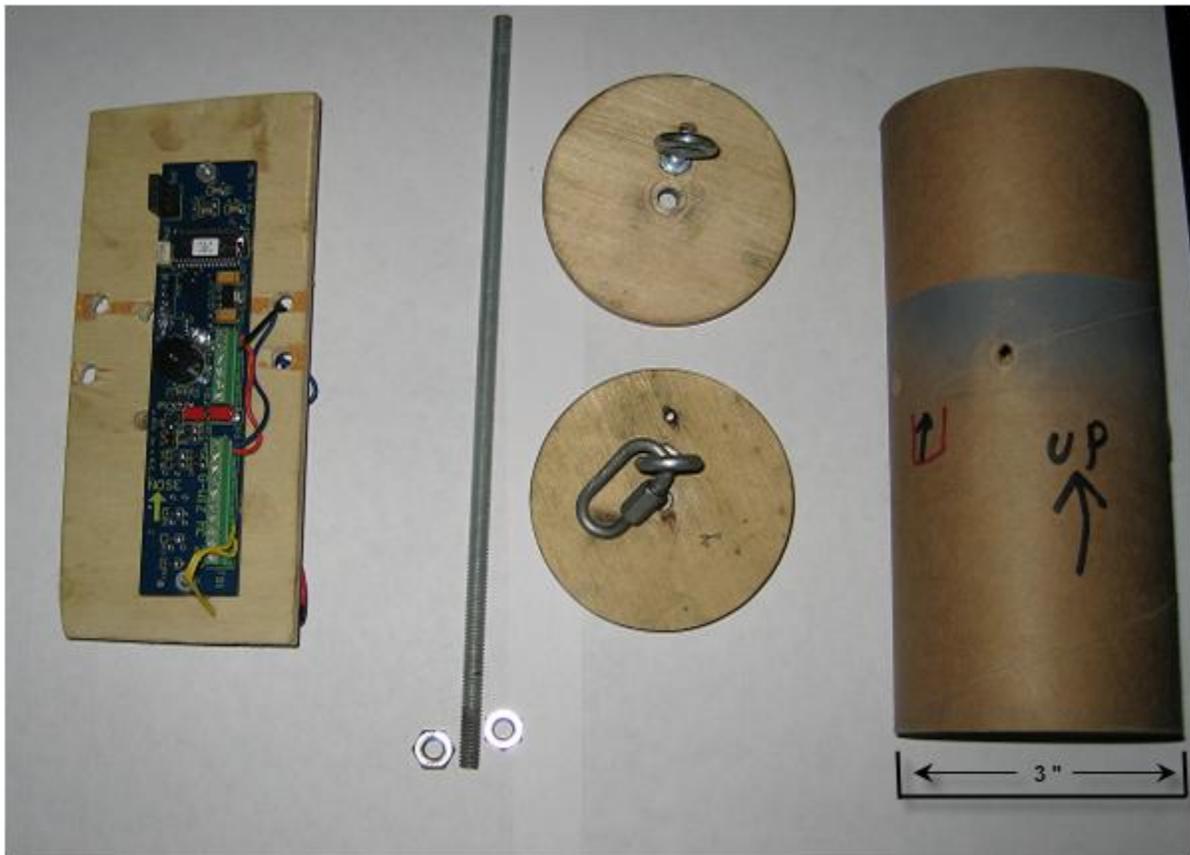


Figure 18: Flight Computer Enclosure

3.1.5 Vehicle Assembly

Once all the rocket components were constructed, the vehicle was assembled. First, the flight computer was properly secured inside its coupler enclosure. The coupler was then slid into the upper body tube and firmly attached with three screws. Next a 0.5 in. diameter nylon shock chord was attached to connect the bottom of the coupler to the fin-can. This shock cord is put in place to ensure that when the rocket begins its descent, its components remain together. Once the shock cord is in place, the bottom end of the coupler is slid into the lower body tube.

Another shock cord was attached to connect the top of the coupler to the nose cone. This shock cord is of great importance because it holds the main parachute. The parachute is attached to the shock cord using a quick link. Once the parachute is attached, it gets neatly folded and inserted into the upper body tube. Finally, the nose cone is placed into the upper body tube. The final assembled rockets can be seen in Figure 19.



Figure 19: Assembled Test Rockets Rocket Height: Left -65.3 in., Right – 79.2 in.

3.2 Aerodynamics

The goal of the aerodynamics research was to develop a simple model to characterize fin flutter that can be used to generate estimates of flutter characteristics over a range of flight conditions and fin designs. The results of the testing were used to determine what designs of fins were used for the flight vehicle to provide for flight stability and maximize measurable signal. In addition to working with fin flutter, the aerodynamics group developed a FCDAQ. This system was designed and built so that it fits

the payload capability of the flight vehicle. The following sections describe the specific steps that were taken to develop the model, manufacture the test equipment, and develop testing procedures. Additionally it details the FCDAQ, how it was designed and manufactured.

3.2.1 Test Stand

A test stand had to be designed and manufactured to conduct wind tunnel testing. The purpose of the test stand was to securely hold the test fins in the wind tunnel during testing. The main design parameters of the test stand were to allow for different angles of attack and have the capability to hold a variety of different fin geometries. Another key design parameter was to ensure that the test stand would not come in contact with any of the wind tunnels surfaces. This was to prevent damage to the wind tunnel in the event of severe fin oscillations during testing. Aluminum was chosen as the construction material because it is a light material that is strong enough to handle the loads that will be present during testing.

The test stand was designed to be mounted to an I-beam tripod that is in the fluids lab, located in Higgins Laboratories, room HL016, at WPI. This I-beam tripod is heavy enough and stable enough so that it will not move during wind tunnel testing. The test stand's base is designed to have four mounting bolts that securely attach the test stand to the I-beam tripod. From this base, a rectangular protrusion enters the wind tunnel through an access panel on the bottom of the wind tunnel. On top of this protrusion sits a circular base. This circular base has four axial slots for which bolts can be inserted through to attach the top plate of the test stand to the bottom plate. The dual plate design will provide the user the ability to rotate the base up to an angle of attack of plus or minus thirty degrees. Mounted on top of the top plate is the fin holder. This holder is designed to securely hold the test fins in place during testing. The fin itself is secured in the fin holder by set screws.

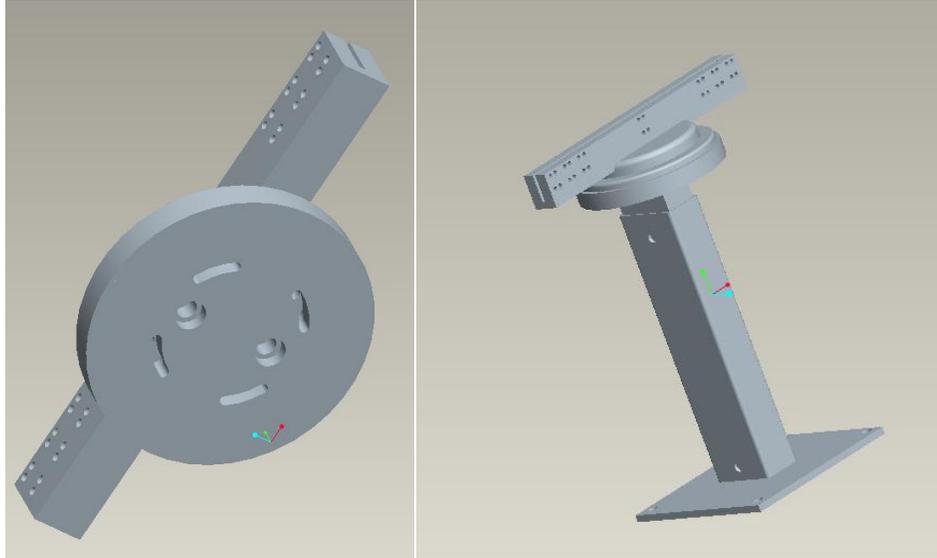


Figure 20 Fin mounting device for wind tunnel testing

3.2.2 *Fin Designs*

Using FinSim to determine flutter frequency and velocity, different fin designs were chosen for testing in the wind tunnel. These designs were chosen because according to FinSim they would flutter at an attainable velocity in the wind tunnel. This allowed recording of flutter data in the LabVIEW program which could then be compared to the initial FinSim calculations. The different fin designs can be seen in Figure 21 to Figure 24.

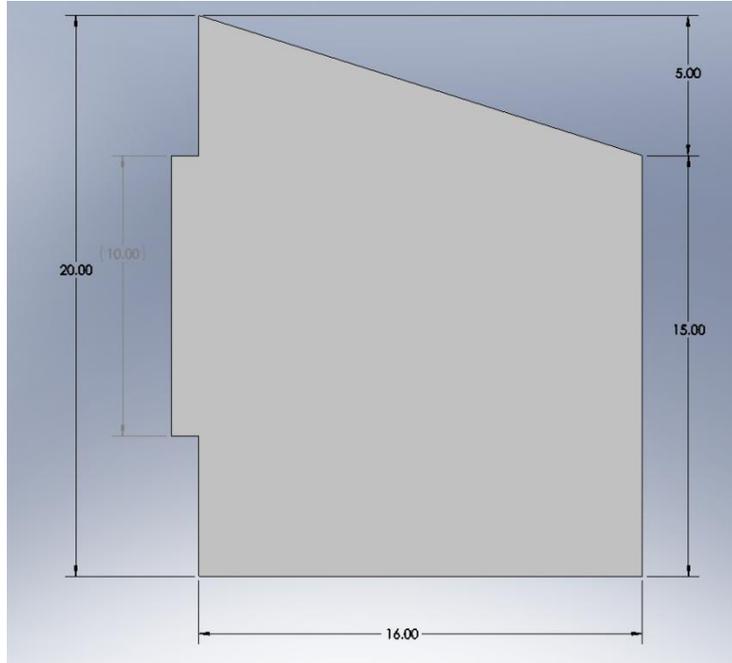


Figure 21: Clipped Delta Fin Design

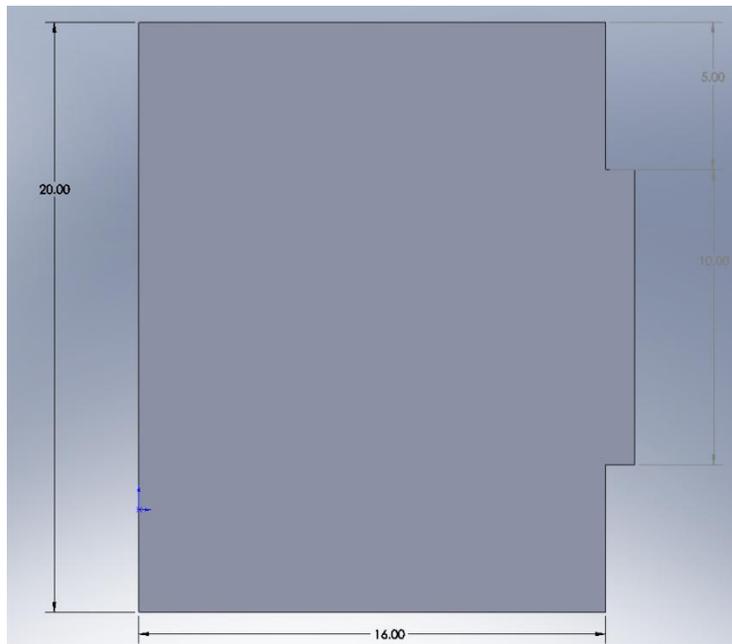


Figure 22: Square Fin Design

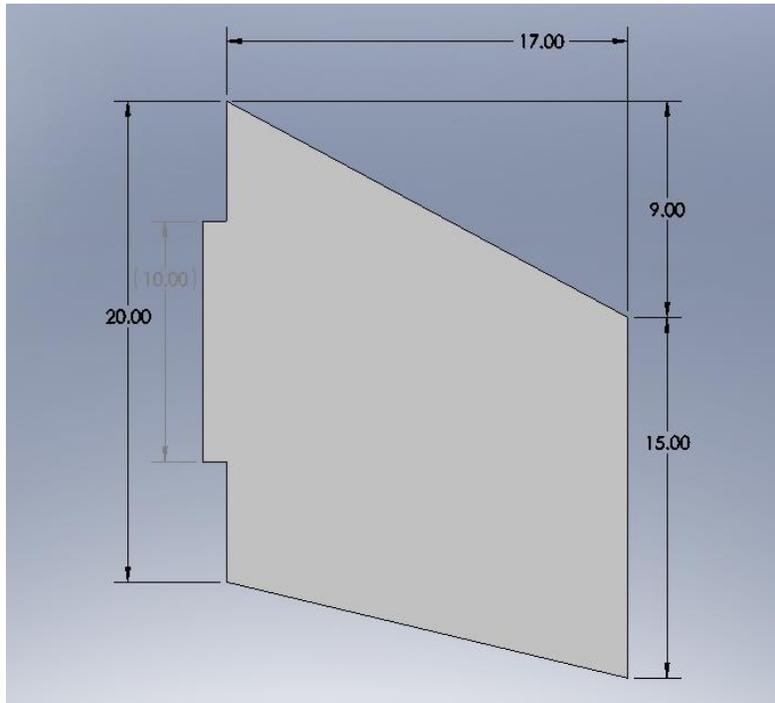


Figure 23: Tapered Swept Fin Design

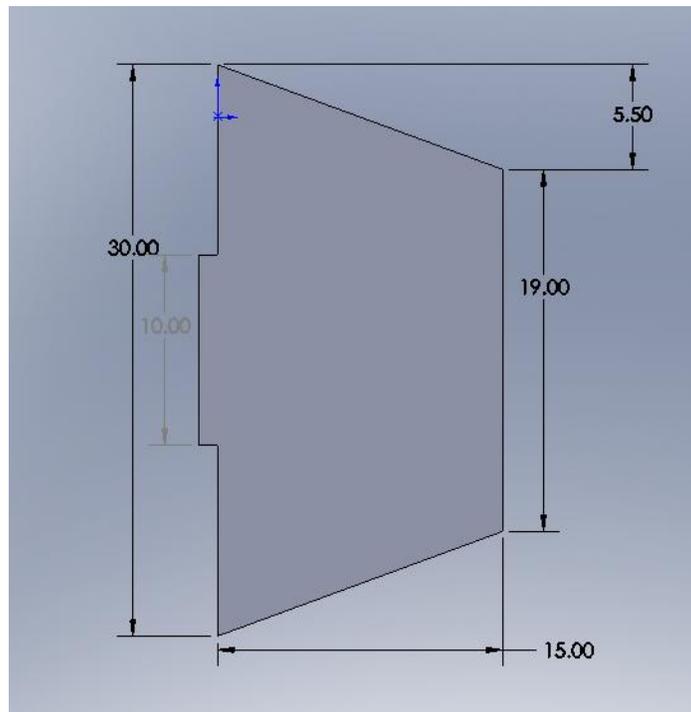


Figure 24: Trapezoid Fin Design

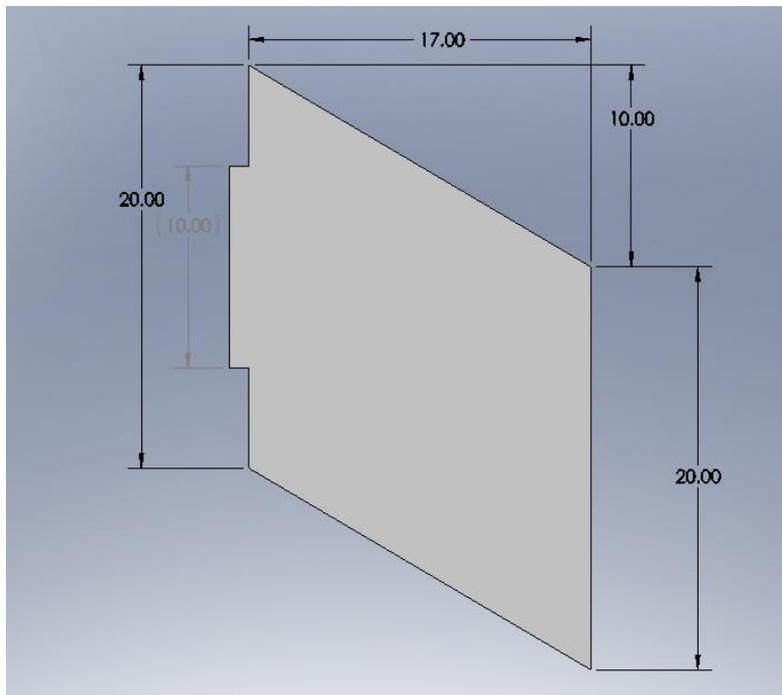


Figure 25: Swept Fin Design

Using the maximum velocity of the wind tunnel and geometries of the fins a Reynolds number was calculated. This number was then used with the maximum velocity of the final flight vehicle to scale the fins to a size that will produce flutter so that data can be recorded from the final flight vehicle as well.

$$Re = \frac{\rho V D}{\mu} \quad (9)$$

V is the velocity that the fin will be subjected to, D represents the characteristic length of the fin, ρ is the density of fluid, and μ corresponds to the fluid viscosity.

The Reynolds number was not calculated to find a specific value instead it was used as a scaling factor between the fins in the wind tunnel versus the fins on the final flight vehicle. Both sets of fins were exposed to similar conditions except for the velocity and therefore, it was believed this was a useful scaling factor. The excel spreadsheet used for calculating the scaling factor can be easily manipulated to change wind tunnel or flight vehicle velocity and then new chord lengths and spans will be found for fins that will produce flutter.

3.2.3 Wind Tunnel Testing

The purpose of the wind tunnel testing was to test FinSim and MATLAB results of fin flutter velocities. The fin geometries were put into FinSim and MATLAB and the flutter velocity was determined to be within the range of the wind tunnel and final flight vehicle velocity. While we were able

to use the models developed to determine theoretically what would be the best design and size, and the actual tests in the wind tunnel provided real data in order to validate the analysis tools (FinSim) used to choose the final fin design.

For the testing, a wind tunnel (Engineering Laboratory Design Inc. Lake City MN, Model 407 A/B 24" Re-circulating) was used. Data was recorded from the attached strain gauges from both the 0.25" plywood fins and the 0.063" G-10 Fiberglass fins. The strain gauges were attached to an amplifier which was then connected to the computer and a LabVIEW program was used to record the flutter. In addition to recording the data, some tests were filmed and pictures were taken to show fin flutter views from testing.

FinSim allowed for the geometries to be determined accurately based the maximum speed of the wind tunnel. At this speed the fins would flutter. The MATLAB program allowed for a graph to be made from the testing data that proved what was seen in the LabVIEW program. Showing the fin flutter in the wind tunnel allowed for the determination of what fin types should be used for the final rocket as it was important to have some fin flutter in order for the on board Data Acquisition System to record data.

3.2.4 MATLAB Modeling of Flutter Equations

The equations used in MATLAB and their purpose are as follows: to create a working mathematical model that calculates the velocity at which the fin will flutter and the natural frequency of each fin geometry. In order to account for the available facilities, a few overall assumption had to be made; no flutter velocity exists for center-of-gravity positions forward of the elastic axis of the fin. The following equations are extracted from N.A.C.A Report 685.

The equation below calculates the maximum vehicle speed before the occurrence of flutter:

$$V_f = V_a \sqrt{\frac{G_e}{\frac{p}{p_0} (\lambda^2 + 1) \frac{39.3A^3}{\left(\frac{t}{c_R}\right)^3 (A + 2)}}} \quad (10)$$

Where V_f is the maximum a speed rocket can reach before flutter occurs. V_a is the of sound, G_e represents the shear modulus of the fin material, p/p_0 is the pressure ratio, which is usually assumed to be one in model rocketry, A symbolizes the aspect ratio, t is the thickness of the fin, c_R is the root chord length.

The fin aspect ratio is calculated using the following equation.

$$A = \frac{S^2}{S \cdot c_T + \frac{c_R - c_T}{2} \cdot S} \quad (11)$$

Where S represents the fin span, and c_T is the tip chord length of the fin.

Output frequency of flutter is then obtained by the following equation,

$$\omega_\alpha = \frac{V_f}{b \sqrt{\frac{2m}{\rho_\infty F_{SA}} \frac{r_\alpha^2}{\delta C_L} (x_\alpha + \frac{e}{b})}} \quad (12)$$

Half of the root chord length is symbolized by b , m is the mass of the fin, F_{SA} signifies the surface area of the fin, $(\delta C_L)/\delta\alpha$ is the fin lift slope, which is assumed to be 2π in two-dimensional flutter. Again this equation r_α symbolizes the fin radius of gyration, x_α is the quantity two times the center of gravity location measured from the fin leading edge divided by chord length minus one minus the quantity of two times the elastic axis of location measured from the fin leading edge divided by chord length minus one, and finally e is the elastic axis of location measured from the fin leading edge divided by chord length minus the aerodynamic center measured from the fin leading edge divided by chord length.

The prevention of fin flutter is important for safety. Using this information from the equations above, it is possible to design a rocket so that it does not exceed the critical velocity at which flutter will occur.

3.2.5 *Fin Simulations by FLUENT*

FLUENT is a program can take a 2D or 3D object and model fluid flow over or through the object. The software code is based on the finite volume method spread over a finite grid. The finite volume method is for representing and evaluating partial differential equations as algebraic expressions. FLUENT has the ability to offer a wide range of physical models with only moderate computations. FLUENT was chosen for this project because it has the capability of modeling static flows over an object as well as being able to provide a dynamic flow field over an object. In order to use computational fluid dynamics (CFD) analysis provided by FLUENT, the object was first modeled in order to see fluid flow over it. In addition to FLUENT software the sister program GAMBIT was also used. GAMBIT is the geometric modeling and grid generation tool used to provide the boundary conditions for FLUENT to allow for CFD analysis. GAMBIT also has the capability of taking geometries from a CAD program and

importing those geometries directly into a mesh. The mesh can then be directly imported into FLUENT for analysis.

2-D flow over the fins was first simulated in FLUENT. The first step in modeling the fins was to determine the geometry to be meshed. To complete this step in GAMBIT, a finite flow field had to be drawn. Next the 2D fin was placed in the center of the flow field. In order to correctly place this geometry in GAMBIT, an overlay grid to place the nodes for the geometry was constructed. These nodes represent geometric intersections. The boundary conditions are set by defining the mesh as a finite flow field and making the geometry of the 2D fin a solid face. After those conditions are set, then the conditions were meshed for the flow field. The higher mesh density will result in an elevated resolution of the FLUENT model. After the meshing was complete, basic boundary conditions in GAMBIT, such as inflow and outflow were defined. The mesh was then imported into FLUENT. Here the more detailed fluid properties were stated and the model was generated. After completing 2D flow analysis, 3D flows were generated.

For 3D flow, a representation of the fin was drawn in Pro/E and then imported and automatically meshed into GAMBIT. This 3D flow field was then imported into FLUENT and CFD analysis was completed based upon the parameters set by the user. This showed the flow fields over the fin to help understand why the fin flutters at a given velocity. This flutter is shown in the FLUENT analysis through disruptions and unbalanced flows in contour plots, velocity flow fields, path-lines and other computational tools provided in the software.

3.2.6 Data Acquisition System

The goal of the DAQ is to collect and display the strain gauge data at an adjustable sampling rate between 1 and 10,000 Hz. LabVIEW will be used to display and analyze the collected data.

3.2.6.1 Overall goal

The Data acquisition system for the WARRIORS III aerodynamics' group consisted of the essential hardware required for all real world test of fin flutter.

There are common components to all data acquisition systems, on both the hardware and the software levels. The first and arguably most vital pieces of the system that have to be selected are the sensors. These determine what is being sampled, how it is being sampled, and what range and frequency can be achieved. For the WARRIORS III project, these were strain gauges (see Wheatstone Bridge

Section 3.2.6.3). An example of a “SGD-1.5/120-LY11 1-axis general purpose” strain gage is shown in Figure 26. A similar gauge was used for this project. The strain gauge function is similar to a potentiometer. Instead of turning a dial to change the resistance the strain gauge is attached to a surface, as the surface deflects, the strain gauge wires change distance increasing or decreasing the resistance.



Figure 26: Sample 1-axis Strain Gauge [5]
©2002-2008 EmAnt Pte Ltd

The testing of this system had two primary phases that which occurred in parallel. The first was the creation of a LabVIEW program that worked in conjunction with a National instruments BNC- 2120 Controller board, and PCI DAQ card. The second phase involved the design and construction of a miniaturized version of this hardware that will record fin flutter data on the final flight vehicle.

There are four phases in the gathering and recording of information with an automated system. It is critical that each of these phases be examined as to achieve the ideal DAQ system for the task. The first phase is determining what has to be measured. This will establish what type of sensors needs to be found, along with what types of signals are to be generated. In this case the fin stresses were measured using strain gages. Since the voltage is extremely low for almost any strain gage, an amplifier was constructed in order to achieve a usable signal.

The next phase was signal conditioning, to do this it is important to have an idea of whether a digital or an analog signal is needed. If this is not determined, proper signal conditioning will not be achieved. This would result in excess noise, fragments of a larger signal, or mistaking a signal for noise. In the case of the strain gauges used, an oscillating signal similar to a sine wave was seen, with peak and trough at ± 2.5 volts after amplification

The third phase was to gather and record the incoming data into an onboard memory chip. This was the most difficult part of creating the data acquisition system, as it is primarily programming. It involved the setup of a communication line between the hardware and a Central Processing Unit (CPU) interface. The most common are Universal Serial Bus (USB) and RS232. If the first three steps were properly completed, the communication selection is a matter of determining the format of the output data.

The last and final stage of the any data acquisition system is the analysis of useful recorded data. This was done using LabVIEW to generate the signal curves and interpret the Fast Fourier transform. These steps when properly followed yielded excellent results.

3.2.6.2 Shannon Sampling Theorem

The Shannon Sampling Theorem is a theory which states that in order to accurately measure a signal, the sampling frequency of that signal must be at least 2 times the expectant signal frequency. This is because any signal will change with time. Digital logic can only measure the signal during a discrete period of time; therefore, the sampling process is performed by taking multiple measurements that result in a set of discreet numbers corresponding to the real world signal at the exact moment in time it was measured. A mathematic interpolation is then performed that connects these points. If the sampling time is too low it is possible that a false signal will be generated. The higher the sampling frequency the higher the resolution of the signal will be.

The data for the FCDAQ system was initially estimated at a speed of 5-7 KHz or between 5,000 and 7,000 times per second. This meant that in order to fulfill the Shannon sampling theorem the FCDAQ would have to collect the data at over 10,000 times a second on each of the 6 ports. Not having an exact number for this led to a FCDAQ that could have an adjustable sampling range from 1 to 10000 samples per second on each of the 6 channels. This was convenient when it was discovered that the actual flutter took place at only 15 Hz.

3.2.6.3 Wheatstone Bridges

Strain gauges are used to measure strain on a system. An ideal strain gauge measurement is usually achieved by connecting several strain gauges to each other in different orientations; this setup is known as a Wheatstone Bridge (See Figure 27). There are three different variations to this setup; the first is known as a quarter bridge, which consists of one strain gauge and three resistors. The second setup is known as a half bridge, which consists of two strain gauges and two resistors. Finally, there is a full bridge configuration which consists of four strain gauges hooked together. This is an ideal setup because the designer can set the bridge in the proper configuration to the direction that the stress is applied to and cancel out the other loads.

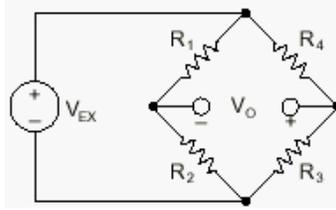


Figure 27: Wheatstone Bridge [25]
© 2008 National Instruments Corporation

At the start of an experiment a properly setup up strain gauge should read zero volts. This is because there are equal resistances on both sides of the Wheatstone bridge. When stress is then applied to the gauge, it undergoes strain. This changes its resistance, which is measured in terms of a voltage gain or drop. The equation for measuring this is,

$$V_0 = \left[\frac{R_3}{R_3 - R_4} - \frac{R_2}{R_1 - R_2} \right] V_{EX} \quad (13)$$

R_1 , R_2 , R_3 and R_4 represent the static resistances of the bridge; R_4 can be set to equal the strain gauge resistance in its equilibrium state. V_{EX} is defined as power supply voltage that is supplied to the Wheatstone Bridge. V_0 is the output voltage that is measured using LabVIEW. Hence the equation for R_4 as the resistance of the strain gauge changes to,

$$R_4 = \frac{V_{EX}(R_1 + R_2) \cdot R_3}{V_0(R_1 + R_2) + V_{EX} \cdot R_2} - R_3 \quad (14)$$

It is critical that the proper bridge, amplifier, and shielded wire are selected for any Wheatstone bridge configuration. This is the balancing equation for a Wheatstone bridge,

$$\frac{R_1}{R_2} = \frac{R_4}{R_3} \quad (15)$$

Any change in length of each arm will result in a non-zero output voltage. Each R represents a different resistor value, and V_0 is the initial voltage. When properly balanced, $V_0 = 0$.

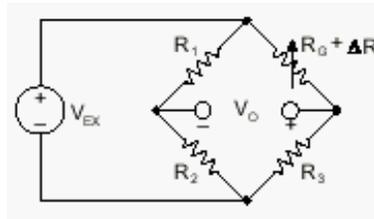


Figure 28: Quarter-Bridge Circuit [25]
 © 2008 National Instruments Corporation

The sensitivity of the quarter bridge can be doubled by adding another active gauge. This configuration is called a half bridge (See Figure 29).

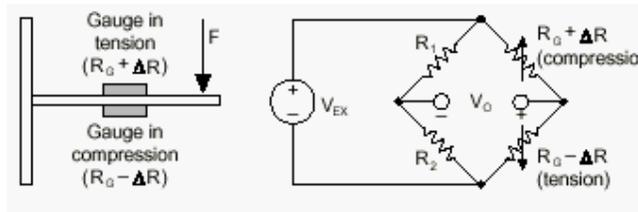


Figure 29: Half Bridge Circuit [25]
 © 2008 National Instruments Corporation

To even further increase the sensitivity the half bridge can be doubled into a full bridge configuration (see Figure 30).

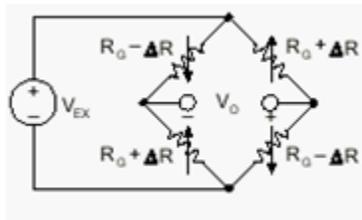


Figure 30: Full Bridge Circuit [25]
 © 2008 National Instruments Corporation

3.2.6.4 LabVIEW Program

The program LabVIEW was used in the first experiments and for data collection for the wind tunnel measurements. This program allows for the creation of “virtual instruments” or VI’s to collect, store and analyze data on several different sensors and setups. The final LabVIEW program is described later in the paper.

The purpose of the LabVIEW program was to provide an easy interface between the National Instruments hardware and multiple test setups. The system easily recorded data between adjustable frequency ranges of 1 Hz and 10 kHz, as well as exported this data to an Excel spreadsheet.

The LabVIEW program was written in several stages, the first stage began with a basic oscilloscope VI program that was taken from example VI's contained in LabVIEW. This program represents an ideal starting point as the frequency of sampling can be chosen, and the interface was already set up to view signals. However, at this point the system still lacked any analysis or recording capabilities.

The second stage of the program development consisted of modifying the original program with the ability to save a specific amount of data to an Excel file. The amount data that was saved could be specified by the user at any time before a record button was pressed. This feature was helpful for figuring out the amount of data that is needed to make reasonable conclusions, as well as aid in the ability to choose the correct memory size for the flight hardware.

The final stage of this program included the integration of a fast Fourier transform that can give the resulting frequency of the fluttering object. The user will have the ability to average the results taken over multiple attempts to get more reliable data. It is important to note that the LabVIEW program was verified for accuracy with a National Instruments BNC-2120 board, which included signal generator.

The first step in setting up a virtual instrument was starting and loading the LabVIEW VI. The virtual instruments for this experiment mimicked the function of an oscilloscope. The next step was to set up the fin. This was done by carefully computing the geometric center of the desired shape, lightly sanding the surface and super-gluing a strain gauge to the desired location in the proper orientation to achieve the correct bridge. When this was accomplished the calibration procedure was followed; data was collected and stored for future analysis.

The VI in Figure 31 was from a previous class project taken at WPI. It has features to adjust the time base and the voltage per division, changing these dials resulted in the change of the resolution of the different signal inputs. This also allowed for each square to have an exact value for time, and voltage. When the input signal from a strain gauge was measured the change of voltage was seen as a function of time. This was the data collected for each fin at different intervals.

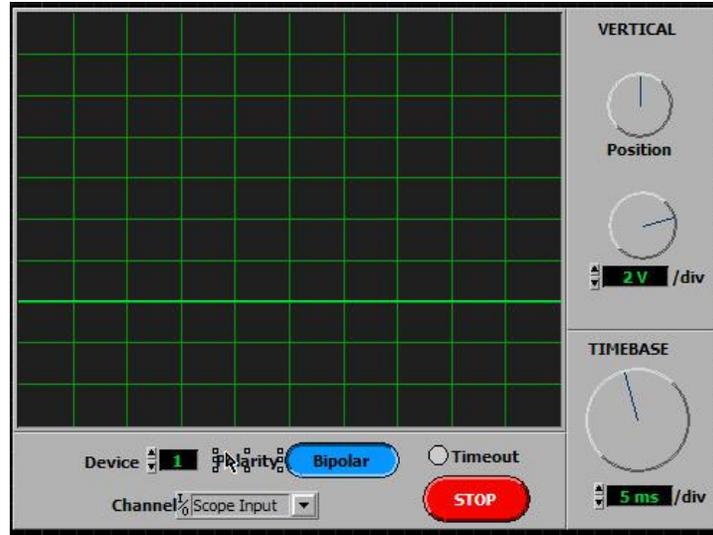


Figure 31: Sample VI Visualizer²
 © 2008 National Instruments Corporation

Figure 32 is a second oscilloscope VI, which left the display length as the time constant, allowing for

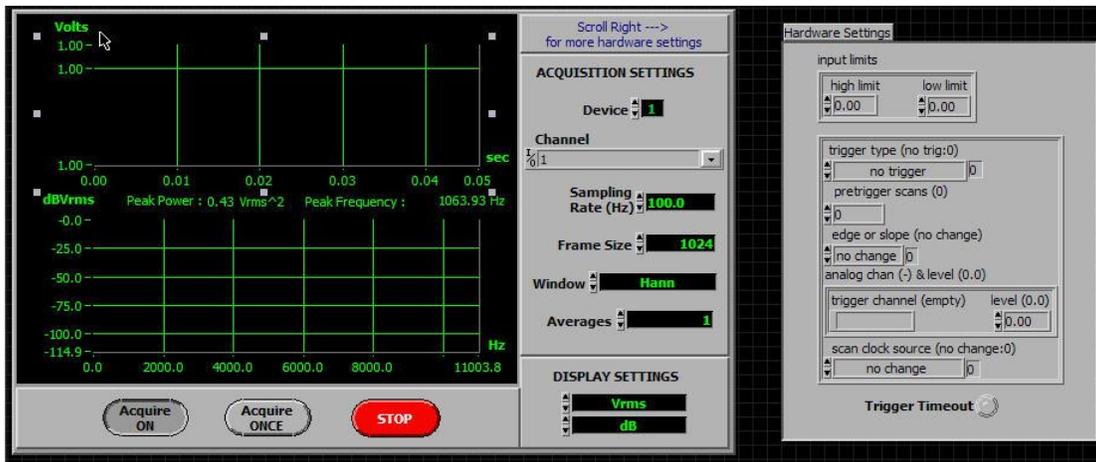


Figure 32: Oscilloscope, with recording function VI
 © 2008 National Instruments Corporation

a choice of sampling rates. With this selected the same measurements were performed using the strain gauge. This data was then plotted in a similar fashion as the first example VI.

² These two VI's are Oscilloscope.VI. They were created by national instruments and distributed freely as example code Nation instruments maintains a copyright as of 2008.

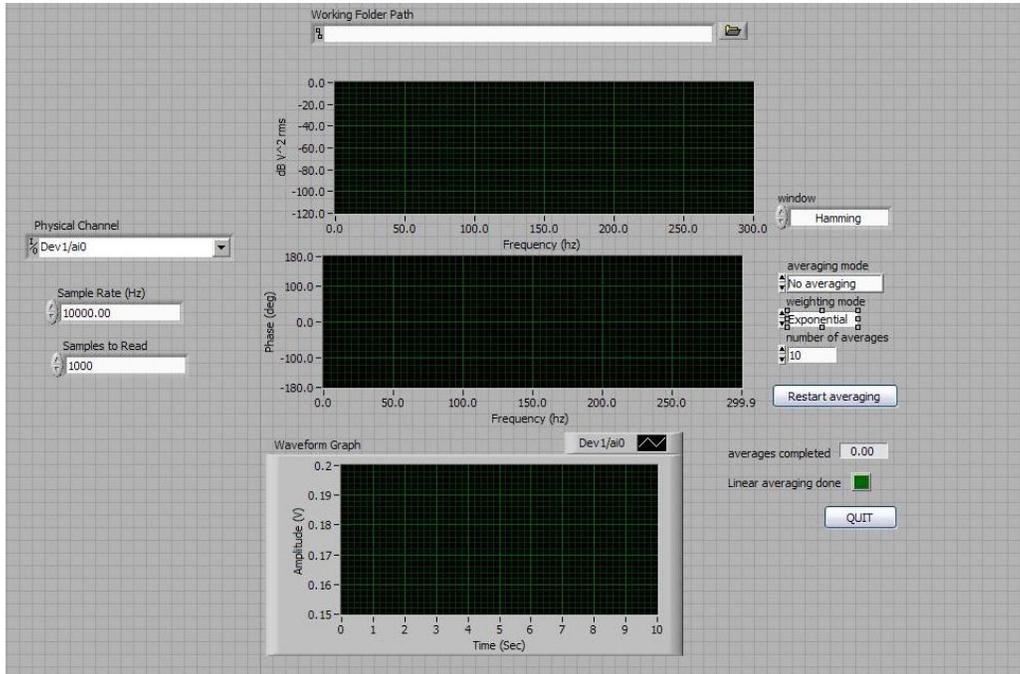


Figure 33: Final Program Interface

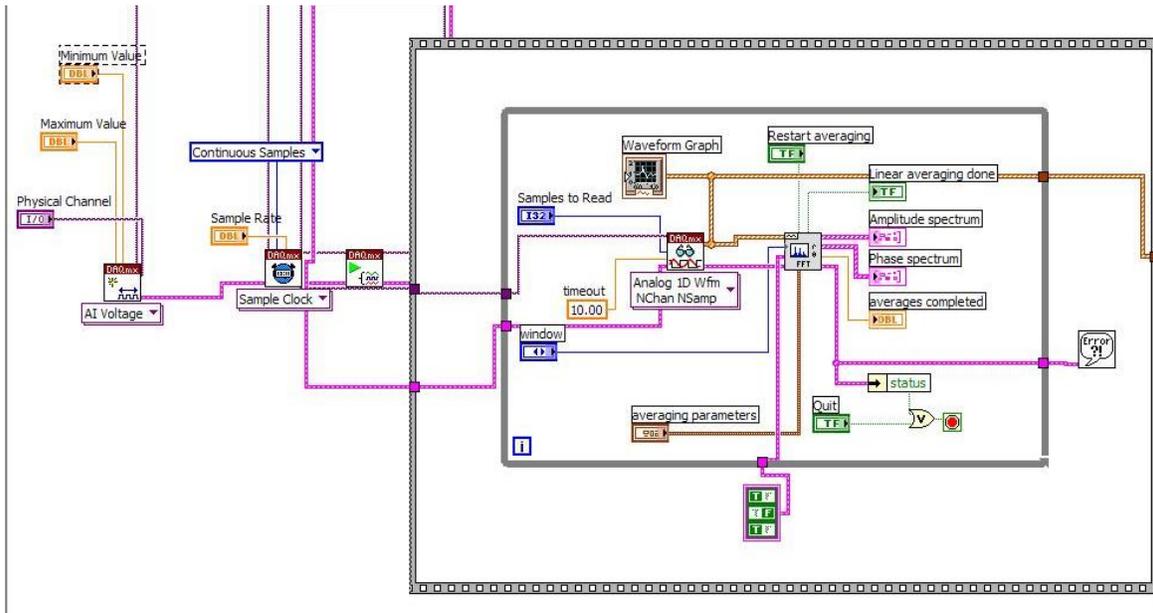


Figure 34: LabVIEW code for FFT, and Oscilloscope

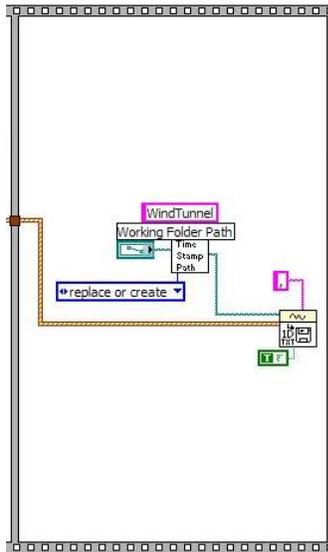


Figure 35: LabVIEW Code for Writing to a File

3.2.6.5 Amplifier Board

There were two amplifiers used for fin flutter testing, which increased the voltage from the strain gauges. The first amplifier was a 2310 Vishay Measurements Group signal conditioning amplifier; this instrument allowed for both single and dual Wheatstone bridge support. It features an adjustable gain setting as well as an adjustable excitation voltage to properly amplify and separates the strain gauge signal

from the background noise. These amplifiers are large in size, and cannot be flown aboard the final rocket. Therefore an onboard data acquisition system was developed to accommodate this issue.

3.3 Hybrid Rocket Propulsion

The primary goals of creating a hybrid rocket engine were to research alternatives to conventional solid systems, as well as integrate new types of fuel into mainstream use. This project sought to build a hybrid engine out of a paraffin fuel grain and use nitrous oxide as the oxidizer. The goal was to produce stable combustion using this configuration.

3.3.1 Motor Design Criteria

This study was conducted in search for increased performance over standard commercially available hybrid rocket motors at thrust levels below 1kN. Two methods of enhancing performance characteristics were identified from research into standard motors; selection of fuel formulations and design of N₂O injectors. Since these new motor designs would be tested in WPI facilities, high factors of safety were necessary.

The motor tested in this study was limited to a total installed impulse of 640 Ns with N₂O as the oxidizer.

$$I_t = F \cdot t = \int_0^{t_b} F \cdot dt_b \quad (16)$$

Since total $I_t = 640$ N s, the average thrust can be calculated for different burn times:

bt (s)	F (N)	F (lbs)
1.00	640.00	145.45
2.00	320.00	72.73
3.00	213.33	48.48
4.00	160.00	36.36
5.00	128.00	29.09
6.00	106.67	24.24
7.00	91.43	20.78
8.00	80.00	18.18
9.00	71.11	16.16
10.00	64.00	14.55

Table 1: Average impulse of a motor with a total impulse of 640 Ns with respect to burn time

The motor was to be designed to be able to lift a small vehicle to an altitude greater than 1500 feet, as well as provide a test bed for injector analysis. As such, the burn time needed to be long enough for some transient ignition conditions to dampen. The thrust and pressure magnitude had to be high enough to provide significant differences in the thrust and other sensors for accurate data collection.

In order to determine performance characteristics of the motor the following values had to be determined: Combustion chamber pressure (P_c), pressure of N_2O (P_{inj}), exhaust pressure (P_e) and thrust (F). The pressure values P_c , P_{inj} were obtained during the testing of the motor with pressure transducers. (PA4338-1500-24318 by Aerojet General Corp.) The thrust was measured using a 50lbf load cell (Measurement Specialties Inc. FC2231).

The design parameters of the motor were established by two methods of simulation. The first of which was a series of equations derived from a collection of relevant previous research. This was compiled in an Excel document detailed below. The second method, mainly for validation, was a spreadsheet produced by SkyRipperSystems, an amateur rocket motor company³. The inputs for both Excel design spreadsheets are common parameters which allow for redundancy in the motor design calculations. Optimization of the design is completed by finding a thrust level and combustion chamber

³ SkyRipperSystems is a motor manufacturer serving the amateur rocketry community. This company focuses on delivering hybrid propulsion at low cost to consumers in the 100 - 5,000N thrust levels. Website and information can be found at <http://www.skyrippersystems.com>

pressure that allows for all the components to be manufactured using common tooling. The part designs were limited to those with no more complexity than four axis milling, or four axis turning.

Only an opacifier was used during the testing of the paraffin motors, due to the high regression rates that were seen as well as time constraints on testing. The opacifier was lamp black, a fine charcoal powder. This charcoal was added to the molten wax during casting in both 0.5% and 1% of the total mass of the fuel grain. Tests were also completed using graphite powder; however, these were found to be a poor additive choice due to the insolubility of graphite in paraffin wax.

3.3.2 CFD Injector Modeling

Due to time constraints and lack of proficiency in FLUENT, only two dimensional models were created of two different injectors. The two injector models created were a single port axial injector, and a four port axial injector. A cross section of each was modeled using SolidWorks, then imported into GAMBIT, and finally used for CFD. The process for modeling each injector is explained in the following subsections.

3.3.2.1 Gambit

For FLUENT to be able to model fluid flow, a mesh of the flow area had to be created. In order to create this mesh, a cross section of each injector had to be taken from the respective models. For this cross section to be imported into GAMBIT it had to be saved as an Initial Graphics Exchange Specification (IGS) file (which have an .igs file designation). It is a file which saves points in space relative to an origin and then describes the specifications of the way those points are connected (such as a face or volume). In order to save the cross section as an .igs file, a face needed to be created. This was completed by using the “create planar face” option in SolidWorks. After this was done to each cross section the files were imported into Gambit for meshing.

Once the file is imported, the walls and face of the object are automatically created using the pre-existing origin and axis. This can cause some initial problems if the axes were not originally set up for the coordinate system of FLUENT. Once oriented correctly the actual mesh was created. It is possible to create a finer mesh in certain areas, such as the injector flow area, in order to achieve a more detailed flow analysis. The four port injector used the same mesh spacing at all areas to save time and computing power. These differences in meshed areas account for only a difference of 0.1 square millimeters compared to 0.5 square millimeters at the largest difference, so the impact on the flow analysis was expected to be minimal.

After the face was meshed the boundary conditions for the flow needed to be set. These included the flow inlets, outlets, and walls. By default when exporting a mesh, Gambit will automatically classify

any undefined boundary lines as walls, and any undefined mesh as the fluid. After defining the inlets and outlets in Gambit, it was ready to be exported into FLUENT as a two dimensional flow. The meshes for the single port and four port injectors can be seen in Figure 36 and Figure 37 respectively.

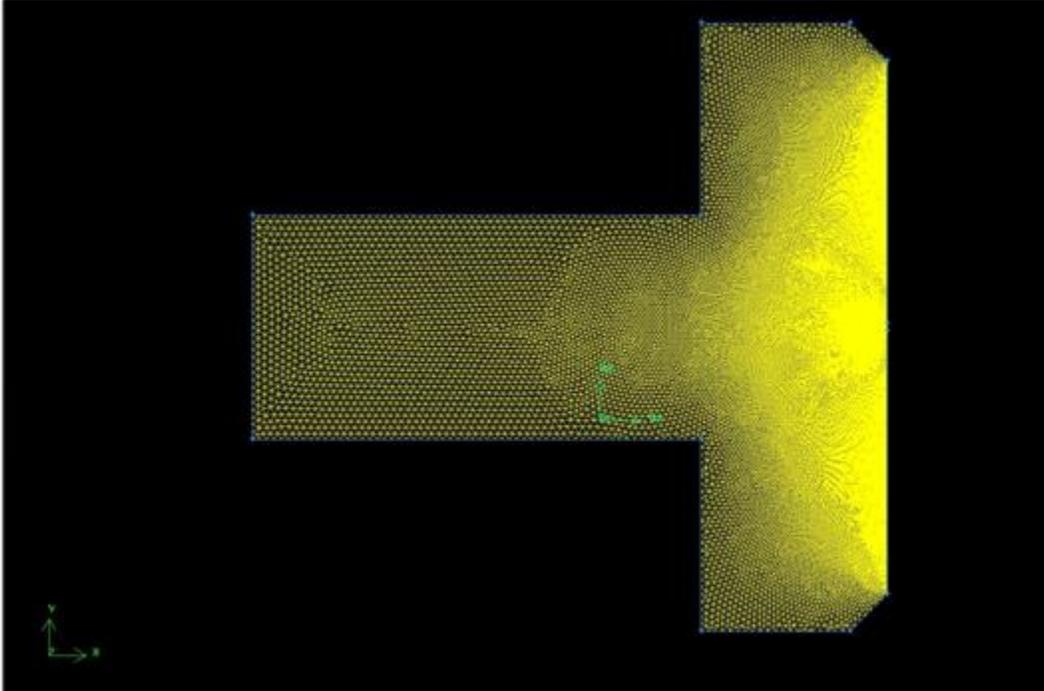


Figure 36: Single Port Injector Mesh

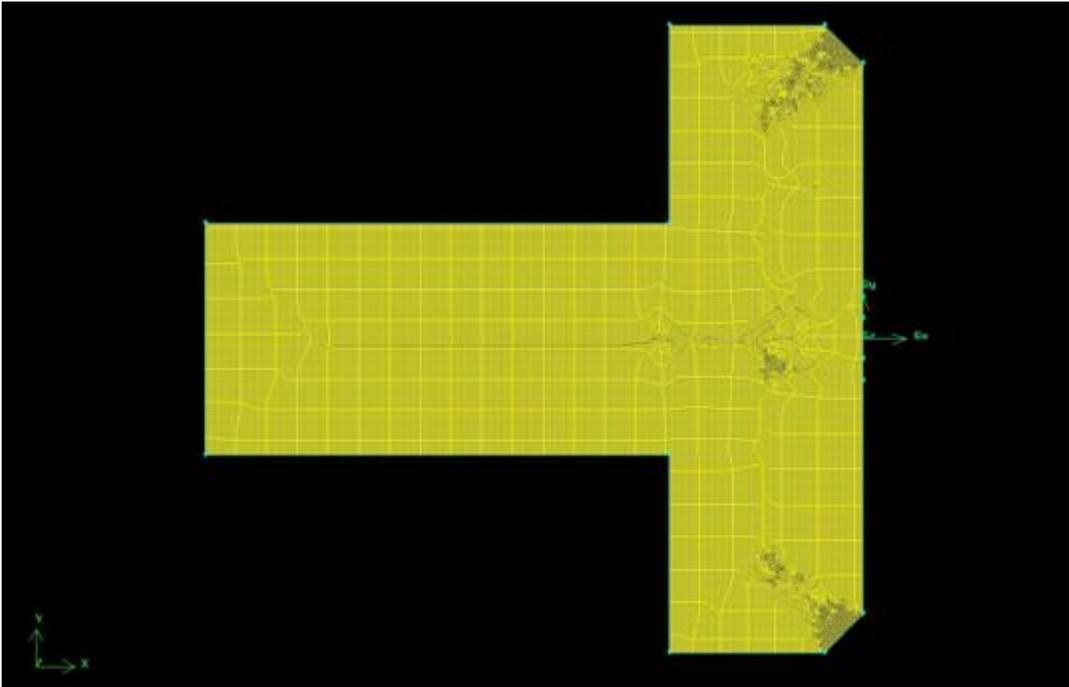


Figure 37: Four Port Injector Mesh

3.3.2.2 Modeling in FLUENT

To increase proficiency in FLUENT, an introductory class was given by WPI Staff member Adriana Hera. The next step was to complete a series of tutorials which are listed below:

1. Introduction to Using **FLUENT**
2. Modeling Species Transport and Gaseous Combustion
3. Modeling Evaporating Liquid Spray[22]

After importing the meshes into FLUENT, a grid check had to be performed. A grid check tests each cell in the mesh to see if it contains a finite volume. Sometimes meshes can become skewed and inverted. When this happens the grid check will fail, and report that non-positive volumes exist. After the check succeeded, the flow properties were determined. In both models the same conditions, solvers, and equations were enabled so that the only differences between the two were the injector configurations.

3.3.2.3 Defining the Solver and Conditions

The next step in solving any computational fluid dynamics problem in FLUENT, is to define what solvers the program will use. There are a variety of factors which can be taken into account such as a pressure, density, dimensionality based solutions, and if the problem should be solved with a steady-state or unsteady solution. Initially solving this problem will require the following solvers to be used:

- Pressure based
- 2D Space
- Absolute Velocity Formulation
- Green-Gauss Cell Based
- Implicit Formulation
- Steady State
- Superficial Velocity Porous Formulation

For a complete description of each solver and what other options are available one can look at Reference 22.

After the solvers have been selected the next step is to enable certain equations to solve for the boundary layers. Normally in a model of a rocket engine one would expect to enable the energy equation for things such as heat transfer coefficients and combustion properties. This model assumes a “cold flow” or no heat exchange or combustion. In using this assumption the energy equation does not need to be

enabled, and was not used for this model. One equation that must be enabled is the viscous parameter equations. Use of this equation does not require assuming laminar flow in the boundary layers, but will calculate factors such as turbulence and viscosity. [22] This model uses a standard $\kappa - \epsilon$ (Kappa - Epsilon) model. The constants in this model have been chosen to provide the most accurate data over the widest range of possible problems. After the equations and models were defined, operating and boundary conditions followed.

Defining the operating conditions consists of deciding the operating pressure and the implementation of gravity. The operating pressure was set to atmospheric pressure (101 kPa). FLUENT operates entirely in gauge pressures so a gauge pressure of almost 2068 kPa in the combustion chamber would be expected. Gravity was enabled as well, although with this model being a jet under very high pressure, 47 atm, it is unlikely that it would have much of an impact on the final outcome.

The boundary conditions were the next factors which needed to be set. It was important to define the models inlet and outlet flows correctly. The inlet was chosen to be a pressure inlet because this was easily measured by the pressure transducer. A value of 4826 kPa was chosen as the inlet pressure. The outlet was simply chosen as a pressure outlet with an outflow weighting of one. This means that all flow in should flow out at each iteration.

3.3.2.4 Material Definition and Solver Controls

In order for FLUENT to accurately solve any problem involving a particular liquid or gas, certain variables about that fluid must be defined. FLUENT has a built in database with a variety of liquid properties for use in simulations. It does not, however, contain nitrous oxide vapor which was used in testing. In this case water vapor, which FLUENT did have in its database, was used as a substitute in this model. The default values for water vapor were again kept constant.

In order for FLUENT to produce more accurate results, the solver controls all had to be set to insure high accuracy and to accurately depict the hybrid motor testing. Low accuracy would be used for faster computations; however a higher accuracy is more desirable for detailed analysis. Pressure, momentum, turbulent kinetic energy, and turbulent viscosity were all set to use second order upwind equations. The Pressure-Velocity coupling was also changed from “Simple” to “Coupled”. This is due to the fact that pressure directly influences velocity in this problem. The Courant Number was kept at the default value; however the Explicit Relaxation factors were changed. In this problem the pressure drastically influences the solution, which is why the momentum factor should be set so that it is lower

than the pressure factor. Experimentation with these values, using the simple injector as a control, yielded values of 0.85 and 0.5 for pressure and momentum relaxation factors respectively.

3.3.2.5 Initial Results and Unsteady Modeling

After all parameters and equations have been defined the problem needed to be initialized and then iteration could begin. Using the inlet as the initialization point, both models were then iterated 127 times. This number was chosen because in both models it represented a low point in the mass imbalances. If left to keep iterating the problem eventually turns into an expanding oscillation and yields no usable data. By definition, a high pressure jet of liquid is not a steady state problem and thus cannot be solved using a steady state solution. After the initial steady state solution, it was necessary to change the solver to an unsteady state solution. The final mass imbalances can be seen in Figure 38 and Figure 39 for the single port and four port injectors respectively.

Mass Flow Rate	(kg/s)
inlet	42941.273
outflow	-42941.07
Net	0.203

Figure 38: Mass Balance (Single Port)

Mass Flow Rate	(kg/s)
pressure_outlet.2	-2837.9373
velocity_inlet.1	2837.9739
Net	0.0366

Figure 39: Mass Imbalance (Four Port)

3.3.2.6 Checking Model Consistency

In order to begin analysis on the models it first had to be determined that simulations were accurate by visual inspection of the velocity contours. Velocity contours of each model can be seen in Figure 40 and Figure 41.

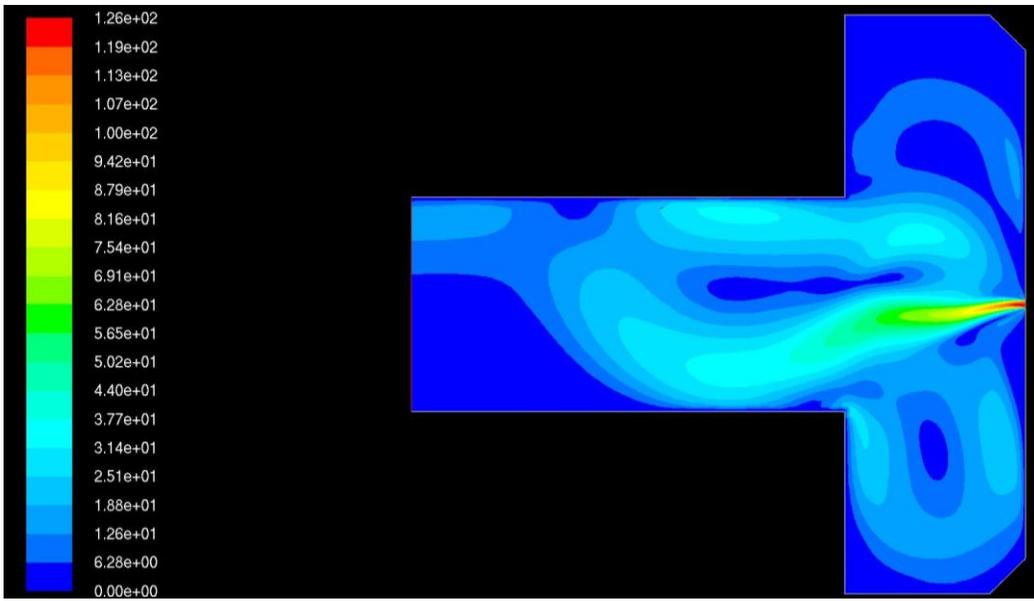


Figure 40: Velocity Contours of Single Port Injector

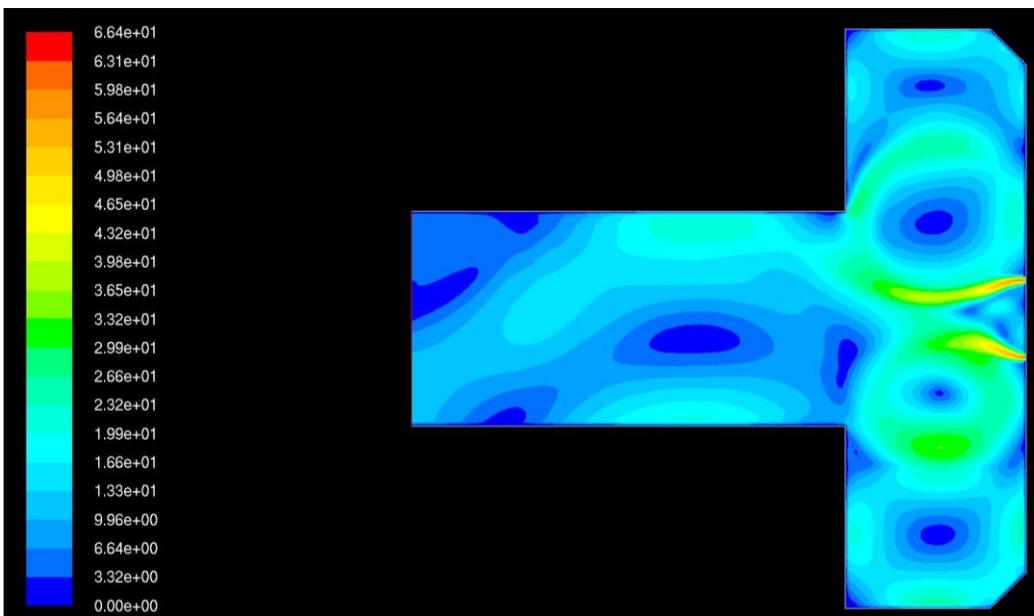


Figure 41: Velocity Contours of Four Port Injector

3.3.3 Motor Hardware Design

For initial testing, a small PVC motor was designed to demonstrate the fundamentals of a hybrid rocket motor, as well as create a starting point for future development. This motor also allowed for testing of all sub-systems that would be used on larger motors.

A motor was designed using simple one dimensional analysis. Since PVC and N₂O were chosen for the propellants, the Isp was found using ProPep⁴, and compared with published data as a check of proper calculation. The combustion flame temperature, ratio of specific heats, and average molecular weight of the exhaust gasses were also derived using ProPep for a given oxidizer to fuel ratio.

Name	Symbol	Value	units
Fuel Density	ρ_{PVC}	1.32	g/cm ²
Oxidizer Density	ρ_{N_2O}	813.7	kg/m ²
Specific Impulse	Isp	178	Sec
Temp of combustion	T _c	2800	K
Regression rate function	n	0.65	
Regression rate function	a	0.02	mm/s
Ratio of specific heats	γ	1.25	
Oxidizer to fuel ratio	OF	4.5	
Molecular weight of exhaust gasses	μ_{exhaust}	24	kg / mol
Pressure at exit	p _{exit}	1	atm
Combustion chamber pressure	p _{chamber}	86	psia

Table 2: Propellant Properties

With these assumed values, a series of calculations were used to derive the necessary physical parameters in order to produce the 22N of thrust. First the exit velocity was calculated.

⁴ ProPep is an amateur rocketry program for solving combustion equilibrium equations, the program can be found at <http://www.rocketry.net/propep.html>

$$M_e = \sqrt{\left(\frac{2}{\gamma - 1}\right) * \left[\left(\left(\frac{P_c}{P_e}\right)^{\frac{\gamma-1}{\gamma}}\right) - 1\right]} = 1.829 \quad (17)$$

The physical design parameters were then calculated through intermediate steps. The following equations lead to the single port injector sizing, nozzle throat diameter, and nozzle exit diameter.

The total impulse is defined by:

$$I = \int F dt = 22N * 10s = 220Ns \quad (18)$$

With a known Isp from ProPep, the total propellant mass could be calculated.

$$m_p = \frac{I}{Isp} = 1.24N \quad (19)$$

Flow properties are based on the desired impulse and Isp. These values are all calculated with a N₂O temperature of 75°F. The total mass flow rate is found by the given burn time and the calculated Isp.

$$\dot{m}_{exit} = \frac{m_p}{t_b * g} = 0.013 \frac{kg}{s} \quad (20)$$

With $\dot{m}_{exit} = \dot{m}_{total}$, the mass flow rates of the oxidizer and fuel can be independently calculated based on the OF ratio. This calculation however assumes the OF to be constant through the operation of the motor. This is further discussed below.

$$\dot{m}_{ox} = \dot{m}_{exit} \left(\frac{OF}{OF + 1}\right) = 0.01 \frac{kg}{s} \quad (21)$$

$$\dot{m}_{fuel} = \dot{m}_{exit} \left(\frac{1}{OF + 1}\right) = 0.00229 \frac{kg}{s} \quad (22)$$

This low mass flow rate allowed for multiple test firing sessions without the need to refill any tanks. With only a tenth of a kilogram of N₂O used per test, the cost of each firing was roughly \$1.47 depending on the supplier.

Throat temperature can be calculated based the calculated gamma.

$$T_{throat} = T_c \left[\frac{1}{1 + \frac{\gamma - 1}{2}} \right] = 2490 K \quad (23)$$

Nozzle parameters are based on nozzle theory of a 2 dimensional compressible flow. The diameter of the throat is the key parameter; where machining of the nozzle must be accurately match this value in order to have an accurate simulation of the experimental motor.

$$A_t = \frac{\dot{m}_{exit} * \sqrt{R * \frac{T_t}{\mu}}}{p_c * \Gamma} = 2.999 * 10^{-5} m^2 \quad (24)$$

$$D_t = 2 \sqrt{\frac{A_t}{\pi}} = 0.00618 m (0.24 in) \quad (25)$$

The exit areas of the motors developed in this project were all optimized for ambient pressure at sea level.

$$A_e = A_t * \left[\frac{\Gamma}{\sqrt{\frac{2\gamma}{\gamma-1} * \left[\left(\frac{p_e}{p_c}\right)^2 - \left(\frac{p_e}{p_c}\right)^{\frac{\gamma+1}{\gamma}} \right]}} \right] = 4.648 * 10^{-5} m^2 \quad (26)$$

$$D_e = 7.693 * 10^{-5} m = 0.303 in \quad (27)$$

Injector port sizing is done by assuming that chamber pressure is constant. Discharge coefficients were found using the table on page 367 of Sutton's Elements of Rocket Propulsion.

$$A_{injector} = \frac{\dot{m}_{ox}}{C_d \sqrt{2\rho_{ox} \Delta p}} = 2.531 * 10^{-7} m^2 \quad (28)$$

$$D_{injector} = 0.022 in$$

Following these calculations, the grain properties were then determined using the OF ratio to find the surface area. With the following equations it is noted how the OF ratio can have an effect on the regression rate. This is due to the inverse relationship between the diameter of the port in the fuel grain and the regression rate.

$$OF = \frac{\dot{m}_{ox}}{\dot{m}_{fuel}} \quad (29)$$

$$\dot{m}_{fuel} = \rho_{fuel} \dot{r} A_b \quad (30)$$

Where we can use substitution of $\dot{r} = aG^n$ where $G = \dot{m}_{ox}/A_{cs}$, where A_{cs} is the cross sectional area of the grain to find \dot{m}

$$\dot{m}_{fuel} = \rho_{fuel} a \left(\frac{\dot{m}_{ox}}{A_{cs}} \right)^n A_b \quad (31)$$

Through more substitution of A_{cs} and A_b where D is the diameter of the port and L is the length of the port

$$A_{cs} = \pi \left(\frac{D}{4}\right)^2 \quad (32)$$

$$A_b = 2\pi \left(\frac{D}{2}\right) L \quad (33)$$

$$\dot{m}_{fuel} = \rho_{fuel} a \left(\frac{\dot{m}_{ox}}{\pi \left(\frac{D}{4}\right)^2}\right)^n 2\pi \left(\frac{D}{2}\right) L \quad (34)$$

Now with manipulation to find $\frac{\dot{m}_{ox}}{\dot{m}_{fuel}}$

$$\begin{aligned} \frac{\dot{m}_{fuel}}{\rho_{fuel} a 2\pi \left(\frac{D}{2}\right) L} &= \left(\frac{\dot{m}_{ox}}{\pi \left(\frac{D}{4}\right)^2}\right)^n \\ \ln\left(\frac{\dot{m}_{fuel}}{\rho_{fuel} a 2\pi \left(\frac{D}{2}\right) L}\right) &= \ln\left(\left(\frac{\dot{m}_{ox}}{\pi \left(\frac{D}{4}\right)^2}\right)^n\right) \\ \ln\left(\frac{\dot{m}_{fuel}}{\rho_{fuel} a 2\pi \left(\frac{D}{2}\right) L}\right) &= \left(n[\ln(\dot{m}_{ox}) - \ln(\pi \left(\frac{D}{4}\right)^2)]\right) \\ \left(\frac{\dot{m}_{fuel}}{\rho_{fuel} a 2\pi \left(\frac{D}{2}\right) L}\right) &= e^{(n[\ln(\dot{m}_{ox}) - \ln(\pi \left(\frac{D}{4}\right)^2])} \\ \left(\frac{\rho_{fuel} a 2\pi \left(\frac{D}{2}\right) L}{\dot{m}_{fuel}}\right) &= \frac{1}{e^{(n[\ln(\dot{m}_{ox}) - \ln(\pi \left(\frac{D}{4}\right)^2])}} \\ \left(\frac{1}{\dot{m}_{fuel}}\right) &= \frac{1}{e^{(n[\ln(\dot{m}_{ox}) - \ln(\pi \left(\frac{D}{4}\right)^2])} \rho_{fuel} a 2\pi \left(\frac{D}{2}\right) L} \end{aligned}$$

This leads to the final form of the OF ratio

$$OF = \left(\frac{\dot{m}_{ox}}{\dot{m}_{fuel}}\right) = \frac{\dot{m}_{ox}}{e^{(n[\ln(\dot{m}_{ox}) - \ln(\pi \left(\frac{D}{4}\right)^2])} \rho_{fuel} a 2\pi \left(\frac{D}{2}\right) L} \quad (35)$$

To match the optimal OF ratio to the middle of the burn, the chosen grain geometry called for a grain that was 1.9 inches long with a core of 0.375 inches. The additions of 0.5 inches pre and post

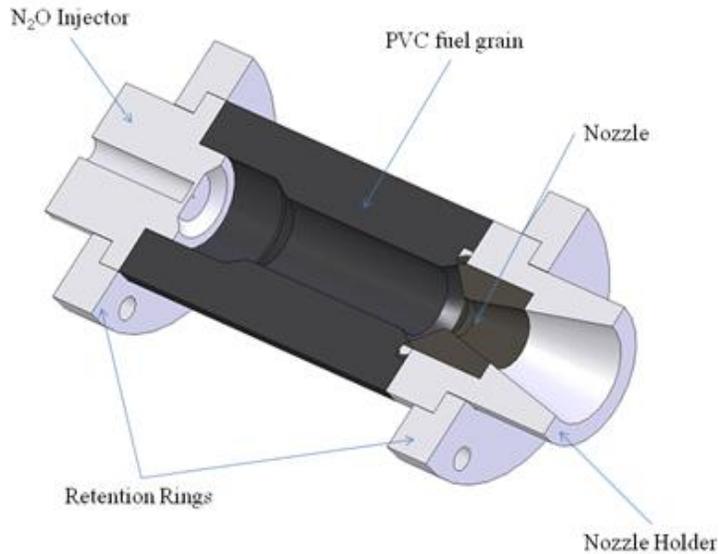


Figure 42: Cut away view of the 5 lb boilerplate motors with PVC fuel grain

combustion mixing chambers were manufactured in the grain itself since this would allow for a simple fuel grain to act as the entire combustion chamber.

The equations were entered into an Excel spreadsheet where optimization could occur for manufacturability since the small injector port was limited to a drill bit size that was available. The calculated pressures and temperatures guided the material selection for the combustion chamber and nozzle.

The nozzle material was chosen to be a dual construction of a graphite insert and an aluminum nozzle holder and expansion cone. The injector and nozzle holder were milled from a billet of 6061 aluminum. The nozzle insert was turned on a manual lathe and pressed into the nozzle holder. The fuel grains were turned from 2 inches extruded grey PVC bar to form a complete motor see Figure 42

A boiler plate motor design, large safety factor with non-flight capable hardware, was chosen to allow for quick disassembly and high turnaround time from test to test. The design allows for any length of fuel grain and casing to be attached, with the only needed change to be the length of the threaded rods. Shown below is the tested motor design for the 5lb and 25 lb N₂O/PVC motors.

All components of the motor are made of aluminum, excluding the nozzle insert and the steel threaded rods not seen in Figure 42. The nozzle insert was manufactured from extruded graphite in order to prevent throat erosion over the course of multiple firings. Aluminum was chosen due to availability and ease of manufacturing. It should be noted that the aluminum components have a short life of operation. This is due to the surfaces being exposed to hot exhaust gases, which quickly causes oxidation.

Aluminum is also the only metal allowed at amateur experimental launches to be under pressure in a flight vehicle.



Figure 43: Nozzle Holder and Injector after machining. Three inch OD Retention rings shown with each for scale.

In order to begin testing paraffin hybrid motors, an intermediate motor was thought to be a good starting point. An intermediate motor between a 22N and 110N motor was selected at 50 N of thrust. Using the same set of equations with different input values for the updated fuel density, OF ratio, Isp, T_c , and a higher p_c . This motor was the first paraffin wax fuel ever tested at WPI; as such a high safety factor was needed in the design. The higher combustion chamber pressure coupled with the low tensile strength of the fuel grain itself required a metal casing around the fuel grain, unlike the 22N motor design. A 2 inches ID and 3 inches OD case was turned from 6061 Aluminum.

The graphite nozzle insert was set into an aluminum nozzle holder to form the nozzle assembly. This assembly also contained a post combustion mixing chamber 0.5 inches in length. The assembly uses

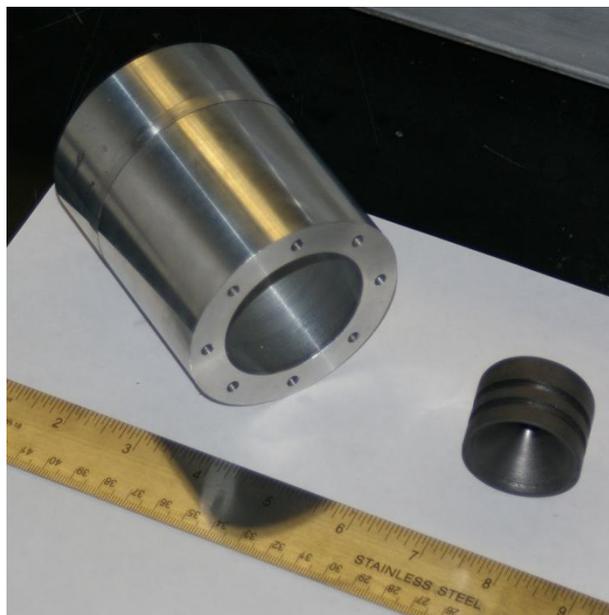


Figure 44: Nozzle holder with combustion chamber attached and nozzle insert

four o-rings to seal the combustion gasses inside the chamber. Two O-rings fit on the nozzle insert, and two larger O-rings seal the nozzle holder to the case. Eight holes for 10-32 bolts to secure the nozzle holder onto the case forced the part to be manufactured, at least in one step, on a mill. Two independent attempts to make this part using a mill failed due to the high tolerance of the hole alignments.

The Injector plate was manufactured in a similar way to the nozzle holder; however, multiple fixtures were necessary to make sure the part was completed as accurately as possible. A pre-combustion mixing chamber again half an inch long was integrated in the injector. This design saved machining time and allowed for a simpler design with fewer parts. The Injector, once milled, was fit on a lathe to turn the O-ring grooves since this could not be easily done on a mill. As a final step in the manufacturing, the injector was taped to fit the oxidizer delivery system as well as a pressure tap for combustion pressure measurements.



Figure 45: Injector assembly

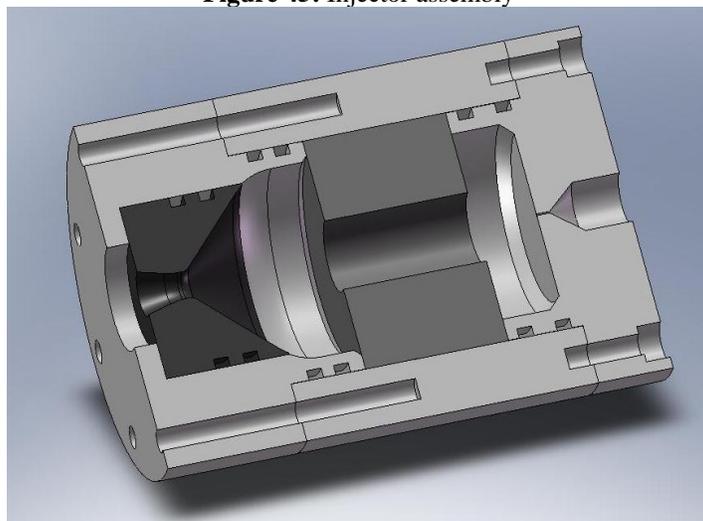


Figure 46: 50 N motor assembly

The entire assembly fit together perfectly when greased, however disassembly proved to be much more difficult than originally anticipated. This was due to the high tolerance that was assumed to be necessary.

Later it was determined that multiple injector orifices were needed. A new injector was manufactured using the same methods described above. Instead of one single orifice, a 4 port injector was machined using a 0.015625 inches drill bit as seen in Figure 47.



Figure 47: Initial flow testing of the four port injector and CAD model showing four port arrangements

3.3.4 Oxidizer Delivery System

Since the motor was designed as a pressure fed system, the oxidizer must be easy to pressurize and nontoxic in case of any leaks in WPI facilities. Oxidizer selection for the hybrid motor was conducted by exploring the safety and costs that were associated with each oxidizer flow system. Oxygen, both Liquid Oxygen (LOX) and Gaseous Oxygen (GOX) were explored. However both would require the flight vehicle and testing vehicle to have all oxygen safe components. Also the necessary safety precautions for storage could not be met in this projects facility; LOX would require a pressurization system, such as a regulated helium supply to maintain delivery pressure. However N_2O , a liquefied gas, maintains a significant pressure for delivery to the system.

The delivery system of N_2O consisted of a 10lb Department of Transportation (DOT) rated tank, safety valve, ball valve, steel braided hose, pressure transducer, 12 volt solenoid, metal to metal seat check valve, and the injector orifice. These components allowed for safe and controllable delivery of nitrous oxide. Later proposed modifications to this included the addition of a second solenoid to allow for

venting of the delivery lines to the atmosphere without going through the injector. For detailed flow chart see Appendix G : Nitrous Oxide Flow Diagram

The fuel for initial testing was PVC, which was machined to the grain geometry seen in Figure 48. PVC was used since paraffin motors often have complex grain geometries, and testing with simple PVC grains, initially, would allow validation of sensors and testing procedures. Paraffin was later used due to its high regression rates, and potential for competing with solids for a highly efficient propulsion system in a small-scale vehicle.

Paraffin Wax is macro-crystalline in nature and is composed mostly of straight chain Alkanes ($C_{18} - C_{36}$) [23]. The remainder is composed of $C_{18} - C_{36}$ Isoalkanes, Cycloalkanes and trace amounts of oil [23]. Paraffin is generally considered non-toxic and is widely used in candle-making, casting, waterproofing, and sealing [23]. Paraffin fuel grains must be cast in a mold from molten paraffin and allowed to cool prior to use. Melting temperatures for Paraffin wax range from 140° to 170° F depending on the grade. This low-temperature manufacturing technique allows for easy casting of fuel grains.

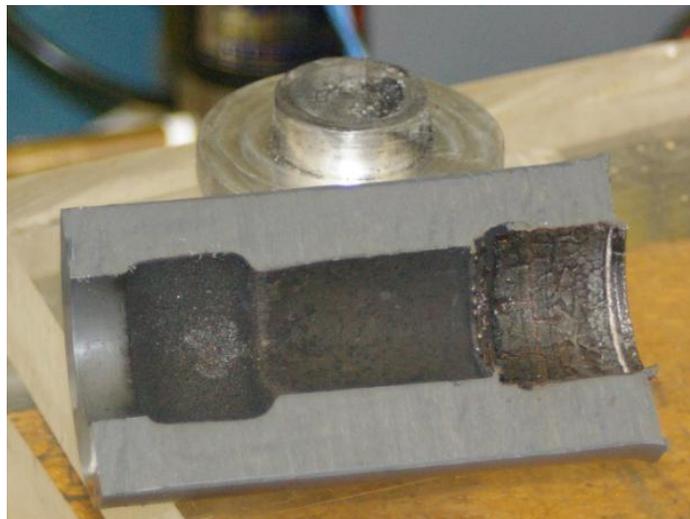


Figure 48: Cut Section of PVC Fuel Grain 3.5 inches in length

3.3.5 Test Stand

The hybrid motor test stand supports both the motor and the electronic components used in data acquisition.

There were different design criteria for the construction of the test stand. In addition to mounting the components above, the design had size, material, and safety constraints. The design had to:

- Fit inside the containment shield with some room to spare, which limited it to approximately two feet high by two feet wide.

- Be strong enough to withstand the force of the motor firing with a significant safety margin.
- Keep the rocket motor as close to the floor as possible to prevent flipping.
- Preferably utilize materials that the WPI machine shops already possessed rather than ordering new material.

These constraints eventually led to the full CAD assembly (with the motor attached) that can be seen in Figure 49. The design is approximately 4 feet long, and utilizes a single linear rail attached to a

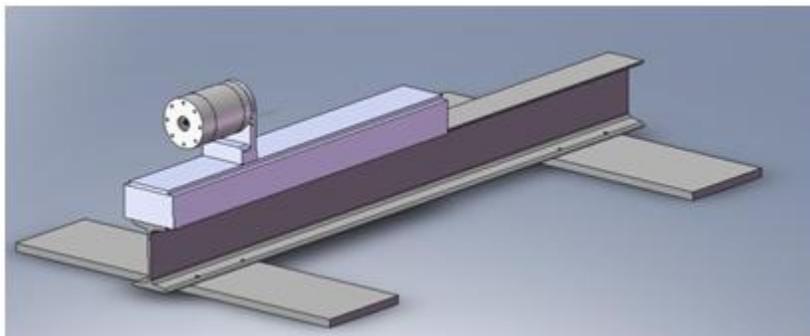


Figure 49: Final Test stand

single I-beam, with aluminum ‘feet’ for stability. The rail itself is three feet long, six inches wide, and is attached using several screws located underneath the rail. A plate was attached to the top of the linear rails to form a flat surface to attach motor hardware to, as the rail which was utilized did not possess an even surface. This also allowed work on the motor while in its test configuration. The center rail could be extended fully backwards outside of the containment shields to allow work on the motor without being cramped.

The motor attached to the plate via a machined stand that was screwed to the rail with small threaded holes. The N_2O and sensor equipment were all attached directly to the motor via simple brass piping; a three foot braided steel line attached the nitrous equipment directly to the nitrous tank.

The test stand went through several iterations before finalization. The very first design utilized small 1010 rails, shown in Figure 50. Two of these fit easily onto the rails of a pre-constructed grooved base beam and could be screwed into the upper piece with 0.125 inches holes. After a design review, it was decided to swap from the bars to a more complex machined design based on safety factors. Placing a

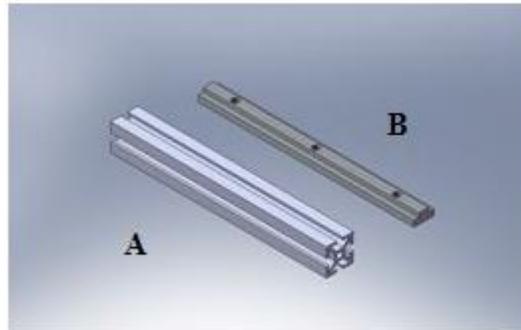


Figure 50: Internal rails used: Internal rails used
A- 1010 rail in initial CAD drawings.
B- Design for interior slide on interim stand

slide inside the grooves, rather than hooking around the top of the rails would provide a larger factor of safety and better stability. Additional minor adjustments were made along the way which related to size issues. The base plates were initially a single large flat plate, but were redesigned into two small strips to save on material and remove unnecessary weight and bulk.

The net result of these design changes is seen in the interim test stand shown in Figure 51, which was constructed and used for the majority of the early non-paraffin grain testing.

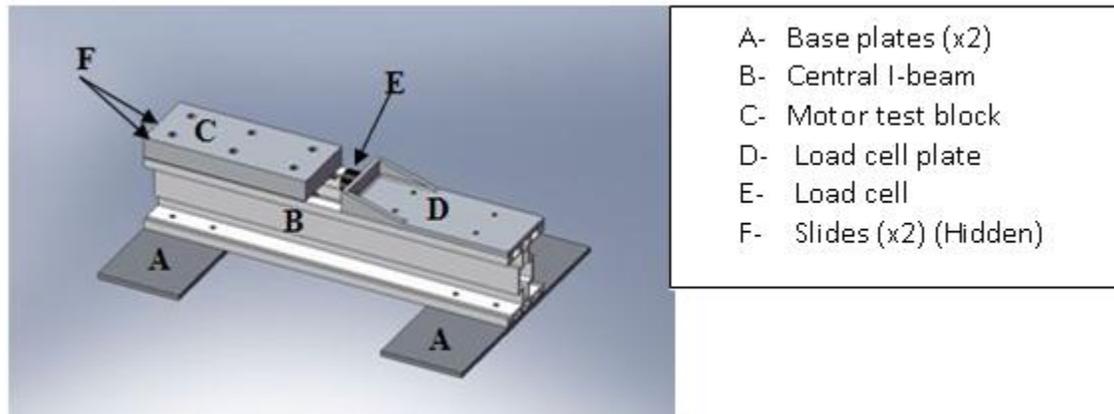


Figure 51: Interim Motor Test Stand

While the safety and stability of this design was satisfactory, it eventually became apparent that there were significant issues with the operation of the test stand. The main problem with this version of the test stand was torque and friction from the mounting of the motor and additional hardware, which caused problems with data collection. While the motor block moved smoothly with no motor attached, the design did not take into consideration that the nitrous lines and pressure transducers directly attached to it. These lines extended over the load cell plate and were heavy enough to put significant torque on the motor plate, causing the front end of the motor block to dig into the central beam. This caused strange force data (the block leaned against the load cell) or, even worse, no data at all when the block became stuck and was not in contact with the load cell.

This issue necessitated a reconstruction of the test stand. The grooved rail was replaced with a solid I-beam attached to a single linear rail, which moved very smoothly using ball bearings. Even with a significant load on top, three Newtons of force is sufficient to move the rail. The rail was also long enough to accommodate all the nitrous control equipment comfortably, which caused very little torque.

3.3.6 Safety Measures

Rocket motors have some inherent dangers. If proper safety measures are implemented, the risk of failure and damage significantly decreases. As a result of the danger of building and testing rocket engines, the group devised a series of safety measures aimed at protecting both itself and the test facility in the event of motor failure. These safety measures include designs for both a containment shield/sound suppression system and a blast deflector. A CAD drawing of the containment shield (A), blast deflector (B), and designed test stand (C) can be seen in Figure 52.

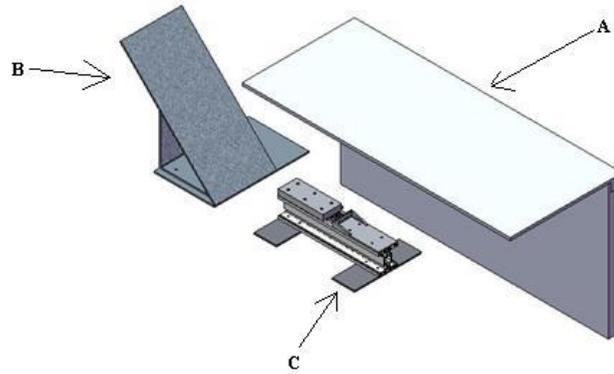


Figure 52: CAD model of Test Stand and Support Equipment

3.3.6.1 Fire lab safety compliance

In order to use WPI's Fire Safety Laboratory, the group had to comply with all the lab safety rules. This required the team to fill out necessary paper work before testing could take place. These forms can be found in 7.4 Appendix F. The purpose of the forms was to outline the type of test and any possible hazards that the test possessed. Along with filling out the forms, certain operational rules had to be followed. These included:

- all persons knowing how to stop the test process in case of emergency,
- wearing proper clothing,
- wearing eye and ear protection at all times,
- Keeping the test room neat and organized.

Many of these rules were implemented into the operational checklist to ensure optimal safety for all the people present during the test and safety for the facility itself.

3.3.6.2 Containment Shield/Sound Suppression

The team first looked into developing a system that would protect against combustion chamber burn through and rupture due to pressure increase, as well as reducing the sound that the motor will generate. It was decided that a three wall enclosure would work best for these goals. The first step was making a preliminary CAD drawing of the system. The size and cost of the containment/sound suppression system was then taken into consideration.

3.3.6.2.1 Size considerations

To accurately determine the size of the system, the test facility was surveyed to see the area and utilities available to work with. It was decided that motor would be placed in an orientation in which the exhaust flame would be directed into the burn room and underneath the vent hood to allow for proper evacuation of the exhaust gases. After the examination of the test area, it was agreed that a side wall of dimension 4' by 2' would offer proper protection. This would allow for the entire test stand unit and support instruments to be properly contained in the event of a motor failure.

3.3.6.2.2 Cost analysis

With the system's dimensions determined, the next step was to choose the materials that the system would be constructed from. The task of assessing materials cost to keep the team within budget was also of importance. The team narrowed the material search down to three materials: plywood, lexan, and aluminum. Aluminum was quickly decided against, because although it would protect the team in case of failure, it would not properly contain any shrapnel that may be ejected from the engine. The next choice, lexan, was also disregarded due to its high cost. It was then decided to use plywood because of its low cost and ease of construction.

While plywood was the least expensive, it was also the weakest of the materials. To compensate for this, a system that utilizes two wall pieces separated by upright supports in between them was designed. To increase the safety factor of the system, it was thought that it would be beneficial to fill the gap between the wall segments with sand. This sand adds more protection for the operators and the test facility in case of failure, and it also suppressed the sound that the motor produced.

The addition of the sand also increased cost. The original design used 2'x 4' wood studs as the internal support structure. To reduce the cost of the sand, the team tried to reduce the area in between the two wall segments. To achieve this reduction, the design was changed from 2'x 4' supports to 2'x 2' supports. This redesign lowered the cost without severely reducing the safety factor of the entire system.

3.3.6.3 Blast Deflector

For further safety precautions, the team decided to design and build a blast deflector. The main purpose of the deflector is to redirect the exhaust upwards into the vent hood. As a secondary safety function, in the event of a nozzle blow-out, the deflector will also deflect the nozzle upwards into the burn room.

3.3.6.3.1 Design

The design process for the blast deflector was kept fairly simple for two main reasons; to reduce manufacturing time and material cost. The deflector was designed so that there was a main bottom plate, to which a vertical up-right piece and an angled piece were attached (see Figure 53). The final design has dimensions of 18 inches wide, 24 inches tall and 10.75 inches deep.

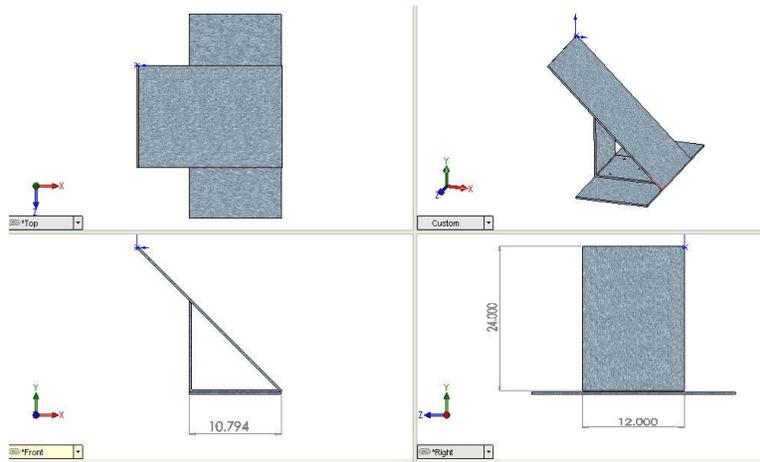


Figure 53: CAD Design of the Blast Deflector

3.3.6.3.2 Manufacture

To save on manufacturing costs, existing material that WPI's machine shop contained was used. The material that was used was a 0.25 inches thick sheet of stainless steel. This material was more than

adequate to handle the exhaust temperatures and debris that may be ejected from the motor during firing. Three pieces of material were measured and cut using a plasma torch to ensure square cuts. After the three pieces were cut, they were welded together.

3.3.7 Testing Procedures

In a process as complicated as rocket motor testing, the possibility of operator error is ever present. In an attempt to reduce any sources of operator error, a checklist was developed to ensure that the set-up and test process are identical for all tests.

3.3.7.1 Complete Checklist of Operations

Before the first test was performed, an operational checklist was developed; see Appendix C: Test Procedure Checklist for the completed check list. The check list was split into three major sections: Pre-Test, Testing, and Post-Test.

The pre test section was dedicated to the set-up and testing of the DAQ and relay systems. The goal of the pre-test checklist was to ensure that all the test and safety equipment was functioning properly prior to the test.

The testing portion of the checklist was used to verify that the test was performed in the same manner every time. This section was also used so that the team did not miss any steps that could cause possible motor failure.

The post test checklist included items to ensure all the testing equipment was in a non-operational state. This meant completing such tasks as purging the nitrous oxide lines and ensuring that the valves and relays we all closed to prevent possible failures.

3.3.7.2 Complete Test Run

Prior to the first hot-fire test, a complete test run using the operational checklist was performed. This test run was to ensure that the checklist included all items necessary to ensure that the tests were performed in a safe and logical order. During these tests, team members were assigned certain portions of the checklist as another means to limit mistakes. With each team member performing the same tasks on every test, it ensured that those tasks were being completed the same way for every test. The test run also served as an opportunity to make alternations to the checklist. As the dry run proceeded, it was noticed that certain tasks were incorrectly sequenced or that more tasks needed to be added. A new test run was

performed for each iteration of the checklist, once both the team members and the fire lab manager were content with the checklist, testing commenced.

3.3.8 Data Collection

The data collection system was a critical part of the hybrid testing section. It enabled verification of motor performance, ability to test various injector designs, and analysis of failed tests to correct errors.

3.3.8.1 Sensors

The sensors are the most basic part of the data collection system. For this study, two sensor types were chosen – pressure transducers, which measured line and chamber pressures, and a load cell, which measured force output. The pressure transducers utilized were rated from 0-1000 psi, with an input voltage of 28V and an output voltage from 0V to 5V. The load cell was rated up to 50lbf, with an input voltage of 5V and an output voltage from 0.5-5V.

The pressure transducers utilized four connections in order to operate; two for the 28 volts in, and two for voltage data out. The load cell uses only three, with a single shared wire that works as a reference for both.

For power supplies, a 28V bench supply was utilized, which provided precisely 28.00V at about .09A. The load cell was powered by a single 9V battery regulated to 5V; with a switch to disconnect the battery to prevent drain when not in use. The battery was checked periodically to ensure it still operated near 9V to maintain a constant voltage.

3.3.8.2 Calibration

Calibration provides the ability to convert the voltage data sampled by the controller directly into pressure and force data via linear relationships (for properly operating hardware):

$$y = mx + b \quad (42)$$

Where x is the input voltage, y is the output data, and m and b are constants dependent upon the equipment used. To find these constants, the sensors were subjected to a known value, and the output voltage recorded; by doing this at two points, the coefficients can be determined. By using additional measurements, the accuracy of the linear regression can be further refined.

The pressure transducers were initially calibrated using a 100 psi (max) air tank, with a calibrated gauge over this range. However, this proved inaccurate for pressures over 100psi, which lead to utilizing

a pressure regulator attached to a very high pressure tank of either helium or nitrogen. This easily allowed maximum pressure values of approximately 400 psi on the transducers.

Load cell calibration was completed using a low friction pulley and low stretch line; known weights were hung from the line which attached to the motor rail, pushing on the load cell. It was important to use the test stand in a testing configuration to account for the losses in oxidizer feed lines, sensor lines and inherent frictional losses.

The calibration process was repeated several times throughout the project in order to further refine and correct the data. To ensure accurate calibrations, many points were collected for each sensor; the load cell was measured to 25lbs utilizing around 20 data points; the pressure transducers were measured from 0-450 psi at 50 psi increments.

3.3.8.3 Interface Cable Connections

A single electronics box for DAQ hardware was built in order to keep everything organized and reduce set up and cleanup time. The box, shown in Figure 54, contains the battery and regulator for the load cell, interface connections for the pressure transducers, and the output connection for main data acquisition shielded cable. Dsub-9 pin connectors were soldered to the pressure transducer wires, and connectors inside the box, to make it easy to connect and disconnect the pressure transducers from the DAQ to minimize setup time. The load cell was attached by a 2.5mm phono jack. The power attachment was through a female banana plug/screw terminal connected to the bench supply.

The cable connection to the DAQ was made from a spare 20ft section of thick shielded wire with eight conductors inside. While the combined sensors only required six, two per sensor.

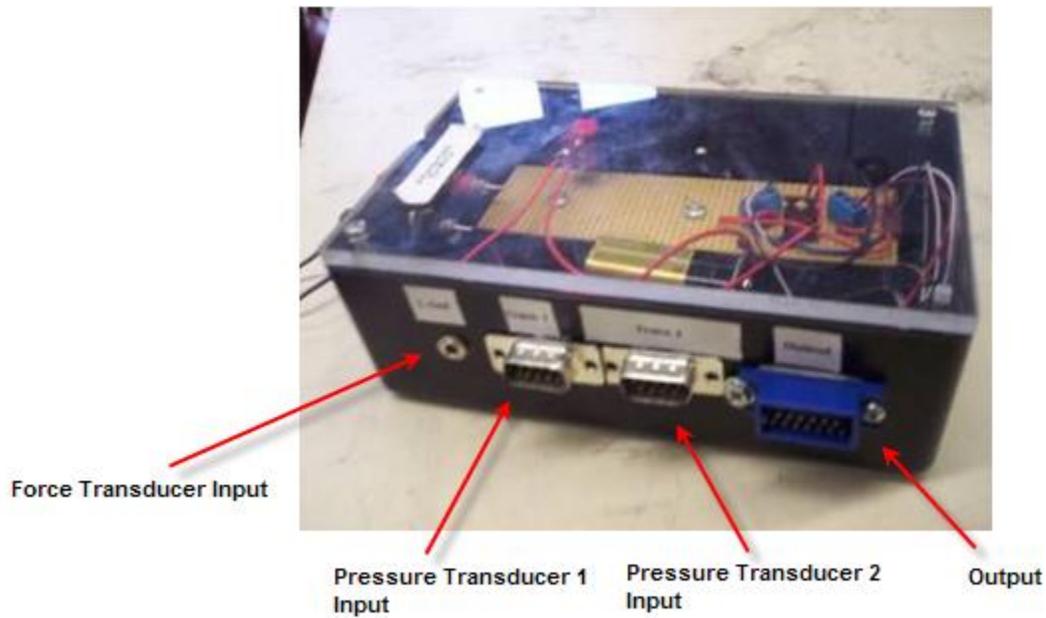


Figure 54: DAQ I/O Box

3.3.8.4 Data Acquisition (DAQ) Equipment

The DAQ chosen was a National Instruments USB-6008, with a 10KHz sampling frequency across up to 4 differential analog inputs. These inputs are multiplexed, meaning that the data is actually sampled one channel at a time, cycling through, rather than all channels sampled at the maximum rate. With three inputs utilized, there is a sample rate of 3.3 KHz per channel. The USB-6008 also possesses 12 digital I/O ports, which later enabled LabVIEW control of the motor operation. The DAQ was mounted on a polycarbonate board with attachments for the analog input and digital output lines. The full final setup for the DAQ can be seen in Figure 55.

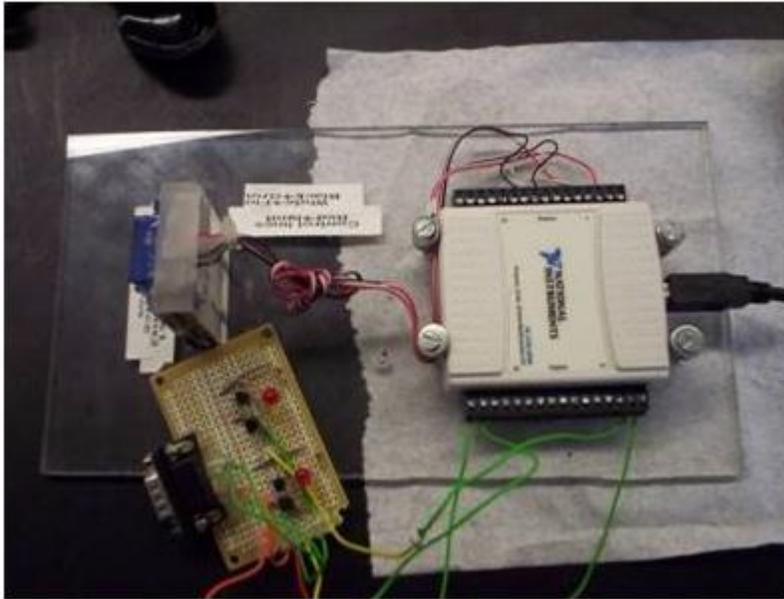


Figure 55: DAQ on connection plate, with connector box cable and digital controller

3.3.8.5 LabVIEW code

The LabVIEW code is what controls the computer side of the DAQ, reading the data and recording it to file. LabVIEW code was specifically created for this project to provide controls that were required for the operation.

The read functions to retrieve the data from the DAQ are basic DAQmx read functions, operating on all three analog inputs. This data is supplied as an array of data structures known as waveforms; these waveforms contain both time and data information combined together for ease of usage. Each waveform contains the data samples from one analog input channel for a short duration of time. By continuously reading the data, the code creates many of these small waveforms. Then, these waveforms are attached together to form a complete waveform array containing approximately thirty seconds of data.

While the data is being attached, the code is performing several other operations. Based on user inputs, the code can convert the data from voltage directly to pressure and force data. The waveform data is also graphed in real time on the VI's front panel, along with displays of chamber and line pressure.

Later versions of the VI also included digital outputs of Boolean data (True/False) through the digital I/O ports on the DAQ. The +5V from the 'true' output of the DAQ was used to control the nitrous solenoid and igniters from within LabVIEW using a series of timers. A waveform is constructed of the Boolean data and attached to the actual pressure and force data so that the ignition and flow triggers could be displayed.

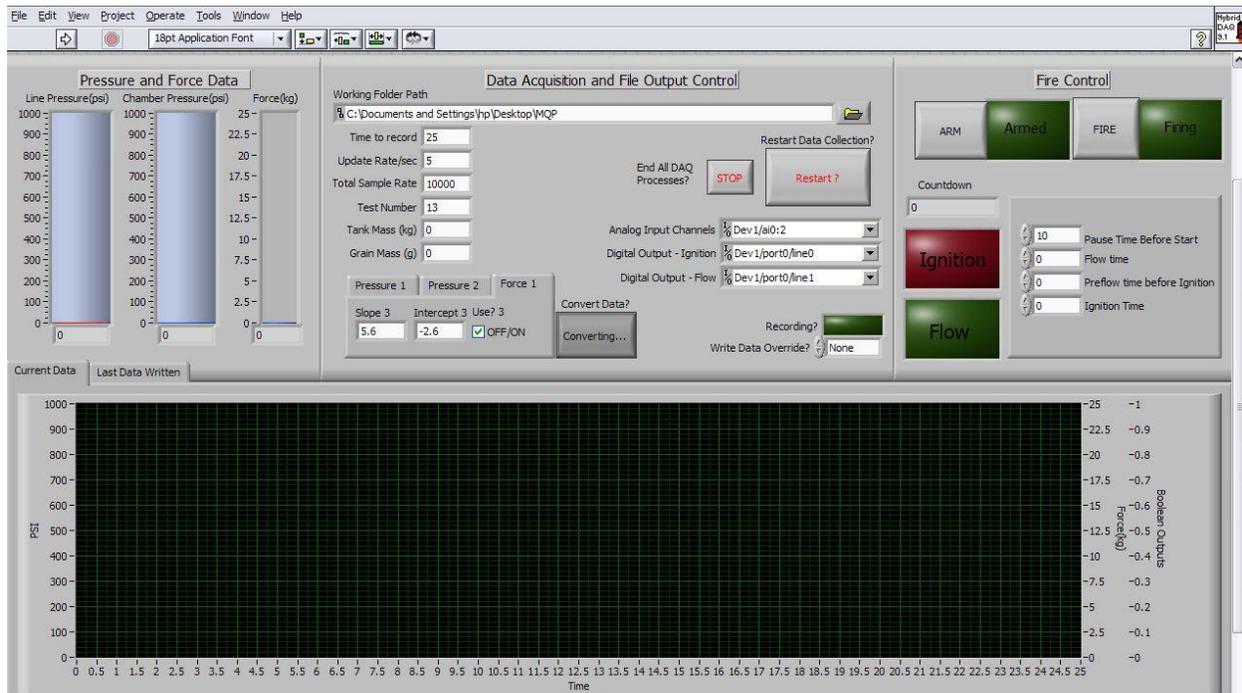


Figure 56: Hybrid DAQ 9.0 Front Panel

After the full thirty seconds of data has been recorded, this data is outputted to a comma-delimited spreadsheet that can be opened in programs such as Excel or MATLAB. The VI automatically generates a time-stamped output file containing the test data. Included with the data is a header that gives the test information (burn time, test number, date, etc) which helps link data to test video footage.

The LabVIEW DAQ/controller went through several iterations based upon feedback generated in testing. In early versions, the sample rate was significantly below the DAQ's maximum capacity. At the time, the code read one sample per channel, wrote them to an Excel file, and then started over again. While this seems like a fairly basic starting point, the slow times associated with file input and output meant that this could only record at a few hundred samples per second, nowhere near the DAQ's capacity of 10 KHz. This first code would also crash if left running on the computer for more than a few minutes, as any of the resources being used by the data collection were not being released from the computer's memory.

The second major iteration of the code was able to sample at 10 KHz, but without any sort of real-time display; thirty seconds of data was recorded, then plotted and outputted to file. The concern with this was that without a reasonable update rate, the potential pressure spikes while the motor was in operation would not be seen.

The final code iterations included the DAQ controller output as well as several controls to make operation more user friendly. One example is enabling the review of the last recorded data after a test had been conducted and direct output of the graph to an image file. Other improvements included front panel organization and file output overrides. The final front panel is shown in Figure 56.

3.4 Final Test Vehicle

A final flight vehicle was designed to satisfy a set of requirements as described in the following section.

3.4.1 Mission Criteria

The overall objectives for the final flight vehicle consisted of the following:

1. Capable of flying an operational Flight Data Acquisition System to record fin flutter
2. Use an onboard flight computer to record flight data and deploy recovery system
3. Maximum installed impulse will not exceed 640 N·sec
4. Capable of attaining minimum altitude of 1500 ft.
5. Allows for interchangeable test fins

The first objective pertains to the system developed as described earlier in order to record strain gauge readings from oscillating fins attached to the rocket. This system was developed so that it was both small and lightweight relative to the rest of the rocket. This objective also includes recovery of both the device itself and the readings which the device acquired during flight from the strain gauges.

The second objective goes hand and hand with the first objective, namely recording data about the rocket's flight including velocity, acceleration, height and time are needed in order to interpret the readings collected from the strain gauges. With this rocket data, it should be possible to draw some conclusions about fin flutter onset velocity during model rocket tests. The computer will also be used for deployment of the recovery system for the rocket.

The third and fourth objectives were more for safety than for scientific purpose. Through mutual agreement, the team's advisor, Prof. Blandino and WPI's administration have agreed to limit the total impulse to a value no greater than 640 Ns. An altitude of 1500 ft was chosen so that the rocket could be safely recovered after launch and not drift too far away from the initial position.

The final objective reflects the fact that this rocket was mainly designed in order to test fin flutter during flight. Allowing for interchangeable fins on the rocket allowed testing of different shapes and sizes of fins so that there was adequate data to analyze and draw conclusions from. The sets to be tested

will also include a set of control fins that should not flutter under the rockets intended speed to make sure that there are not other factors coming into play with these tests.

3.4.2 Design

A main goal of this project was to design and build a rocket that could accommodate an on-board DAQ capable of measuring strains on the fins. Along with being able to house a FCDAQ, the rocket also needed to have the ability to inter-change the fins with relative ease. By doing so, it would make it easy to obtain data for several different fin geometries. This section will discuss the design process used to achieve these goals.

3.4.2.1 Fin-Can

For the team to have the ability to quickly change fin sets, a fin-can was designed to accomplish this task. Several design ideas were brought before the team for consideration. The two main designs that were considered were a design in which the fins would slide into a dove-tailed groove and then held in place using a bottom retention plate. The other design that was considered was to build multiple fin-cans that could be easily attached and removed from the rocket body. Each fin-can would have their own distinct fin sets permanently attached.

The design that was chosen for the final rocket was the multiple fin-can design. The team chose this design because it replicates how fins are usually attached in model rocketry. The main design of the fin-can is nearly identical to the fin-cans built for the initial test vehicles. The main difference was that instead of building the fin-can into the lower body tube, they would be their own separate component. The fin-can would attach to the DAQ housing using a coupler tube that would join the two sections. The use of a coupler tube is what allows for quick changes of the fin-can segments.

It was also determined that by using this design, the wiring for the strain gages would be easier and more efficient. By using fin-cans with the fins permanently attached, the wiring for the strain gages could also be permanently attached. This design would allow the team to pre-wire the strain gages and properly secure all the wiring internally. The method also aids in the ability to quickly interchange fin-cans. With wiring already in place, to change a fin-can would be a two step process; remove the existing fin-can and disconnect the current strain gage wires from the DAQ, then attach the new wires to the DAQ and secure the fin-can to the DAQ housing. A cross section of the fin-can design can be seen in Figure 57.

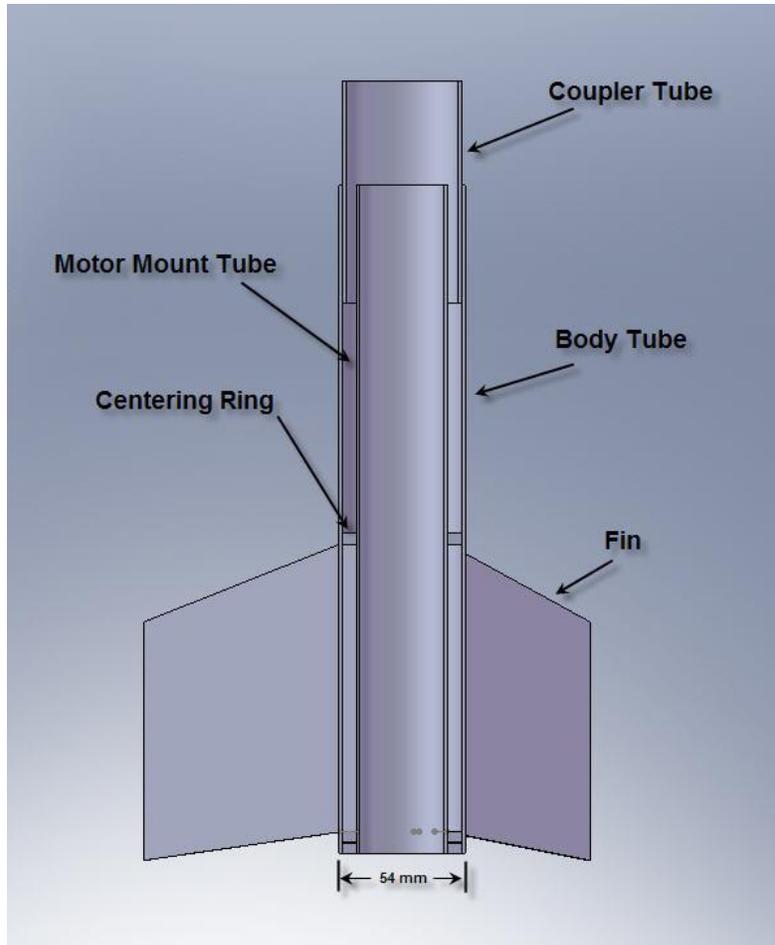


Figure 57: CAD Design of Fin-can for Final Flight Vehicle

3.4.2.2 Electronics Bay

The upper electronics bay was designed to hold the MC-2 flight computer, an on-board camera, and two 9-V batteries. One main goal of this design was to keep the total weight to a minimum. The main component of the electronics bay was the sled that the components attached to. The goal was to design a sled that was both very light and still maintains structural integrity to hold its components. First, the components were laid out to decide how they should be arranged inside the rocket. The most optimal placement was to attach the flight computer to one side of the sled, the two batteries to the opposite side of the sled, and to attach the camera using a slide through mount with a set screw to hold the camera in place. The design can be seen in Figure 58.

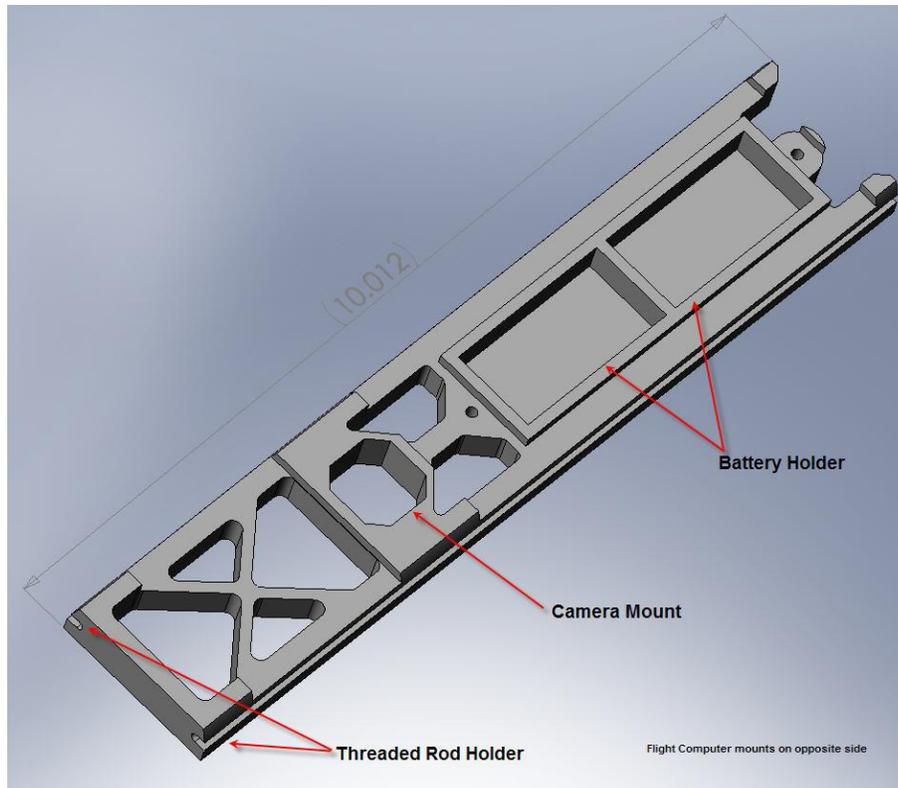


Figure 58: Electronics Bay Sled Design

Once the basic arrangement was laid out, modeling of the sled was performed using SolidWorks. The design called for two holes for the flight computer to mount to. On the opposite side of the sled, two battery holders were created. The holders were placed in between the two mounting holes for the flight computer. By doing this, the same holes could be used to mount the computer and to place a retaining strap over the batteries to secure them in place. After the mounting points had been designed, unnecessary weight was removed from the sled. The next task was to determine how the sled would be held into the coupler tube. This was accomplished by putting a 0.125 inches channel into each side of the sled. A threaded rod would slide into the channels properly securing the sled to the coupler. Once the design was completed, the sled was manufactured from a piece of 0.5 inch thick Delrin sheet. The manufactured part can be seen in Figure 59.

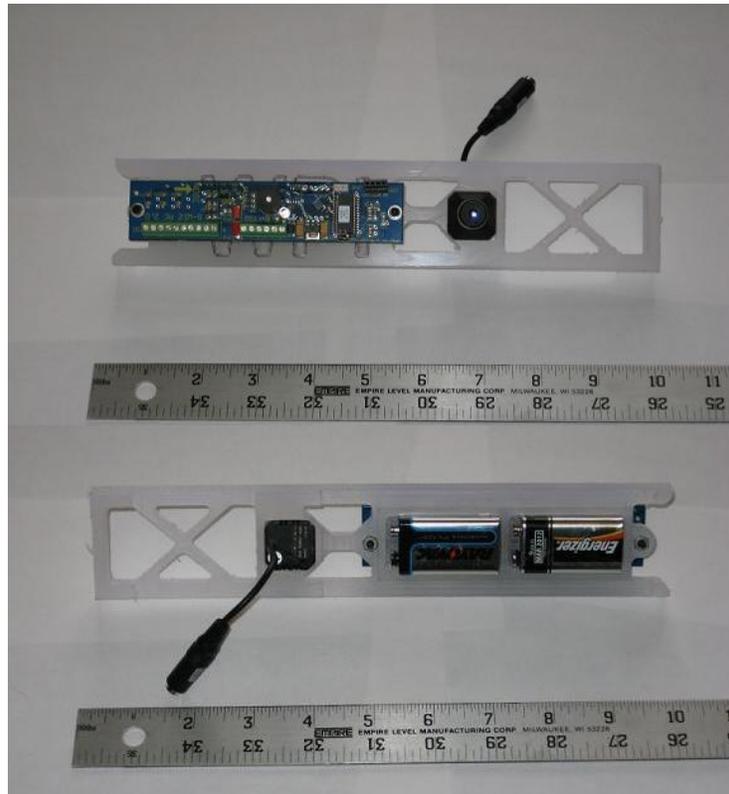


Figure 59: Assembled Electronics Bay Sled

3.4.2.3 Main Body

In order to produce measurable fin flutter, the rocket was designed for maximum velocity. To help achieve the highest maximum velocity, the rocket was designed to be as thin as possible. The widest component in the rocket was the DAQ. Using the width of the DAQ as a starting point, the thinnest body tube that could accommodate it was a 2.15 inches diameter tube. With the body tube diameter now determined, the next factor that was taken into consideration was the geometry of the fins. Using RockSim, a base model of the rocket was designed that did not include the fins. Three fin geometries were chosen: one that should produce significant flutter, one that might produce flutter, and one that should produce no flutter at all.

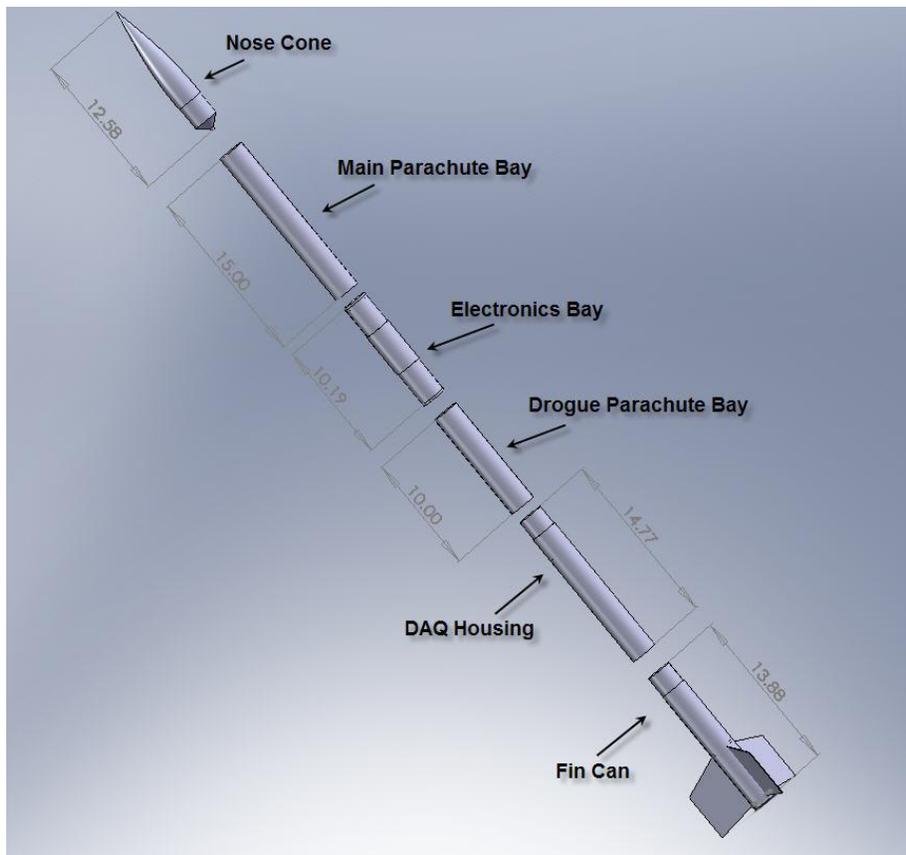


Figure 60: CAD Model of the Final Flight Vehicle (Dimensions in Inches)

3.4.2.4 Fin Geometries

Five common fin geometries were chosen as to test in the wind tunnel, but only three were chosen to be used on the final flight vehicle. It was determined that three designs had a significant flutter velocity, because it was recorded in LabVIEW and the vibrations could be seen and recorded on film. These three geometries were clipped delta, square and tapered swept. These three designs were tested in the wind tunnel with plywood and G-10 fiberglass. In order to have wind tunnel tests simulate the flight vehicle, scaled down versions of those tested in the wind tunnel were purchased to put on the final rocket. The material chosen for the three types of fins was 0.063 inches G-10 fiberglass.

3.4.3 Data Acquisition System

The goal of the FCDAQ is to collect and store the strain gauge data at an adjustable sampling rate between 1 and 10,000 Hz. This data must then be uploaded in the form of a text file. However the format of this txt file must be the same as a comma separated variable file. The entire package must be able to withstand the G-forces of launch and be self-powered by a 9 volt battery. It must also be self-triggered at take off, and record at least 7 seconds of data.

3.4.3.1 DAQ Hardware

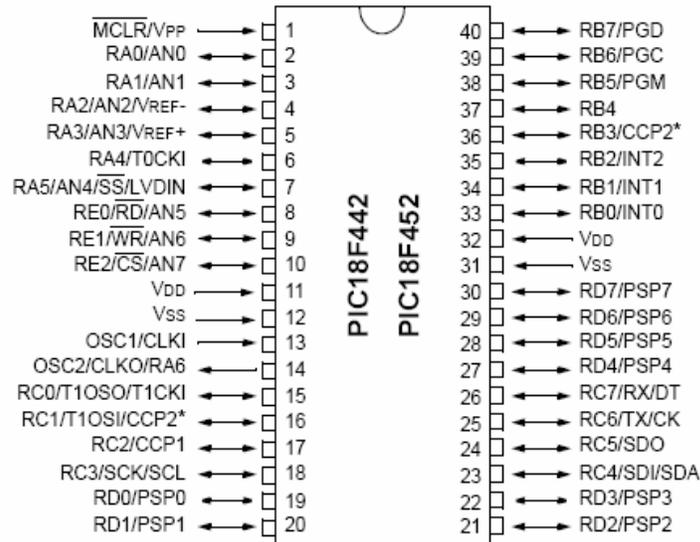
The goal of the onboard data acquisition system was to collect seven or more seconds of data from each of the strain gauges mounted on the fins. This information was collected with at least 2.5 times the Nyquist frequency (see section 3.2.6.2 on Shannon sampling theorem for more detail) of the predetermined oscillations. This data was stored in an onboard memory unit and able to communicate with a ground based computer via an RS 232 link. The power source for the entire system was a 9 V battery.

This system was a designed in a gradual process. As data from the previously described set ups was acquired in the wind tunnel, the onboard system was designed to perform the same functions as the system tested in the wind tunnel setup. The initial estimates were for a flutter speed of 5 kHz.

The next decision was power consumption; most of the microprocessors on today's market require 5 V for proper performance, as do most of the timing, and amplifying electronics. For this reason a voltage regulator needed to be acquired. The LM7805 was readily available and is compact enough to accomplish this task. It requires minimal circuitry to filter the power and is reliable. With this regulator, a 5 V source was now on-board.

Armed with this information and a now stable 5 V power source, the brain of the system was chosen. The PIC 18F452 series microprocessor was selected for size, weight, power, amount of I/O ports, and assembly language instruction set. This chip controlled all of the functions for the onboard data acquisition system. Figure 61 is a pin-out diagram for the selected chip. Seven analog inputs are required for the strain gauges. Port B will be configured for programming the chip, along with the external clock input. All of Port C will be configured as data lines, and Port D will be configured for the address counter. Finally Port E will be used to make the correct timings for the read and write cycles of the memory chip.

DIP



Note: Pin compatible with 40-pin PIC16C7X devices.

Figure 61: PIC 18f452 Pin Layout [28]
© 2008 Microchip Corporation

The next step in the design was to choose a suitable memory chip. The requirement for this chip was a writing speed on the order of micro and if possible nano-seconds per cycle. This was necessary in order to not be limited in the amount of data the system could collect in such a short period of time.

Another consideration was the need to be as lightweight and compact as possible. In case of a rocket failure, the information on the chip should be held when there was no power and be able to be recovered if everything but the chip was destroyed. There was also the limitation of matching the amount of input lines to the amount of pins dedicated to data collection on the microprocessor. For this reason the TI BQ4015ma-70 was chosen. The read and write cycles can be performed in as little as 70ns. This chip also possessed a self contained battery back-up for unexpected loss of power. The stored data on the chip could be saved for up to five weeks without power. The complete specifications can be found from the datasheet. [5]

Now that the memory and the brain of the system were both selected, the code had to be written for proper memory allocation. With the ports chosen for memory use, and the external clock speed set with a 20MHz timer, a process was derived to recover the information from the memory chip. RS 232 serial communication was chosen, both for the minimal parts required, and its ease of programming.

One issue with serial communication is that it requires 12 volts; however it is a common method for data delivery. The Maxim 232 chip along with accompanying capacitors solved this problem. The

Maxim 232 chip was designed to take the 5 volt output of any microprocessor and output the same result at 12volts to a serial communication line. With the given clock speed, the data was written to the Windows based computer at 9600 baud. The circuitry for the communication was then added to the schematic.

The only thing that was left was to design and build the amplifier board and the proper Wheatstone bridge setup. For this the TI 2272 dual operation amplifier chip was used. This compact chip, combined with a few potentiometers, allowed for an adjustable gain to properly amplify the signal and not overpower the microprocessor. The voltage divider was added to create an adjustable reference voltage that was set to 2.5 volts.

Since all the pieces were selected, the components for the entire setup were placed on a breadboard for testing. Once testing proved the entire system was operational, the system layout was compacted to 2.5 in wide, by 8.0 in long and soldered to a perforation board.

The final testing procedure consisted of an identical test with the onboard system replacing the previous system in the wind tunnel. The test mimicked the same speeds, with the same gain. Once the data was recorded, it was analyzed and compared with the data gathered from the previous set up. If the information matched, the system had successfully been double checked and deemed flight worthy.

The objective of the code is to allow the PIC 18F452 to act as the brain of the entire operation. To successfully accomplish this, a series of subroutines coupled with one main loop were written. This code must combine a non-volatile memory chip with built in data retention battery that is coupled to a 20MHz microcontroller, which allows it to acquire strain data from two places on each of the 3 fins.

Because the frequency will change with each fin set, it was measured in previous wind tunnel testing to have natural frequencies of approximately 7-40 Hz. Because of this wide range of expected frequencies, the code was designed with an easily adjustable acquisition rate between 1Hz-10 kHz.

The data is acquired on all six strain gauges in each acquisition pass. This data is then stored to six pages of memory with 65535 data points each. The entire acquisition is started when an acceleration sensor is triggered at takeoff. When recovered the data will be translated by means of an RS232 serial connection transferring the data at 9600 baud to a Windows computer where the signal can be analyzed.

3.4.3.2 FC-DAQ Code

The entire code is written in assembly language and can be found in Appendix F: FCDAQ Software Coding. The code begins with a routine that sets all the ports as a proper input or output. It

then defines all the variables used in the subroutines. This leads to an initialization routine, which sets the active bits on each pin, defines the clock setting, and sets all the variables to zero.

Once this is accomplished, the program will enter its main loop. It will remain in an infinite counting loop until the sensor that detects takeoff is activated. Two things will happen at take off, a main count loop will be activated and will read the analog output voltage of each strain gauge and then store them in their own individually assigned variables.

The processor will then take the variables and store them into the memory location that matches there assigned number based off the main count loop. This process is then repeated until the main count loop times out, at 524288 or 2^{19} . When this happens, a flag registrar will be triggers that indicate that the memory is full.

Once this flag is set, the FC-DAQ must be restarted by disconnecting and reconnecting the 9 Vbatteries. A button then triggers the program to enter the aptly named “waitforserial” routine. This will start the transfer of information over the serial port and into windows “hyper terminal.” The data can then be exported into Excel, and saved as a comma separated file. This can then be analyzed using LabVIEW or MATLAB.

4 Results and Analysis

The following sections detail the results and analysis obtained by the team in three groups. The first was the initial test vehicles which were each launched two times. This is followed by aerodynamic and hybrid propulsion system results and analysis.

4.1 Initial Test Vehicles

The overall goals for the initial flight vehicles were sufficiently met. Two rockets nicknamed “wicked awesome” and “wicked sweet” were designed and built. There were two separate launches, the first on November 3, 2007 and the second on December 15, 2007. After completing these test launches the group felt comfortable designing, building, and launching a model rocket.

After the launches were completed the results were analyzed. In both launches the rockets were successful; both rockets reached the objective altitude before the recovery system was successfully deployed and the rockets were recovered. Both rockets utilized the MC2 flight computers for in-flight triggering of ejection charges and data recording (see WARRIORS I, Appendix G for MC2 information).

For the initial flight vehicle nicknamed “wicked awesome”, the flight computer was successful in both flights. However, the parachute was not fully deployed in either flight. Even though the parachute

was not deployed, the vehicle remained intact. The flight computer successfully recorded data on both flights. On the second flight, an on-board wireless camera was used to record in-flight video.

The first launch on November 3, 2007 was in Amesbury, MA with the Central Massachusetts Space Modeling Society (CMASS). The results from the flight computer include: flight information, flight summary, flight configuration, overlay graph, altitude graph, acceleration graph, integrated airspeed graph, and an integrated altitude graph. The basic flight set-up which is provided by the user is displayed in the flight information table when the flight computer is connected to a computer. Table 3 displays the information that was obtained from the first flight.

The same information was also obtained for the second flight, displayed in Table 4

Recorded Max Acceleration: 409.19 f/sec/sec	Recorded Max Speed: 208.89 f/sec
Recorded Max Altitude: 500 f	
Maximum Altitude (barometric): 500.23 f	Maximum Altitude (integrated): 379.43 f
Maximum Airspeed: 135.91 f/sec	Maximum Acceleration: 94.58 f/sec/sec
Maximum Mach: 0.12	Maximum Acceleration: 2.94 G
Altitude of Max. Airspeed: 122.33 f	Altitude of Max. Acceleration: 42.41 f
Time to Mach: 0 sec	Altitude of Mach Trans.: 0 f
Time to Apogee: 5.52 sec	Flight Length: 5.79 sec
Time to booster burnout: 1.7 sec	Time to 2nd Stage burnout: 0 sec
Altitude of Booster Burnout: 126.44 f	Altitude of 2nd Stage Burnout: 0 f
Descent Rate: 0 f/sec	
This computer has been calibrated	

Table 3: Flight Summary Information for First Flight of “Wicked Awesome”

Recorded Max Acceleration: 582.42 f/sec/sec	Recorded Max Speed: 555.09 f/sec
Recorded Max Altitude: 2543 f	
Maximum Altitude (barometric): 2799.6 f	Maximum Altitude (integrated): 4772.44 f
Maximum Airspeed: 561.52 f/sec	Maximum Acceleration: 588.4 f/sec/sec
Maximum Mach: 0.5	Maximum Acceleration: 18.29 G
Altitude of Max. Airspeed: 609.36 f	Altitude of Max. Acceleration: 2081.51 f
Time to Mach: 0 sec	Altitude of Mach Trans.: 0 f
Time to Apogee: 19.06 sec	Flight Length: 58 sec
Time to booster burnout: 1.88 sec	Time to 2nd Stage burnout: 5.33 sec
Altitude of Booster Burnout: 626.35 f	Altitude of 2nd Stage Burnout: 2091.5 f
Descent Rate: -47.86 f/sec	
This computer has been calibrated	

Table 4: Flight Summary Information for Second Flight of “Wicked Awesome”

The second rocket was launched in Sterling, CT on December 15, 2007 at the CATO launch site. The flight information was not displayed in the flight information tab when the data from the MC2 computer was hooked up to a computer and read. The motor for this launch was an I211; the length, maximum diameter and takeoff weight were the same. This motor is much more powerful than the G75 which is why the rocket had a faster launch speed and obtained a higher altitude.

The graphs displayed in Figure 62 and Figure 63 show the altitude vs. time graphs for each of the two flights. The x-axis represents the time and the y-axis is the altitude.

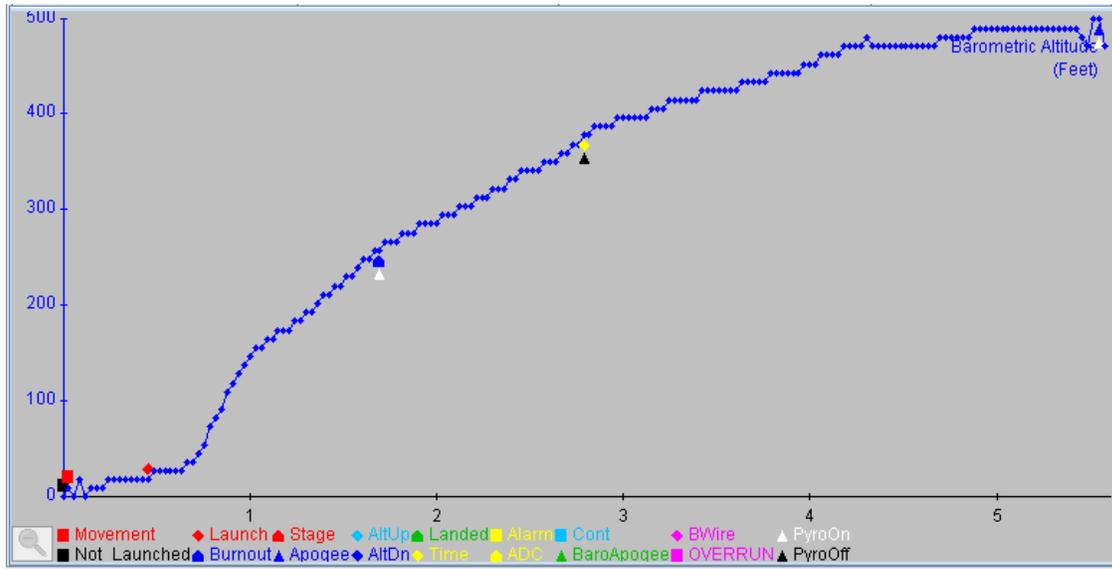


Figure 62: Altitude for Initial Flight Vehicle "wicked awesome" flight 1

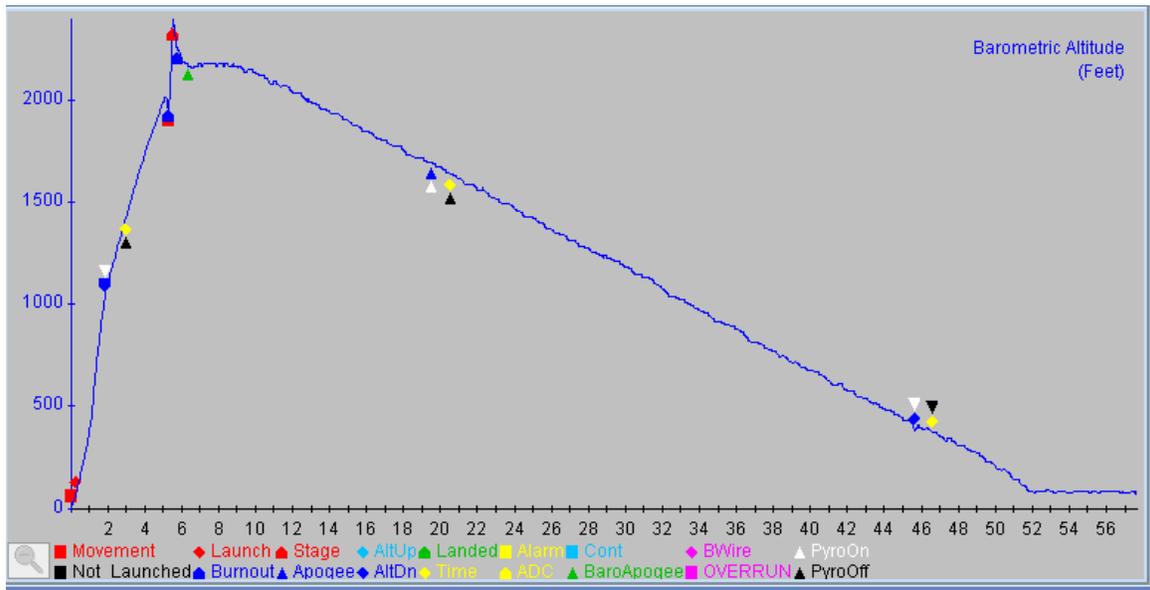


Figure 63: Altitude for Initial Flight Vehicle "wicked awesome" flight 2

From the flight summary, it was determined that the time to the 500 ft apogee was 5.52 seconds. Unfortunately, the flight computer stopped recording after this point, and all descent data was lost. This was due to the battery being disconnected by heavy forces during launch and parachute deployment.

For the second flight the entire flight was recorded. The flight lasted a total of 58 seconds and reached a maximum altitude of 2799.6 ft. This flight was successful in achieving the goal of designing and building a rocket that could obtain an altitude of 1500ft.

Both Figure 64 and Figure 65 display the graphs of the acceleration during the rocket's flight. Again the x-axis displays the total flight time and the y-axis is the acceleration in ft/s^2 . The first flight had a consistent acceleration, but the second flight was erratic. Overall, these graphs are useful to determine the overall acceleration throughout the flight of the rocket.

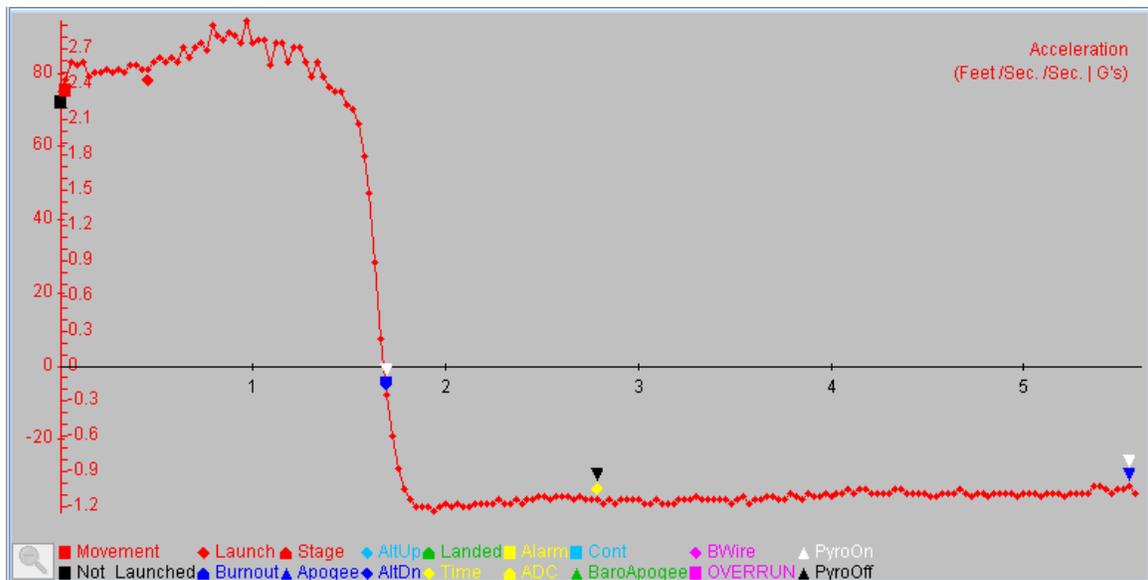


Figure 64: Acceleration Graph of "wicked awesome" flight 1.

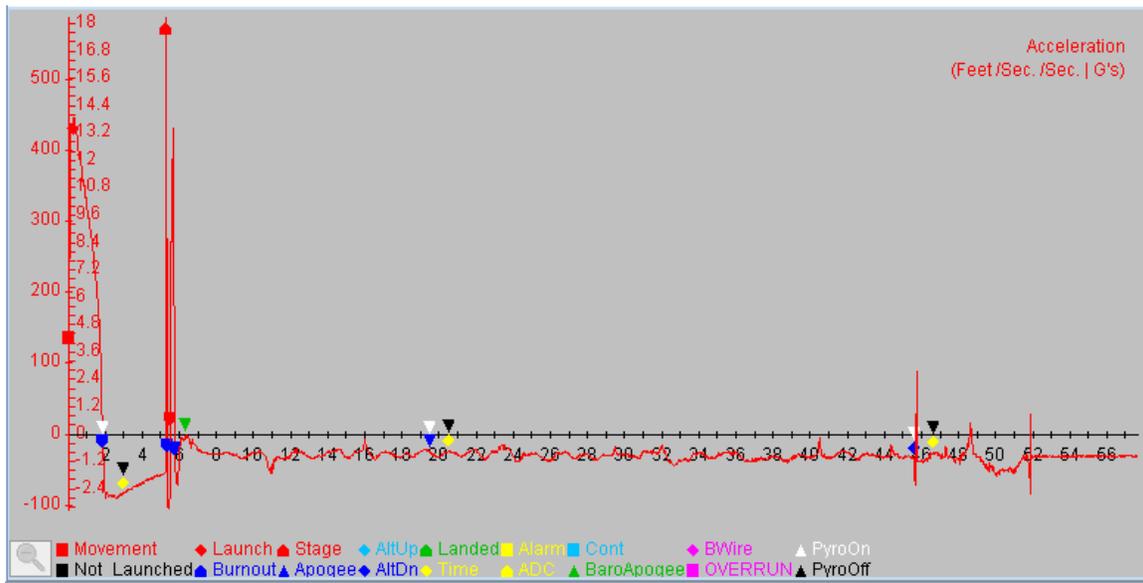


Figure 65: Acceleration Graph of "wicked awesome" flight 2.

For the first flight of “wicked sweet”, the parachute was deployed using ballistic ejection because the flight computer did not work. To understand the problem, the computer was tested in the lab after the flight. The testing was completed by running the flight computer through a diagnostic using the prescribed procedure in the manual. It was then set up for the next launch.

In the second launch the computer again did not work properly. Therefore, the decision was made to not use that computer for the final flight vehicle. With more testing in the lab, it might have been possible to debug the problems. Only one flight computer is needed for the final flight vehicle, the one that was successful both times was selected for use. The initial testing successfully allowed the team to determine which flight computer would be more suitable for the final flight vehicle.

Flying the initial rockets also provided an opportunity for the team to learn how to interpret the data gained from the MC2 flight computer that will be used on the final flight vehicle. The initial flight vehicles proved that an altitude of 1500ft can be obtained while remaining under the total impulse limit. The team successfully learned how to design, build, and launch a high power model rocket.

4.2 Aerodynamics

This details the results found from the CFD analysis, MATLAB modeling of equations, wind tunnel testing and the flight capable data acquisition system.

4.2.1 2D CFD Analysis

Utilizing FLUENT, 2D analysis was performed on the leading edge and trailing edge of the rocket fins to determine whether one design fin would be better than another. A parabolic leading edge

with a square trailing edge fin and a fin with parabolic leading and trailing edges was chosen. After meshing the 2D objects in Gambit with around 860,000 flow nodes, the objects were imported into the 2D FLUENT software. The velocity of flow over the fin was determined to be about 224 m/s or 500mph from the estimated rocket velocity on the final test vehicle. When choosing conditions of the flow air was chosen as the fluid with the properties of sea level and standard temperature. After running FLUENT for 150 iterations, for both the parabolic leading edge (LE) fin and the parabolic leading edge trailing edge (LETE) fin, there was a clear distinction between the two fin designs.

Observing the contour plot of static pressures for the two designs shows a clear difference. A static pressure plot shows the pressure at a given point in the flow, which is useful when determining pressure build up on an object. The static pressure on the LETE fin is slightly higher in value than that of the LE fin. Figure 67 shows that the pressure build up is off the fin itself. Figure 66 shows that on the LE fin there is significant build up on the upper front parabolic leading edge. The pressure build up is also fully detached on the LETE fin where as on the LE fin the pressure is built up on the surface of the fin producing pressure moments on the fin.

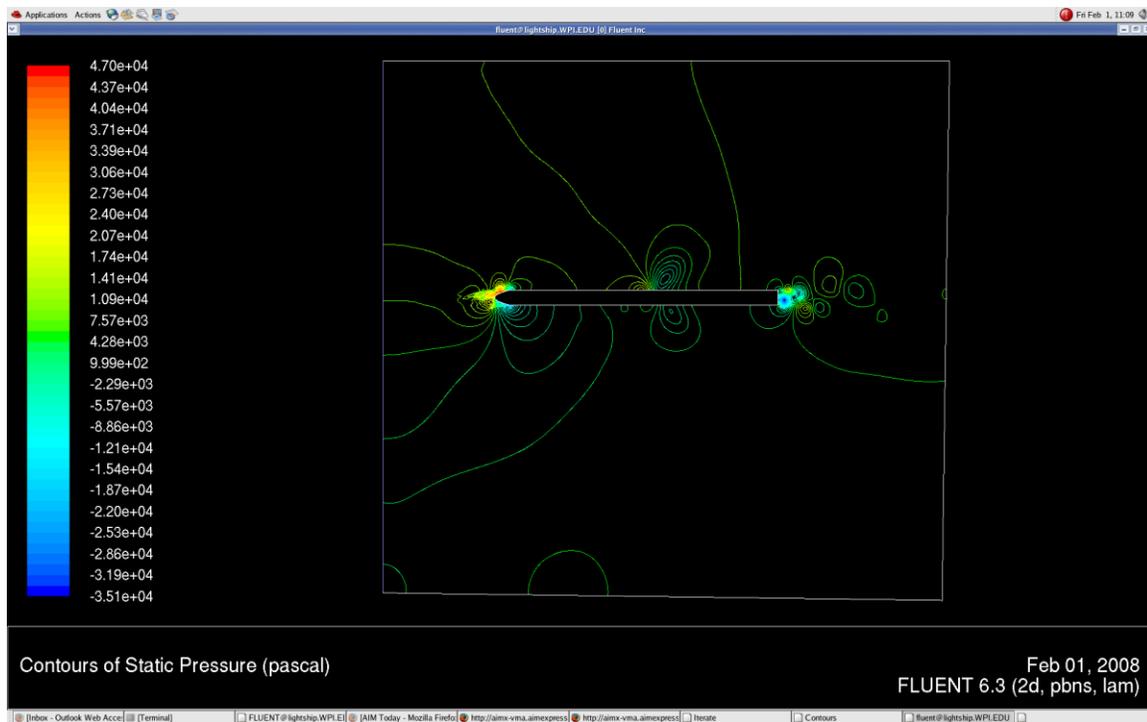


Figure 66: Static Pressure LE

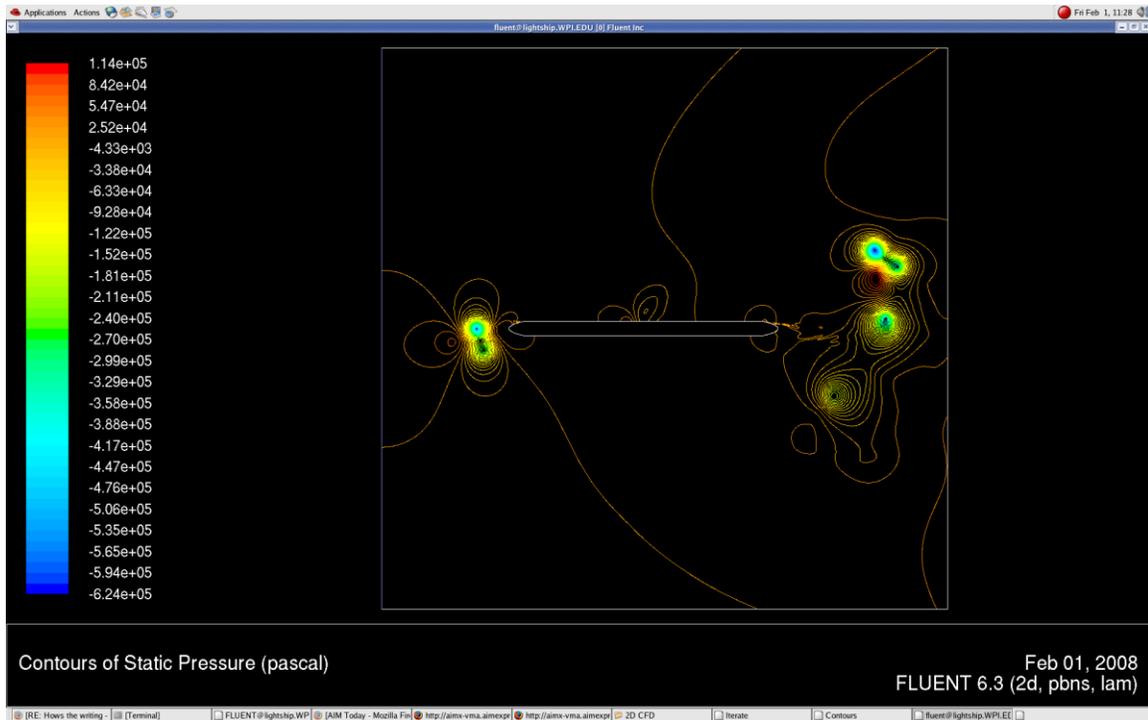


Figure 67: Static Pressure LE TE

In Figures Figure 68 and Figure 69 the dynamic pressure is shown for the two designs. It can be seen that there is significant surface pressure build up on the LE fin versus a much more unaffected LE TE fin. Dynamic pressure was investigated because it is closely related to the kinetic energy of fluid particles. The dynamic pressure contours of the LE TE, show higher pressure contours before and after the fin. This can contribute to a more turbulent flow over the fin. This result was not expected in initial analysis. Still the LE fin is more directly affected by pressure than the LE TE fin.

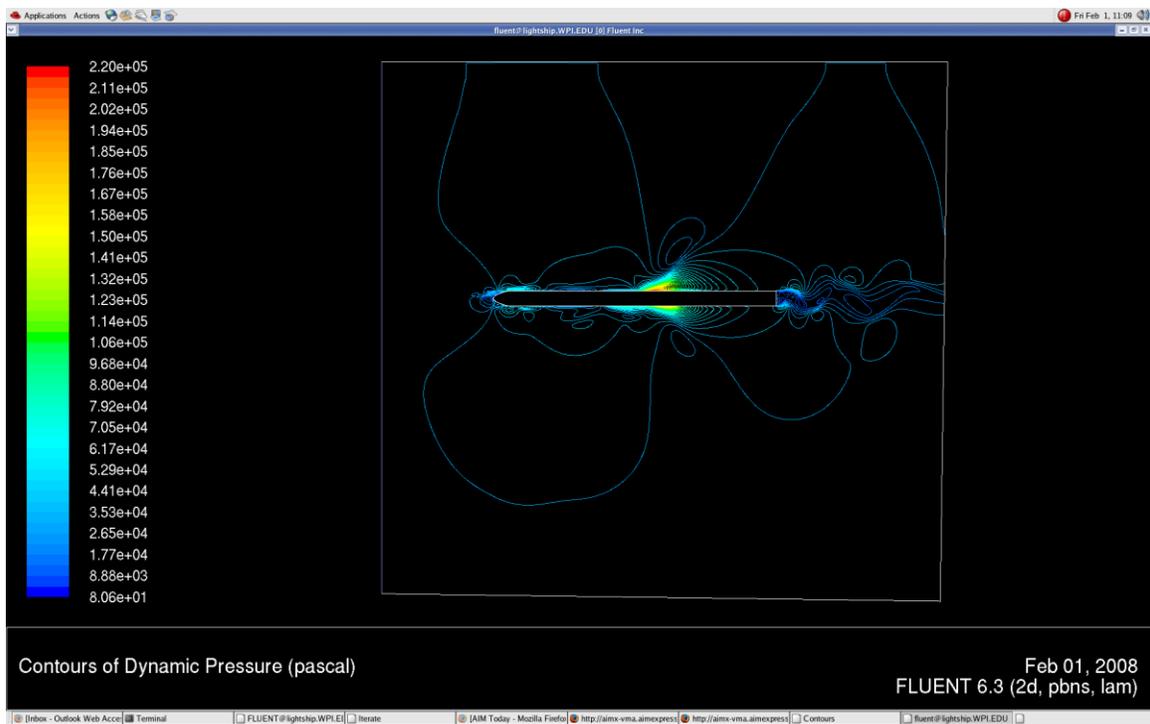


Figure 68: Dynamic Pressure LE

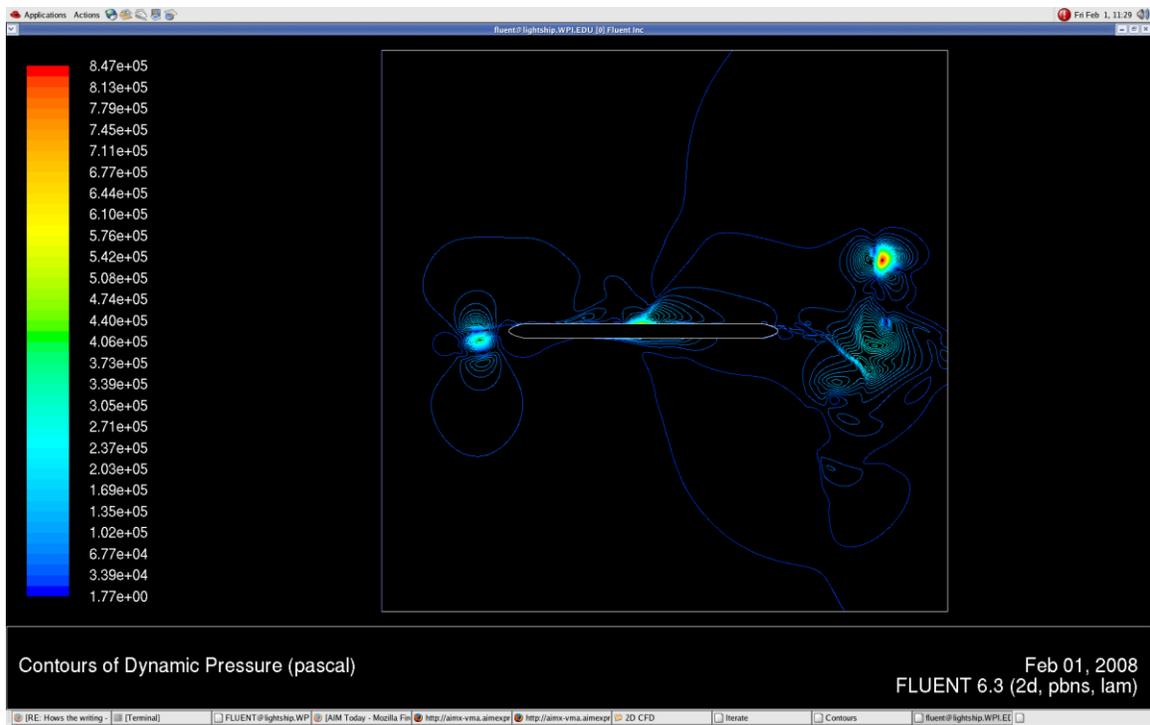


Figure 69: Dynamic Pressure LE TE

In the case of stream function contours, laminar flow over the LE fin and turbulent flow before and after the LETE fin was seen. This contributed to an unstable fin when parabolic shaping for the leading edge and trailing edge was used. The LETE fin-can lead to turbulent flow due to the fluid trying to reach equilibrium. The fluid must flow over two parabolic shapes that return to the same point in the flow behind the fin this causes flow instability.

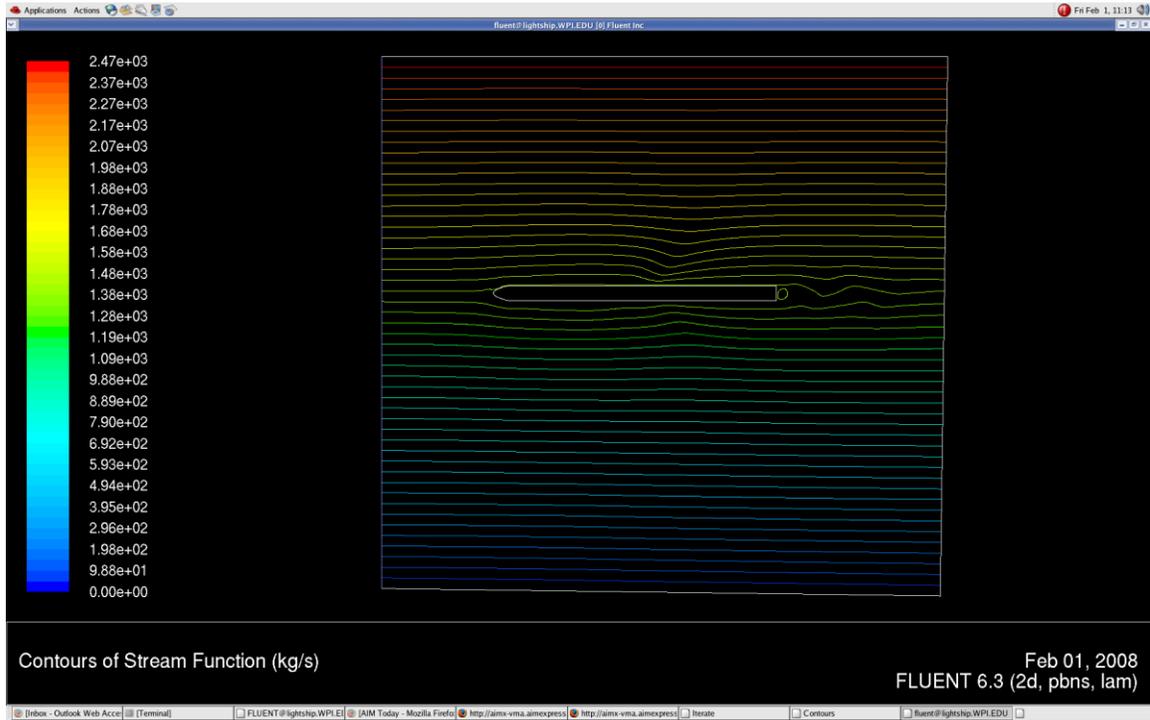


Figure 70: Stream Function LE

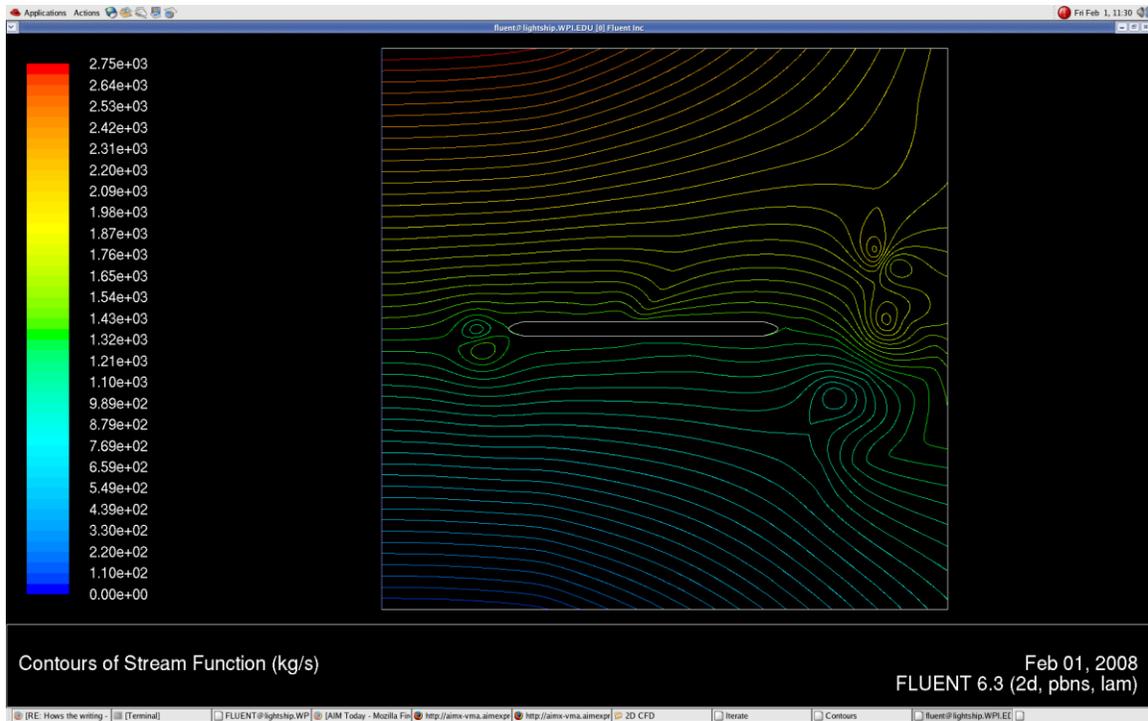


Figure 71: Stream Function LE TE

The contours of velocity magnitude show an unexpected result. It appears that the velocity over the LE TE fin is nearly twice as fast as the LE fin. Figure 72 shows a more symmetric velocity magnitude over the fin where as the LE TE fin shows a slight unbalanced velocity magnitude. The contours in Figure 73 show leading and trailing velocity swirls for the LE TE fin.

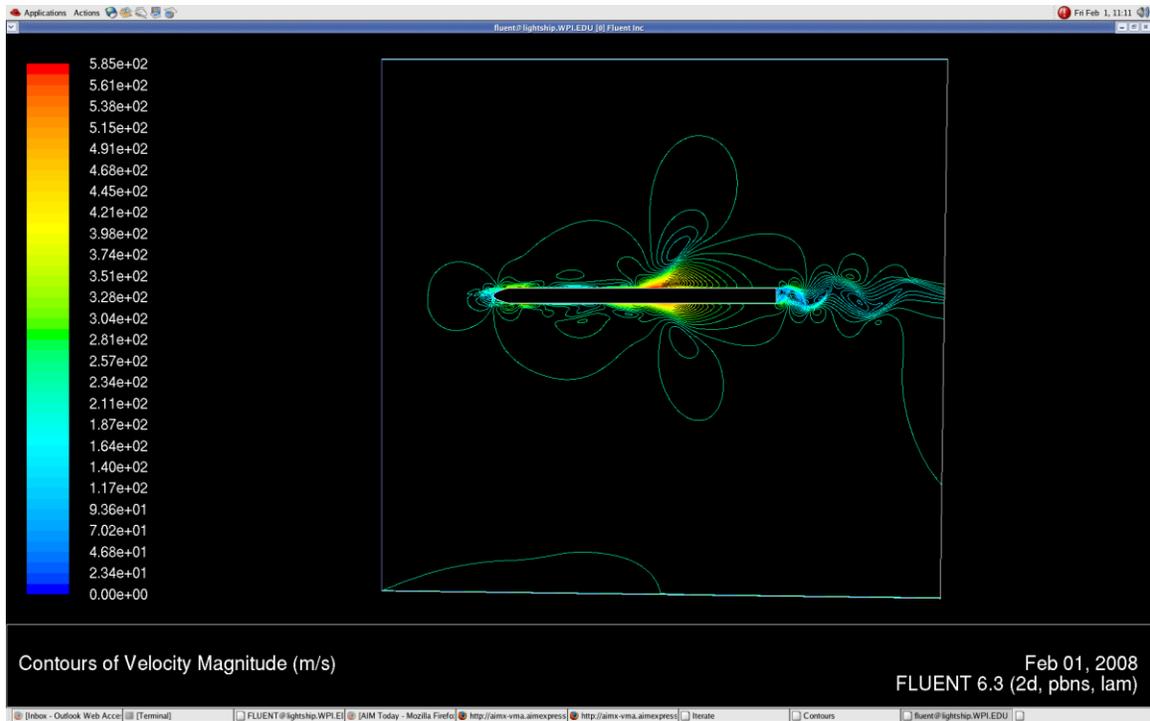


Figure 72: Velocity Magnitude LE

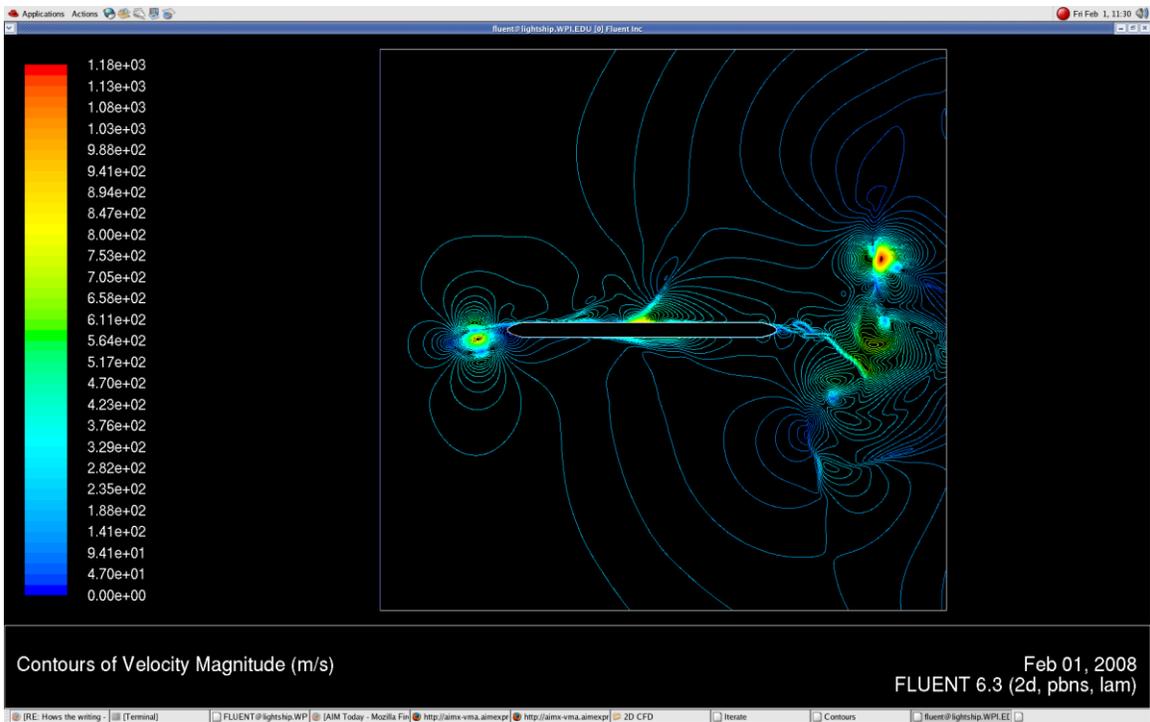


Figure 73: Velocity Magnitude LETE

From the vorticity magnitude contours found in figures Figure 74 and Figure 75, it was determined that the trailing edge of the vorticities on the LETE fin was significantly less than that of the LE fin. Vortices on the leading edge of the LETE fin were significantly less than the leading edge on the LE fin. Vorticities on the LE may be due to fluid shear stress. These contours were not surprising due to the more stream line nature of the LETE fin.

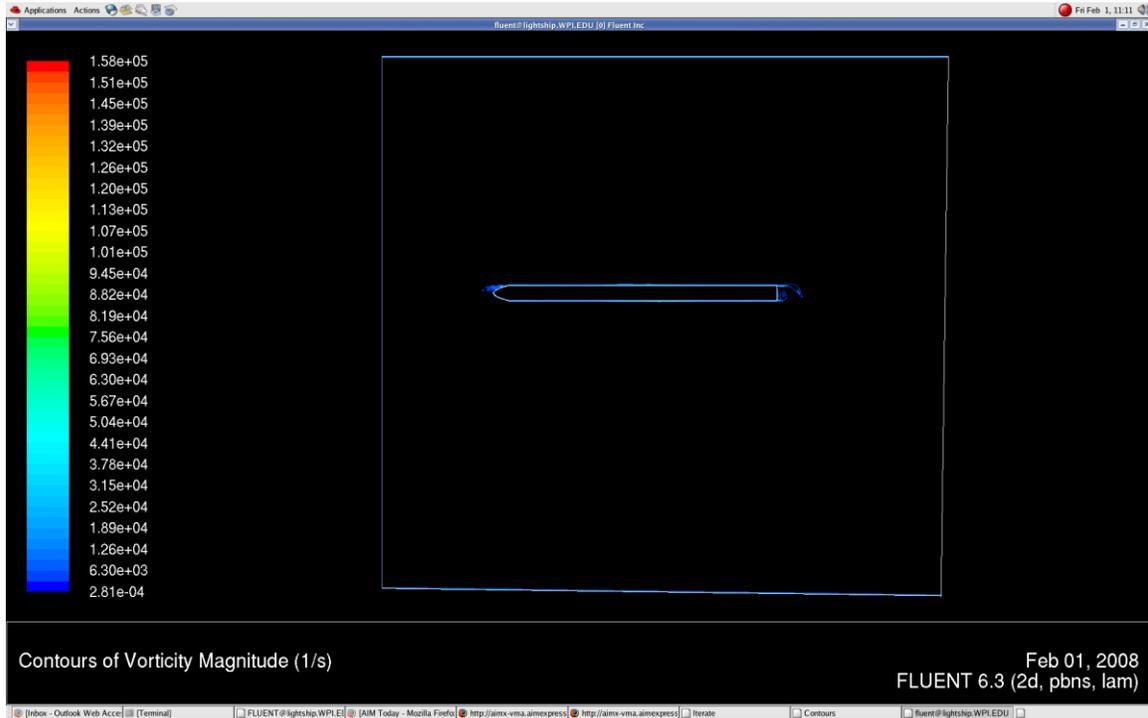


Figure 74: Vorticity Magnitude LE



Figure 75: Vorticity Magnitude LETE

The implementation of 3D CFD analysis of fin geometries led to no useful conclusions. The fin thickness is small in comparison to the velocity flow field that it produces laminar 3D flow. There are no substantial changes in the analysis utilizing 3D flow rather than the already shown 2D flow contours. With these laminar flows produced by FLUENT and with all the fins having very similar surface geometries, the conclusion that we can draw from 3D flow analysis is very insubstantial and very inconclusive for the fins.

4.2.2 Mathematical Modeling of Fin Flutter

The equations used in MATLAB, yielded a flutter velocity higher than FinSim. The error of the results of these programs is unknown because flutter velocity was not empirically tested.

Obtaining the natural frequency of particular fin geometries proved to be too difficult to model in MATLAB. Because of the credibility of the FinSim program, it was deduced that the values computed were acceptable.

4.2.3 Wind Tunnel Testing

Wind tunnel testing was completed on two different fin materials. The first material used was 0.25 inches plywood; this was used because it was readily available and inexpensive. Also the plywood could easily be cut to the correct geometry using a band saw. The second material was 0.063 inches G-10 composite fiberglass sheet. This material was laser cut to the correct geometry by an outside source. It was used because it is a common fin material and because the size would easily flutter within the maximum wind tunnel velocity. The conversion of hertz to airspeed was found by assuming a linear distribution with a maximum wind tunnel speed of 120 miles per hour divided into 60 hertz with a resolution of 1 hertz. The following sections detail the results of different fin materials that were tested.

4.2.3.1 Testing of Wood Fins

The initial results from FinSim suggested the wood fins would flutter within the range of the wind tunnel velocity. Of the five fin geometries, the first to be tested was the square fin; strain gauges were attached in the optimal place determined by stress analysis performed using ProE Mechanical. Figure 76 shows where the major stresses occurred on the rocket fin. The strain gauges were initially attached horizontally, so that strain gauge was in line with the fin tab and parallel to the flow along the z-axis.

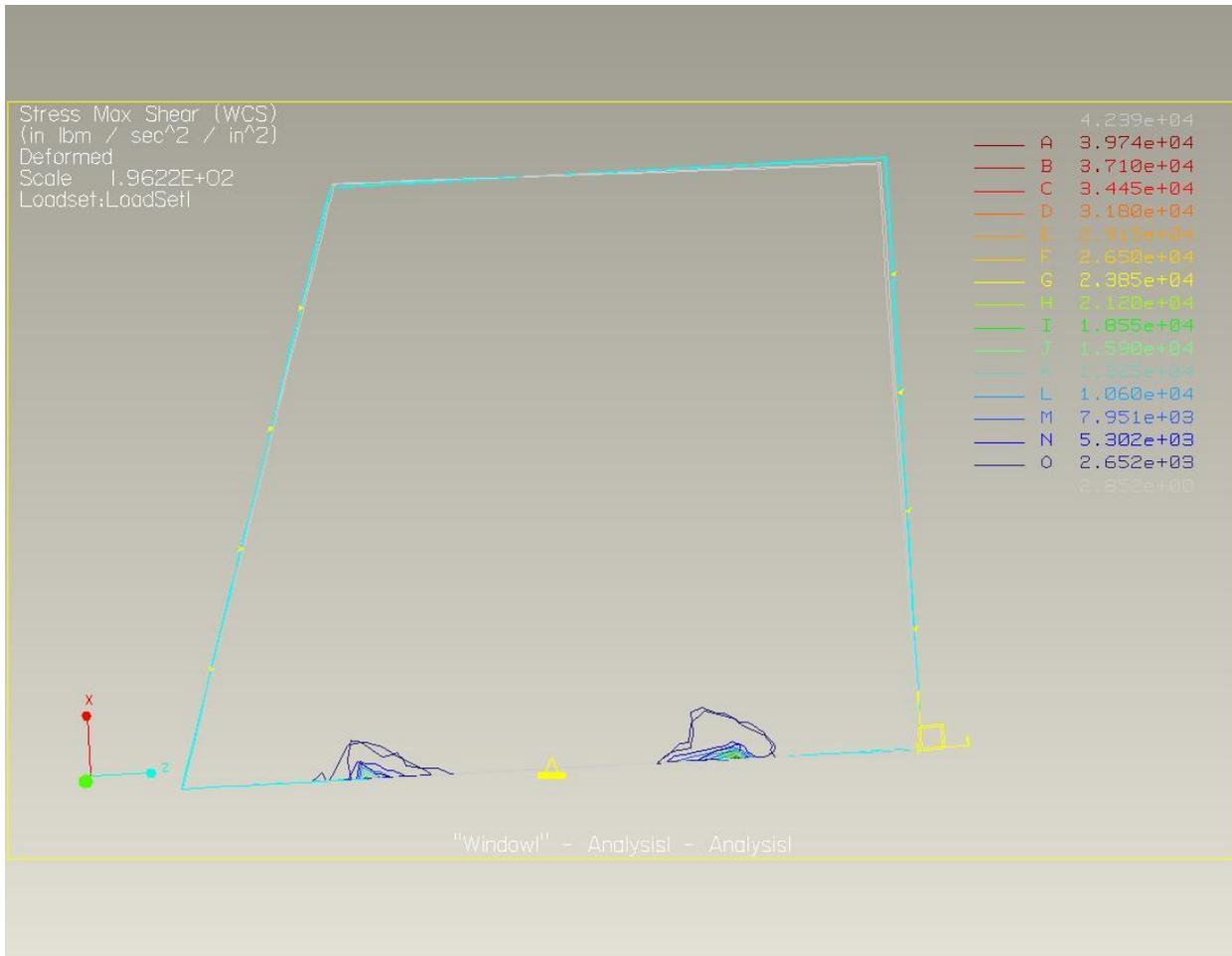


Figure 76: Stress analysis results displaying stresses on clipped delta rocket fin

Table 5 through Table 14 display the testing set up information for each of the five plywood fins. The tables provide the wind tunnel speed and temperature in addition to the excitation and gain of the amplifier. Also the geometry and where the strain gauge was located are listed as a heading above the table.

Square Fin

Test 1 (using strain gauge on trailing edge)

Wind Tunnel Speed	Temperature	Excitation	Gain
32 Hz	77 F	1.4 V	400
60 Hz	83 F	1.4 V	400

Table 5: Square Fin, Test 1

Test 2 (using strain gauge on leading edge)

Wind Tunnel Speed	Temperature	Excitation	Gain
32 Hz	62 F	1.7 V	400
60 Hz	71 F	1.4 V	300

Table 6: Square Fin, Test 2

Swept Fin

Test 1 (using strain gauge on trailing edge)

Wind Tunnel Speed	Temperature	Excitation	Gain
32 Hz	63 F	1.4 V	300
60 Hz	70 F	1 V	300

Table 7: Swept Fin, Test 1

Test 2 (using strain gauge on leading edge)

Wind Tunnel Speed	Temperature	Excitation	Gain
32 Hz	64 F	1 V	300
60 Hz	71 F	1 V	250

Table 8: Swept Fin, Test 2

Clipped Delta

Test 1 (using strain gauge on trailing edge)

Wind Tunnel Speed	Temperature	Excitation	Gain
32 Hz	63 F	1.4 V	170
57.4 Hz	69 F	1.4 V	250

Table 9: Clipped Delta Fin, Test 1

Test 2 (using strain gauge on leading edge)

Wind Tunnel Speed	Temperature	Excitation	Gain
32 Hz	63 F	1 V	250
57.4 Hz	71 F	1 V	200

Table 10: Clipped Delta Fin, Test2

Tapered Swept Fin

Test 1 (using strain gauge on trailing edge)

Wind Tunnel Speed	Temperature	Excitation	Gain
32 Hz	63 F	1.4 V	300
60 Hz	71 F	2 V	400

Table 11: Tapered Swept Fin, Test 1

Test 2 (using strain gauge on leading edge)

Wind Tunnel Speed	Temperature	Excitation	Gain
32 Hz	64 F	2 V	400
60 Hz	70 F	2 V	400

Table 12: Tapered Swept Fin, Test 2

Trapezoid

Test 1 (using strain gauge on trailing edge)

Wind Tunnel Speed	Temperature	Excitation	Gain
32 Hz	63 F	1 V	300
60 Hz	71 F	1 V	400

Table 13: Trapezoid Fin, Test 1

Test 2 (using strain gauge leading edge)

Wind Tunnel Speed	Temperature	Excitation	Gain
32 Hz	63 F	2.7 V	400
60 Hz	72 F	1.4 V	300

Table 14: Trapezoid Fin, Test 2

Following data collection, the data was put into MATLAB using a discrete fast Fourier transform (FFT) was performed to see how accurately the theoretical predictions matched the experimental results.

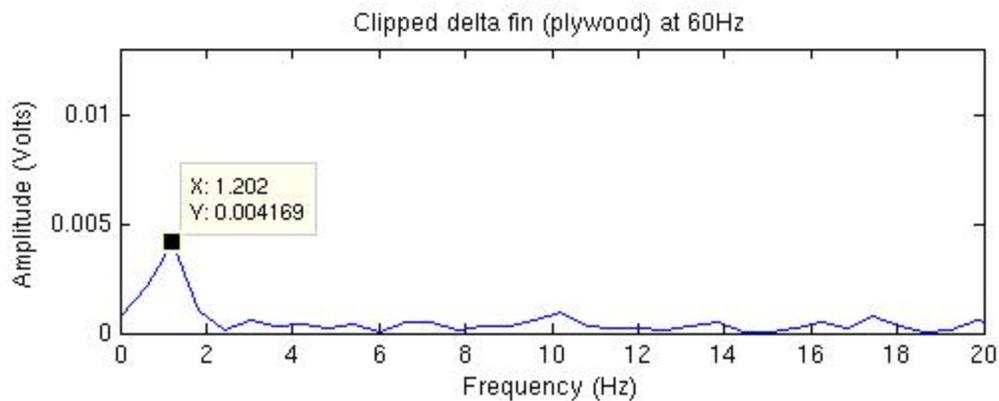


Figure 77: FFT of the Clipped Delta fin with Horizontal Strain Gauges

From this graph no information can truly be determined because there is too much noise to be filtered out. The first change made to the test procedure was to attach the strain gauges vertically in the hopes that the flutter might be better recorded.

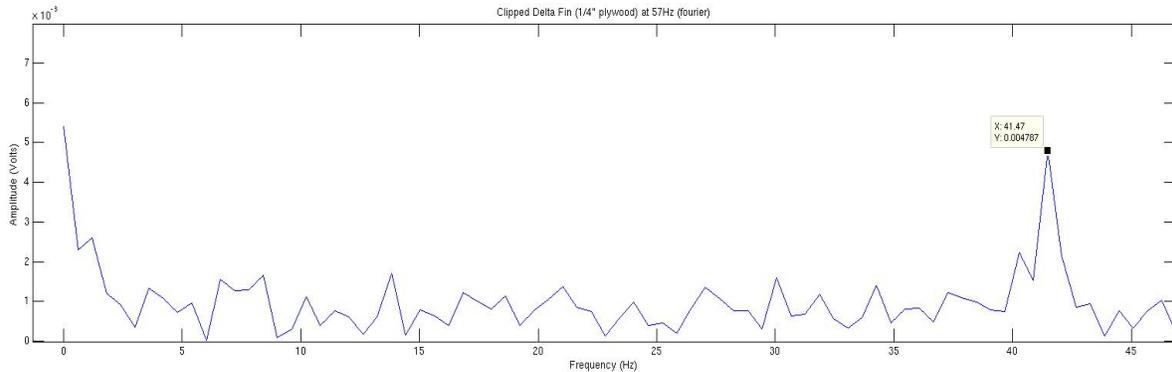


Figure 78: Clipped Delta Fin with Vertical Strain Gauges

4.2.3.2 Bar Test

Because of the inaccuracy of the test results on the plywood fins, it was deduced that there could be a problem with the data acquisition system. A test using a speaker driven by a signal generator was designated to test this hypothesis.

A speaker with a frequency range of zero to two thousand hertz was placed under a thin aluminum cantilevered bar. Connected to this bar was a strain gauge with two different orientations, one placed vertically and the other horizontally. This was to test if there was any difference in the results due to the orientation of the strain gauge. A signal generator was then connected to the speaker and adjusted to drive the speaker at approximately ten hertz. The speaker was arranged such that the bar was in contact with the speaker and driven physically. This was then wired the same as the typical setup with the fins through the Wheatstone bridge, and then to the LabVIEW system that was recording the data. It was easily determined that noise was still a severe problem. However, because of the strength of the driven signal, confirmation of the operation of the system was still obtainable.

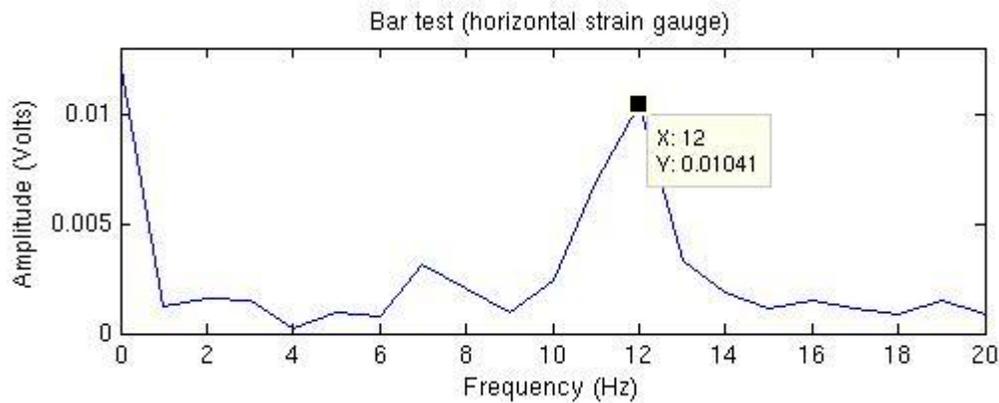


Figure 79: Aluminum bar test Fourier transform results with horizontal strain gauge placement.

Figure 79 shows the results of the test with horizontal strain gauge placement. When a Fourier transform was performed on the data, there was a spike on the graph of the transform at 12Hz. Although the speaker was driven at what was thought to be 10Hz, it was concluded that this was reasonable considering the possibility of the signal generator not being precise.

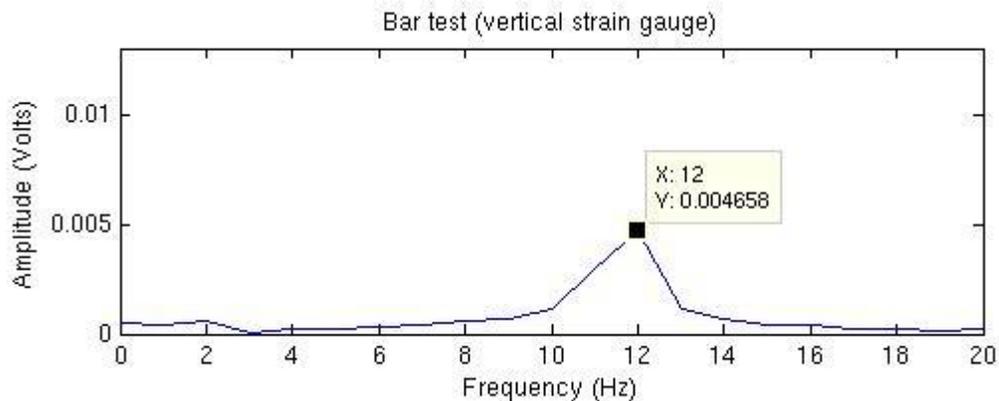


Figure 80: Aluminum bar test Fourier transform results with vertical strain gauge placement.

Figure 80 shows results of the experiment done with a vertically orientated strain gauge. It was noticed that less noise was seen with the vertically oriented strain gauge than the horizontally oriented strain gauge.

This experiment verified the functionality of the data acquisition system, and confirmed that the LabVIEW setup was operational. It also showed that placing the strain gauges vertically on the fin-can return a clearer signal, and henceforth may reduce some of the noise in the signal.

4.2.3.3 G-10 Fins

It was determined that the next material to be tested was G-10 fiberglass. Due to the cost of purchasing pre-cut fins, only three of the five fin geometries were ordered. The geometries acquired were the square, clipped delta, and tapered swept fin designs. The trapezoid and swept fins were not purchased because they were the largest of the five fins making them the most expensive. The strain gauges were attached vertically for optimal data collection.

Figure 81 shows a graph of the Fourier transform obtained from the data on first run. In this graph one can see that there is no peak at 7 Hz, or anywhere nearby, and therefore, there was an additional problem with the setup. It was found that there was a problem with the signal amplifier being used. A new amplifier was implemented and another test was performed.

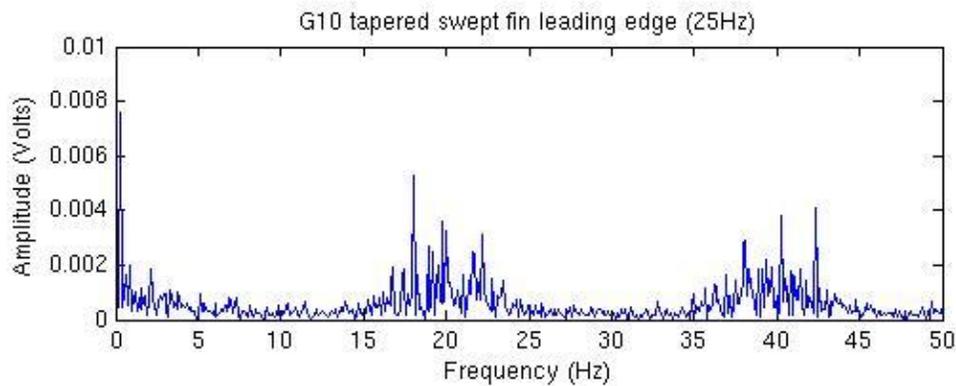


Figure 81: Fourier transform of G-10 fiberglass fin, tapered swept geometry at 25Hz

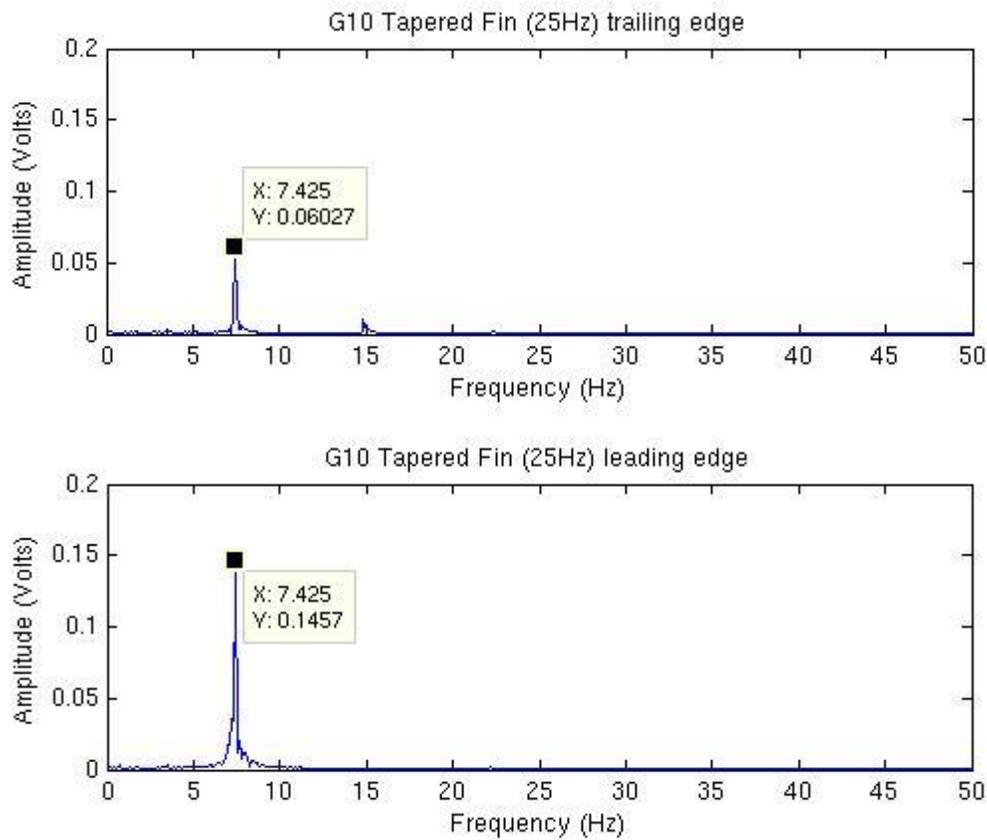


Figure 82: G-10 fiberglass fins Fourier transform results for the tapered swept fin with the wind tunnel at 25Hz

Figure 82 shows the results of the test run with the new amplifier. The test was performed with vertical placement of the strain gauges. The wind tunnel was adjusted to 25Hz and the data was collected from both the leading and the trailing edge of the fin. The Fourier transform shows a peak at 7.425Hz on both the leading edge and the trailing edge, which is only a 0.525Hz difference with the results that were received from FinSim. From this test we can see about a 7.6% error. Following this additional tests were ran on the remaining fins to obtain their natural frequency.

Figure 83 yields the Fourier transforms on the G-10 square fin. Like all of the fins, the strain gauges were placed vertically, and tests were performed on the leading and trailing edges.

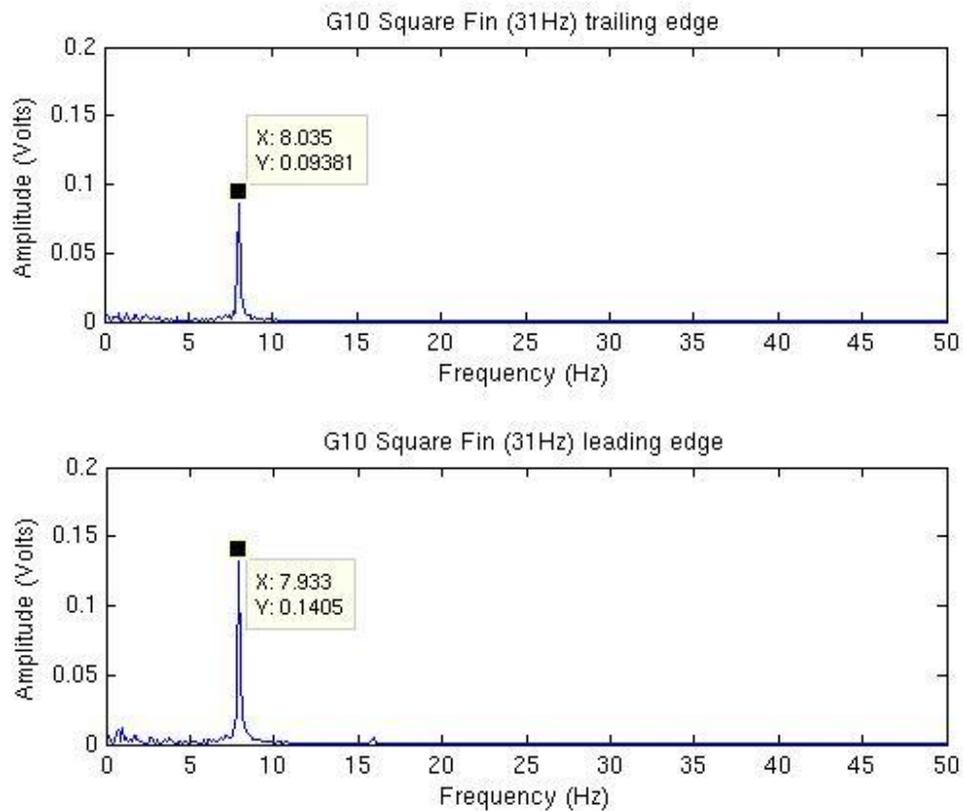


Figure 83: G-10 fiberglass square fin with vertical strain gauge orientation ran at 31Hz

For this test the wind tunnel was run at 31Hz in order to obtain flutter. The natural frequency was found to be 7.993 for the leading edge and 8.035 for the trailing edge. FinSim calculated the frequency of this geometry to be 6.9Hz also, or a difference of 16 %.

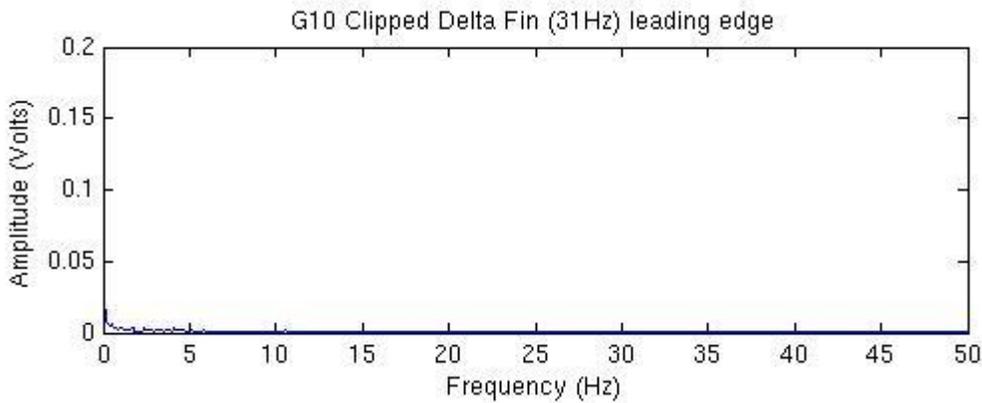


Figure 84: G-10 Fiberglass; Clipped Delta fin geometry Fourier transform results. Wind tunnel speed of 31 Hz.

Figure 84 shows the final fin which was tested, the clipped delta fin. The trailing edge strain gauge was non-functional, so no data was collected from this edge. In analyzing the Fourier transform performed on the fin, with wind tunnel speeds of 31Hz, no natural frequency was collected for this particular geometry. It is possible that the critical velocity required to generate this frequency was not attained. FinSim calculations indicated the natural frequency of this geometry to be 7.5Hz. Table 15 shows each fin geometry which was tested and their corresponding frequencies from both FinSim and the wind tunnel.

Fin Geometry	Frequency (FinSim) Hz	Frequency (Empirical) Hz
Tapered Swept	6.9	7.425
Square	6.9	8.035
Clipped Delta	7.5	--unobtainable

Table 15: Fin Geometry and corresponding frequency results from FinSim and wind tunnel testing.

4.2.4 Data Acquisition System

The FCDAQ was tested in the laboratory to guarantee its functionality as well as to compare the acquired data with the fast Fourier transform obtained in LabVIEW. Due to time constraints only a calibration could be run on the FCDAQ hardware, and the LabVIEW program verified its data output.

In order to test the hardware, an easy system was devised by recycling premade components. The oversized clipped delta fin was converted to a cantilevered beam. One of the two half Wheatstone bridge

configurations was soldered to the mating connector of the FCDAQ's inputs. This allowed an easy test configuration so that each port of the FCDAQ could be tested individually.

The fin was then connected to the FCDAQ and the input that was selected. Following this the Wheatstone bridge had to be balanced. This was accomplished by turning the corresponding potentiometer input to "zero out" the now connected bridge. Refer to for the component layout. The gain was then set by turning the corresponding "Gain Adjustment Pot"

With this done the FCDAQ was then triggered on by pushing the "Get Data" button. The fin was then plucked and allowed to oscillate for approximately 35 seconds. The fin was plucked a second time in the opposite direction and allowed to oscillate once more.

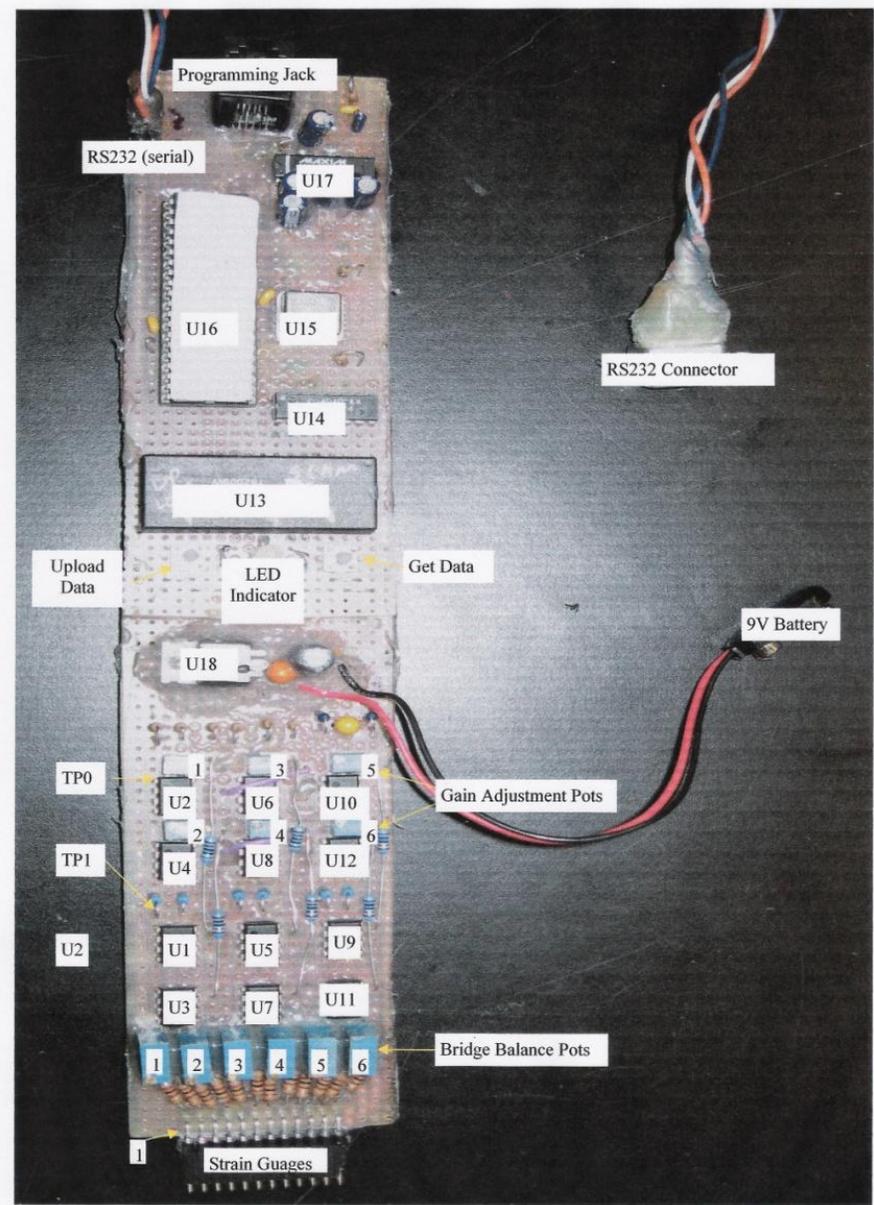


Figure 85: Component Layout

This data was then stored in the FCDAQ onboard memory. When the memory becomes full the FCDAQ LED indicator will blink red. The unit was powered off, the Wheatstone bridge disassembled and the RS232 connector attached. A computer using HyperTerminal was next connected and the correct “COM” port configured for 19200 baud, no parity, 1 stop bit, and no hardware control. The “capture text” command was then activated on HyperTerminal. Once this was double checked the “Upload Data” button was pushed and the data began its download onto the computer hard drive.

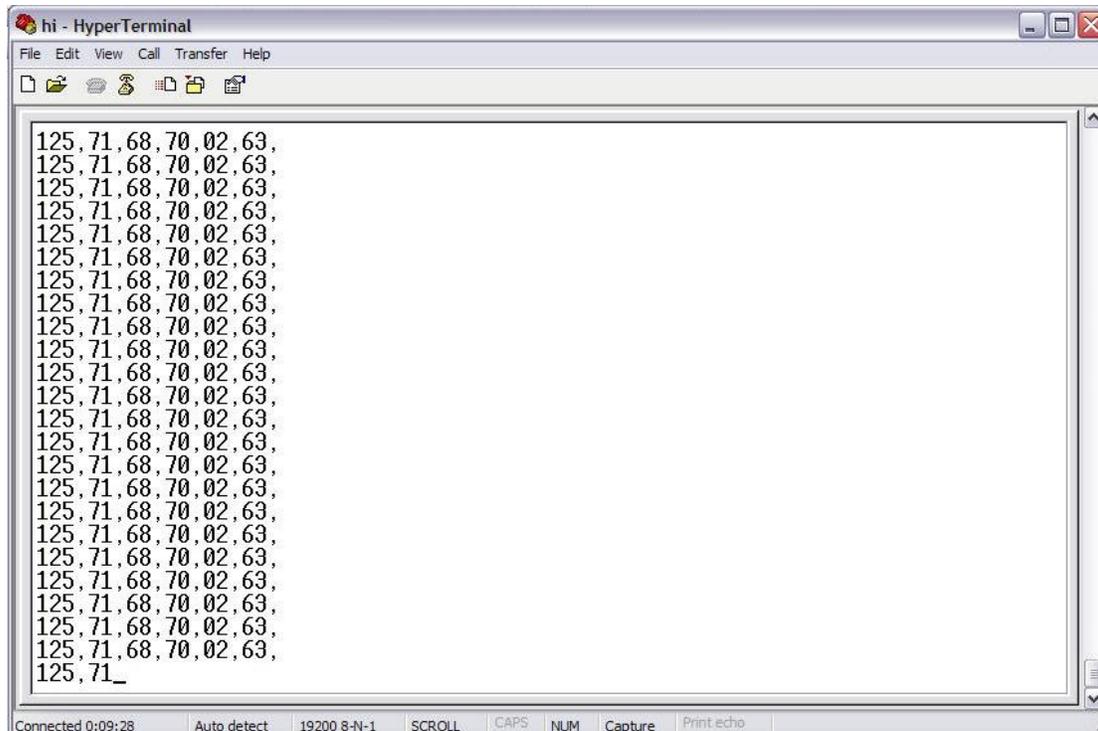


Figure 86: Hyperterminal Download

Figure 86 is a picture of the data being uploaded from all 6 inputs. The only channel with stored information will be channel 1 located in the first column of data. This data is then output in the form of a text file (.txt) but is formatted for a comma separated values (.csv) file. The file extension is then changed and the data imported into LabVIEW. Because this data is a critically dampened oscillation, no fast Fourier transform is valid. The results are plotted and compared with the calibration test of the data acquisition system used in the wind tunnel.

The recorded signal is very well defined; however, there are some issues in the collection process. The stray lines that are out of place could be caused by several factors, the first being noisy operational amplifiers. Since there are no filter capacitors currently attached, a .1 microfarad capacitor could be added to each amplifier. Another possible solution is to change the test setup to accurately depict the fin flutter. Since the fin is very thin and is a set up as a cantilevered beam, gravity constantly pulls it down. If the fin is placed vertically it will factor out any noise or offsets due to the orientation. It is also important to note that the noise is only found on the upward side of the fins oscillation. Finally a RS232 port analyzer or a precision oscilloscope could be set up to measure the exact time between output pulses of the FCDAQ communication with HyperTerminal. This data can then be verified using MPLAB and by checking the stored binary number. This would eliminate any chances of data mismatch between the collection and receiving process. The first issue with the FCDAQ that must be resolved before it is ready for operation

pertains to getting a clean signal. Once this is solved the FCDAQ must be properly configured to analyze the data by adjusting the bridge and gain potentiometers. A lithium polymer or nickel cadmium battery pack could also be configured to extend the battery life. More test points could also be added to make calibrating easier. The program could be updated and more sensors could be added to allow the unit to also control the firing of the ejection charge and recording altimeter data.

4.3 Hybrid Rocket Propulsion

4.3.1 Testing Results

A summary of the hybrid rocket motor testing performed is shown in Table 16. The testing information shows that at least thirty ignition attempts were performed; however, the complete number is probably much higher, as earlier test failures were not recorded. Each full test number represents a clearly successful ignition and burn. For later testing, an additional letter was added to signify which ignition attempt in each test number it was; for example, 13A was the first paraffin test, followed by 13B, and so on. After 13K, a successful test, the numbers increase to 14, giving the next test as 14A.

Test Number	Fuel	Specified Burn Time	Notes
7	PVC	5 seconds	First burn
8	PVC	5 seconds	
9	PVC	5 seconds	Data not recorded
10	PVC	10 seconds	Data not recorded
11	PVC	1 second	
12	PVC	8 seconds	Force data not recorded due to mechanical issues
13A-J	Paraffin	3 seconds	Ignition not sustained.
13K	Paraffin	3 seconds	First successful paraffin test. Short duration.
14A	Paraffin	3 seconds	Ignition not sustained.
14B	Paraffin	3 seconds	Second successful test. Short duration.
15A-15F	Paraffin	3 seconds	Ignition not sustained.
15G	Paraffin	3 seconds	Successful burn, full duration
16A-I	Paraffin	3 seconds	Ignition not sustained.

Table 16: Hybrid Motor Testing overview

Our first successful motor tests were named tests 7 and 8, and were done in mid-November. At the time of these tests, most of the DAQ, sensor network, and computer control systems were in not in their final state, thus a very limited amount of data was recorded. Because of this, the data which was recorded was limited to force and line pressure. The line pressure produced little useful data to show, and thus was excluded from Figure 87. The data acquisition system and LabVIEW code were also incomplete versions during these tests, and only recorded approximately 600 samples/sec.

For tests 7 and 8 the solenoid opened at approximately 0.6 seconds after ignition, as determined by a short-duration line pressure drop. A short duration thrust spike is visible from the ignition of augmentation, a small piece of ammonium perchlorate propellant (APCP). The thrust drops off temporarily while the motor ignites and the combustion chamber achieves operating pressure, then returns to near design levels for six to seven seconds. The test was intended as a five second burn, however, and the slightly increased duration of the entirety of the test lead to concerns that the computer controller was inaccurate. However, after some additional testing of the timing, this inconsistency was attributed to mechanical delays and some lingering nitrous in the line between the solenoid and the motor.

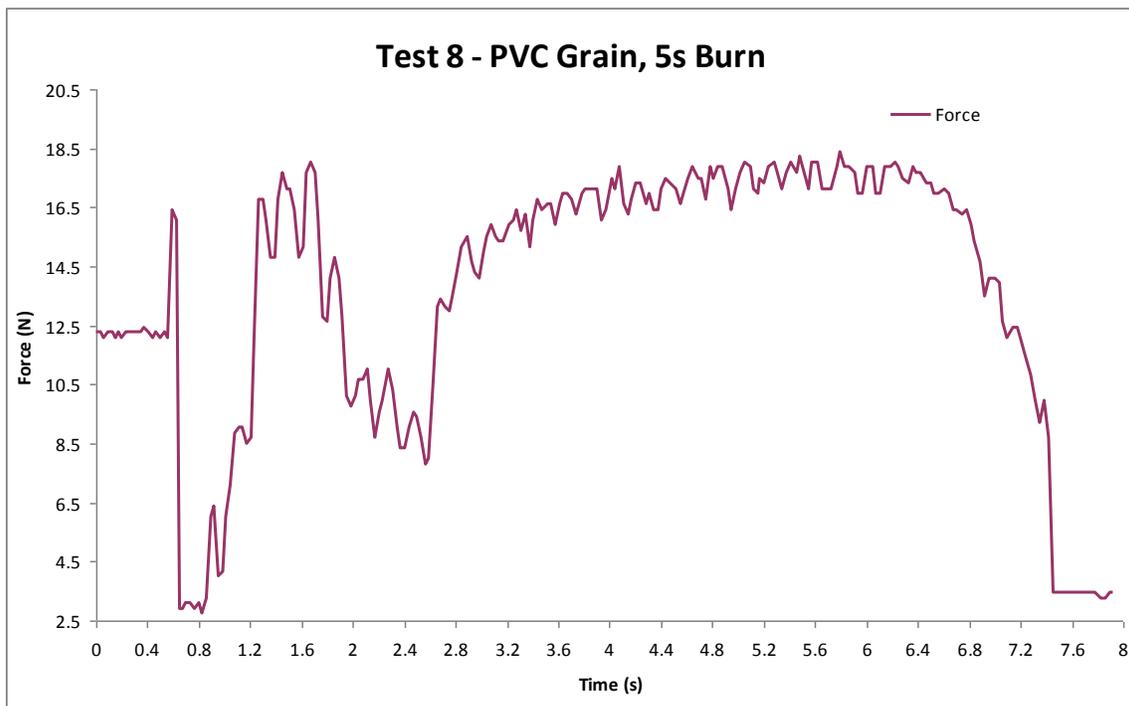


Figure 87: Test 8, 12N PVC Grain, 5s burn

Information about the timing of when the solenoid opened and closed was found based upon changes in the line pressure; while the solenoid's operation was computer-controlled, there was initially no other way to see the solenoid's opening and closing. However, this allowed us to examine some of the timing, and it was obtained so that fine tuning for the ignition and flow could be achieved.

For both tests, the overall magnitude of the thrust curves matched the specifications for the expected motor thrust. The thrust in Test 8 experienced a momentary drop for 0.75 seconds to around half the rated thrust. On the film footage, this corresponded to a short duration of reduced activity from the motor. This was the first view of combustion instabilities that would become significant issues during later paraffin motor testing.

Due to recording errors, the information from Tests 9 and 10 were not recorded to file. This included the first usage of a pressure tap into the chamber to record pressure, as well as longer duration burns up to 10 seconds.

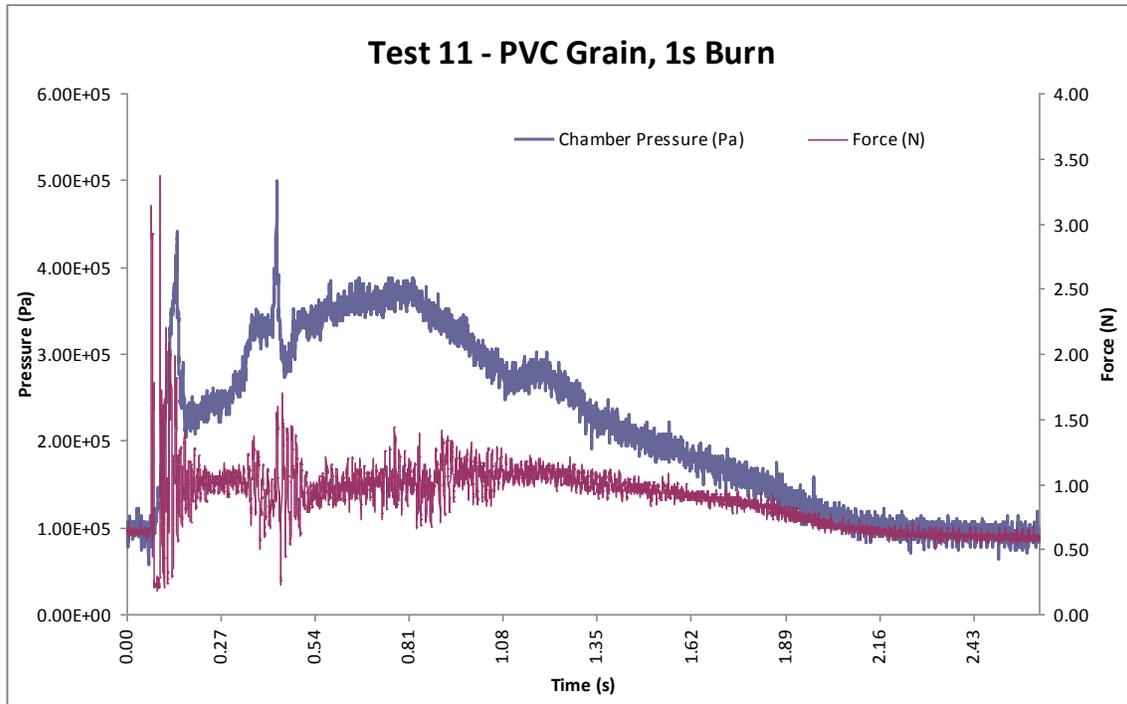


Figure 88: Test 11, 12N PVC Grain, 1s burn

Test 11 was a short duration one second burn, which tested the motor's short-term operational characteristics. This was also the first test of the new LabVIEW code sampling program at 10 kHz across three channels. The motor never came up to the proper thrust levels due to the duration of the burn, but was still an excellent operational test of the motor and the new data recording system.

For Test 12 an 8 second burn time was selected to obtain additional data on the motor operation. The data for this test is shown in Figure 89. Unfortunately, while the test was a success, no force data was recorded as the test stand did not press against the load cell. After this test, the team decided to reconstruct the test stand in order to solve this issue and prevent any further loss of data. Another issue revealed in this test is the occurrence of a combustion pressure anomaly at approximately two seconds in, where chamber pressure dropped significantly (nearly 100 kPa) briefly before rising back to normal.

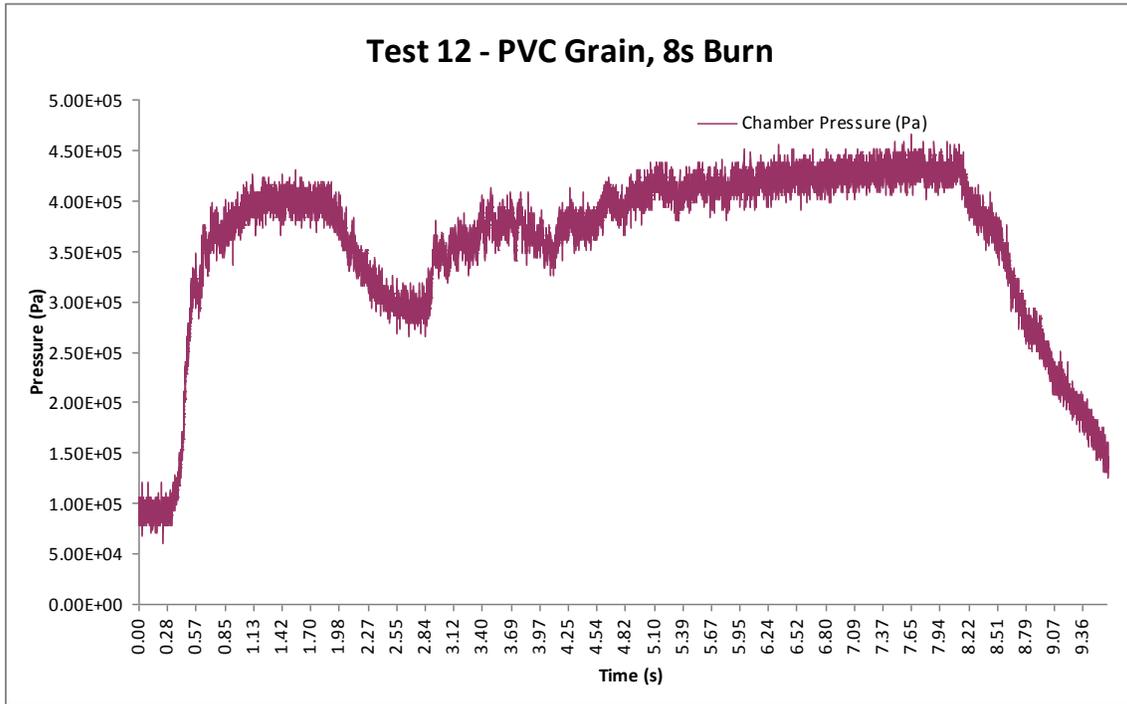


Figure 89: Test 12, 12N PVC grain, 8s burn

Test 12 also signified the end of 12 N PVC grain testing, with Test 13 as the beginning of the 50N paraffin motor testing. In this series of testing, combustion irregularities became an issue, with only a 10-15% success rate over all ignition attempts. Any successful burns lasted a short duration compared to their planned burn times.

Test 13K was the first success of the paraffin hybrid, following flame holding or kicked igniter issues on tests 13A-13J. The motor only sustained stable combustion for 0.5 seconds. After this phase, the motor produced repeated momentary combustion. Shortly thereafter, the burning completely subsided.

The data shown in Figure 90 clearly shows the burn becoming unstable as chamber pressure drops, past about 1.2 seconds into the test. What is unclear is whether the pressure drop was caused by the oscillations, or the oscillations caused the pressure drop. Considering the motor was operating smoothly at much lower pressure values earlier in the burn, it would seem the burning instabilities lead to the drop in pressure.

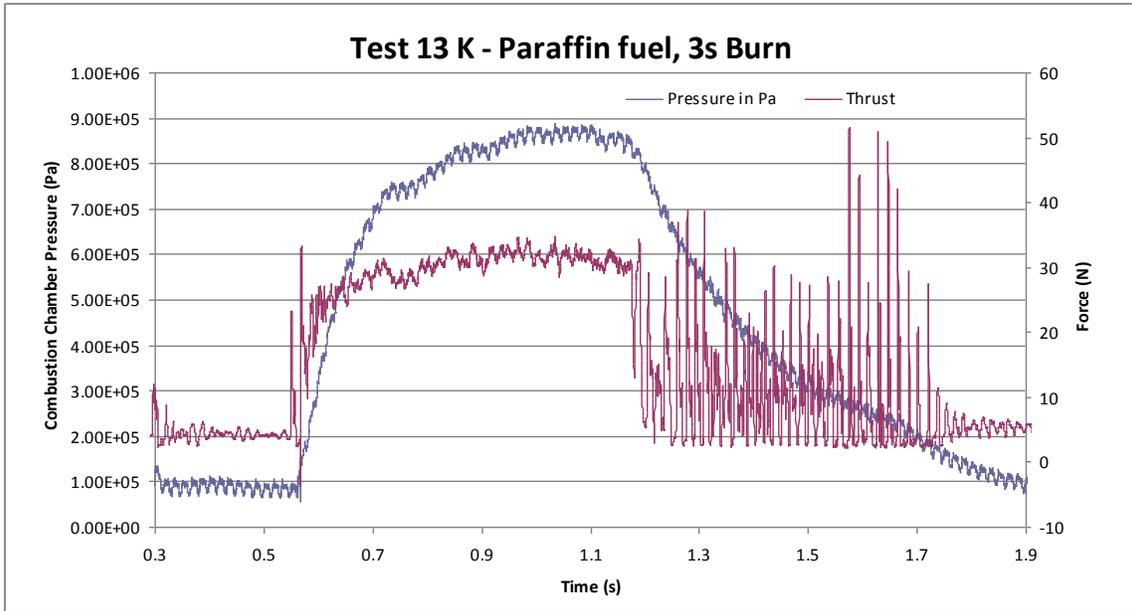


Figure 90: Test 13K, 50N Paraffin motor, 3s burn

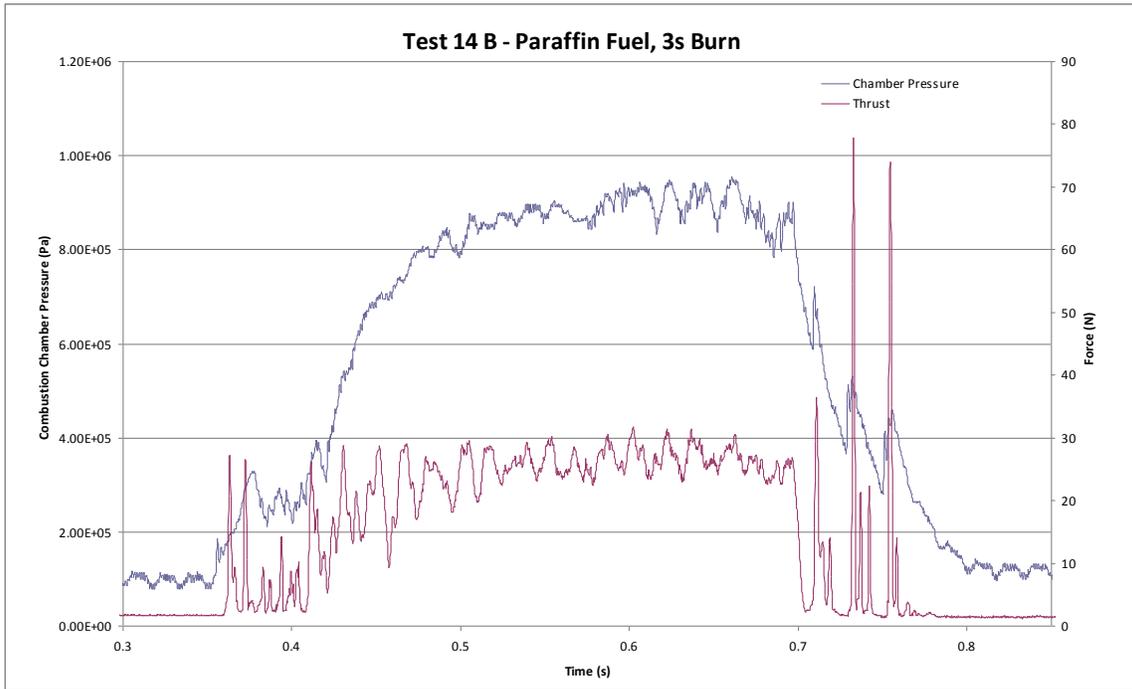


Figure 91: Test 14B, 50N Paraffin, 3s burn

While the motor operated inconsistently for much of the burn, the period of time during which the motor operated smoothly provided some interesting information about performance characteristics. The motor operated at only 30N, compared to its rated thrust of 50N; the chamber pressure was also significantly under planned values.



Figure 92: Exhaust Plume From 50N Paraffin Motor with 4 Port Injector, Test 14b

Test 14B, the next successful paraffin test, repeated many of the same general characteristics as the earlier 13K test. The motor had a short stable burn period before becoming unsteady and shutting off early. This time the stable burn period lasted only two tenths of a second before subsiding completely. The significant force spikes lasted for a significantly shorter time as well, and the motor's entire period of operation ended in under a second. Key motor characteristics were significantly under design specifications, but still consistent with test 13K.

A picture of Test 14B is shown in Figure 92. This photo of the motor was taken near the middle of the test. The large bulge near the end of the plume is thought to be evidence of fuel rich combustion. This further suggested that the nitrous oxide was not completely decomposed. Also noteworthy are the well formed Mach diamonds in the supersonic exhaust flow.

Figure 93 shows the exhaust plume of a test where the motor performed for only a short period of time, under 0.125 second. This test was regarded as a failure due to the extremely short duration.

Test 15G, shown in Figure 94, was the last successful motor firing performed, and the most successful paraffin motor test. The motor was able to maintain stable combustion for nearly two seconds, producing an average thrust of 35N and an average chamber pressure of 900 KPa. The grain did not burn out completely, but did regress significantly during the test. Similar to the other paraffin tests, the motor went through a brief period of unsteady combustion near the end of its operation. The burn in this test was significantly smoother compared to the other successful tests.

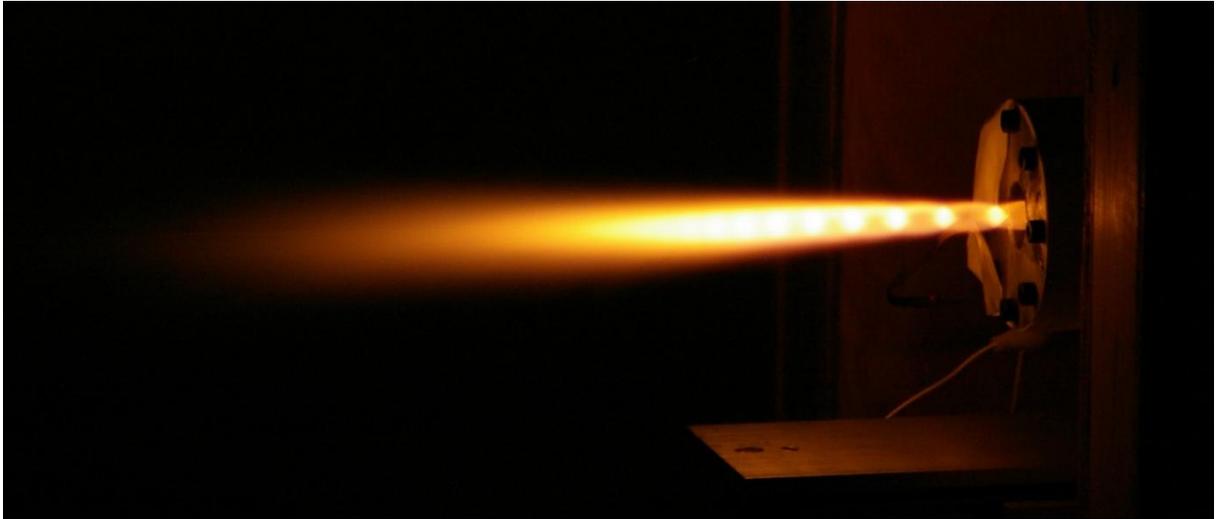


Figure 93: Just after ignition of test 15 c, which failed to continue to burn

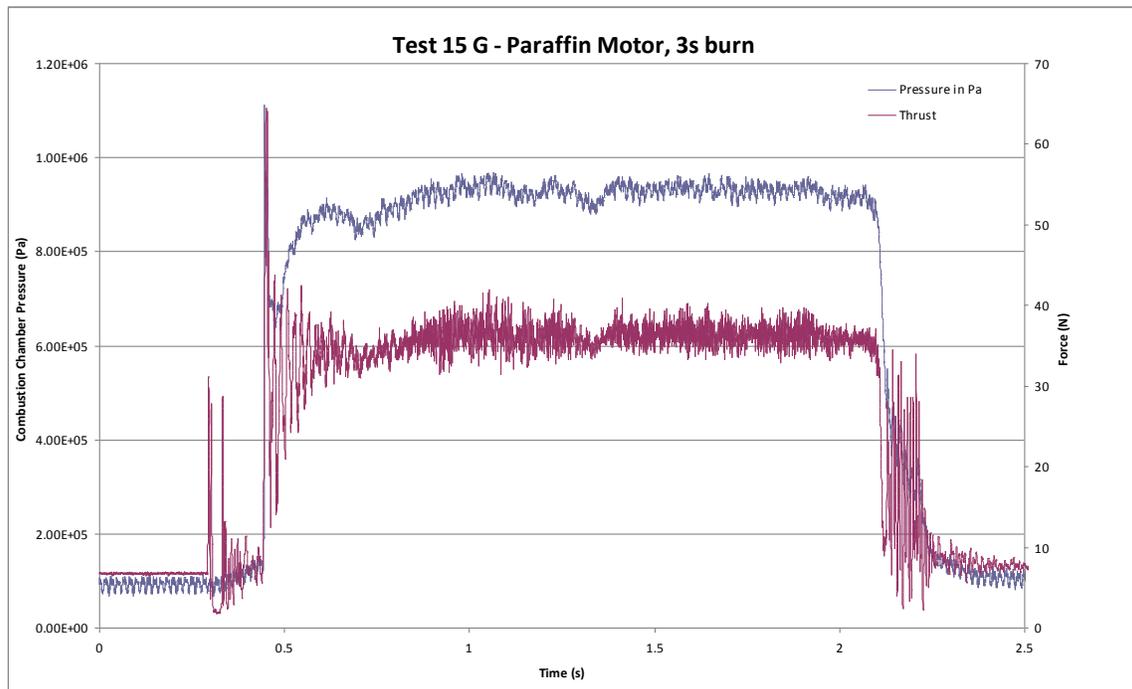


Figure 94: Test 15G, 50N Paraffin motor, 3s burn

Data analysis for the motors was done mainly in Excel, due to its ease of usage. The data is output from LabVIEW as a .csv (comma separated values) file that Excel can open. The data went through several overall structures throughout the project, and most of these had to be analyzed individually.

The original data structure contained a time column and the three data lines: Pressure 1 (chamber), Pressure 2 (line), and Force, along with a header label. No additional information was placed into the sheet, and the data was not converted using any calibration information. The second version was able to convert the data from volts to whatever units were specified and wrote a basic header containing labels, time between samples, and start time. The final LabVIEW version wrote the entirety of the motor settings to a header, converted all data from voltage information, and included Boolean data on when the ignition and flow had been triggered. Each was different, and thus had to be analyzed differently, depending on when the data was taken.

4.3.2 CFD Results and Analysis

The main purpose of creating the CFD models of the four-port and single port injector was to see how effectively each injector could disturb the boundary layers which form at the surface of the fuel grain inside the combustion chamber.

In order to analyze the disruption of boundary layers in each model, two factors were chosen to base the analysis upon. These were vorticity and turbulence at the fuel grain surface. As defined by Robert Fox, “The vorticity is a measure of the rotation of a fluid element as it moves in the flow field.” [22] These rotations help to interrupt the formation of the boundary layer by mixing oxidizer from the main flow as well as facilitating the transfer of any heat formed from combustion. The second factor which will be studied is turbulence. A turbulent flow can be defined as a flow “...in which the fluid particles rapidly mix as they move along due to random three-dimensional velocity fluctuations.” [22] In FLUENT, turbulence intensity is defined as “... the ratio of the root-mean-square of the velocity fluctuations to the mean flow velocity.”[26] This causes turbulence to be displayed as a percentage. Creating a turbulent flow in a boundary layer is desirable because it also improves the mixing of pure oxidizer with the fuel grain surface as well as to facilitate heat transfer.

After the factors to investigate were decided upon, the next task was using FLUENT to display the appropriate contours. Contours of vorticity for each injector can be seen in Figure 95 and Figure 96. Any dark spaces appearing in the contours represent values out of the range investigated. By limiting the range one can get a better representation of the vorticity near the boundary layers.

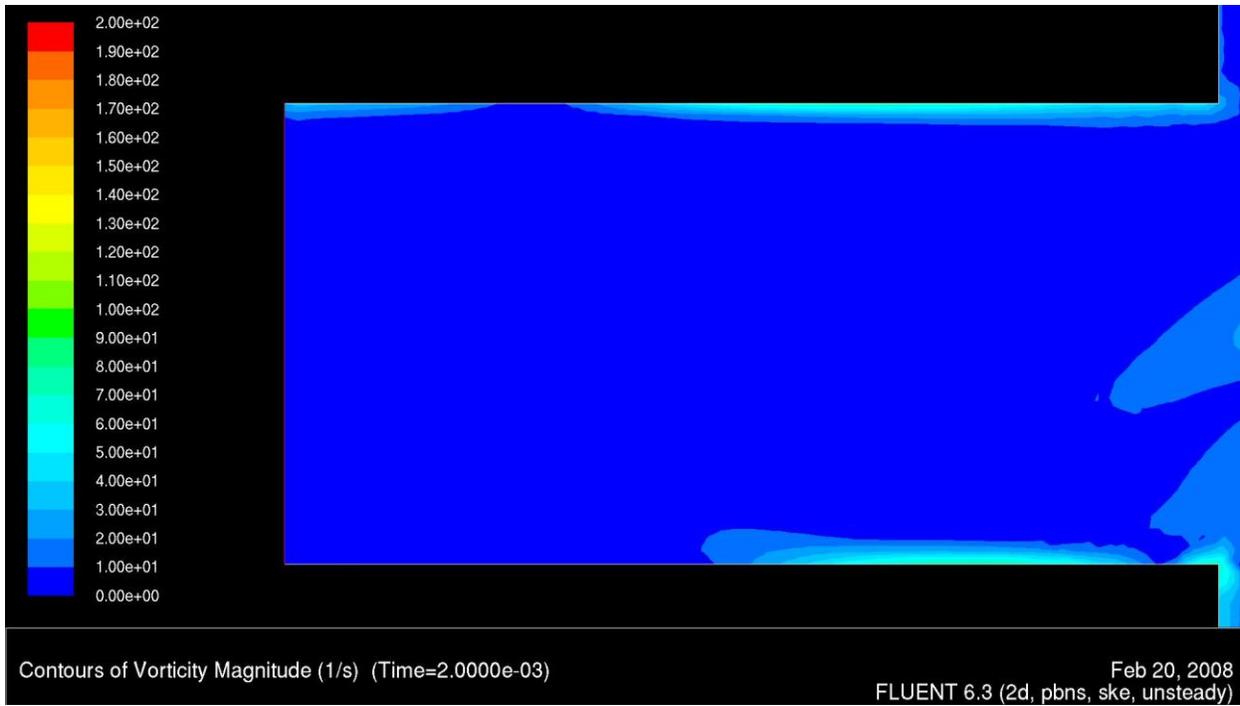


Figure 95: Contours of Vorticity for Single Port Injector

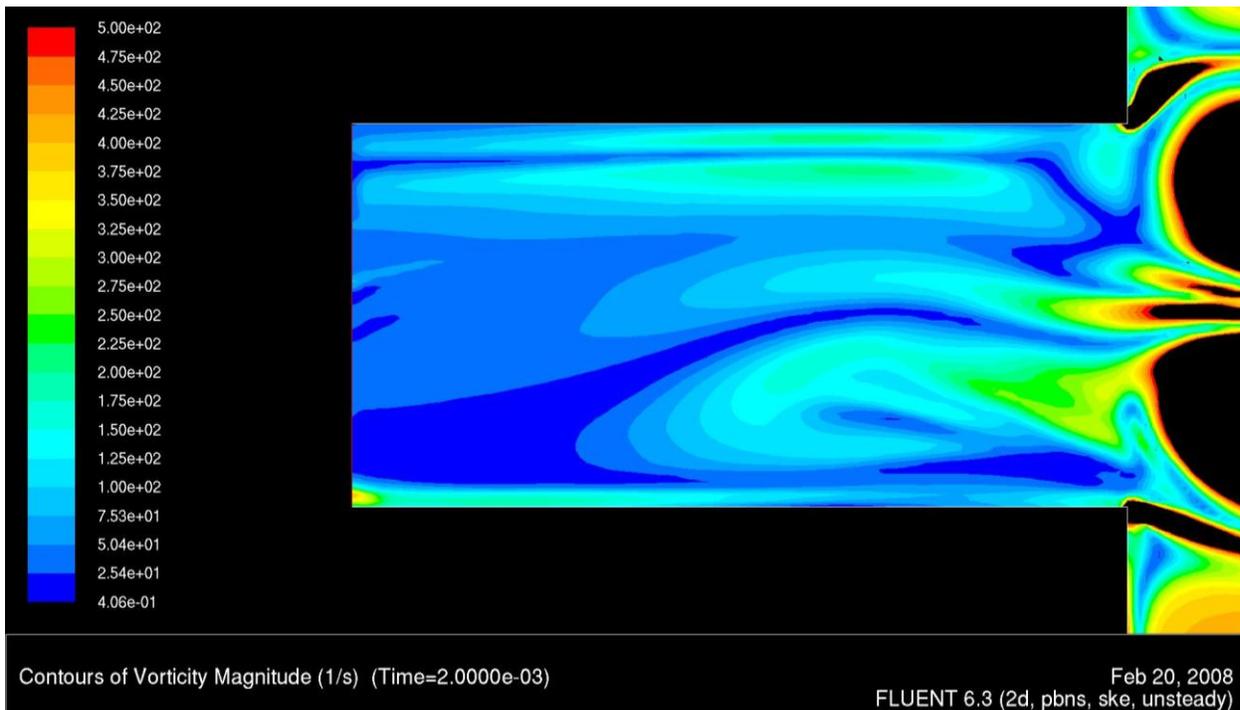


Figure 96: Contours of Vorticity for Four-Port Injector

From these two figures, it can be seen that larger vorticities exist in the four-port injector than in the single port. Small vortices form at the grain surface of the single port injector at a magnitude of approximately 6sec^{-1} . Almost no vortices form in the main velocity stream, which would hinder fuel

mixing and heat transfer. When looking at the four-port injector it can be seen that vorticity magnitude at the fuel grain surface is around $15\text{-}20\text{ sec}^{-1}$, or almost three times as large when compared to the single port. Vortices also exist in the main oxidizer flow stream which will help to increase the fuel mixing and heat transfer in this case.

The next factor used in the analysis of these two injectors was turbulence. As explained earlier, turbulence is a measure of how chaotic a flow is. The higher the turbulence, the greater the heat transfer rate and fuel mixing rate at the fuel grain surface. Contours of turbulence for both injectors can be seen in Figure 97 and Figure 98. In these figures, the injector is in the same location (right boundary with flow from right to left) as in the previous figures.

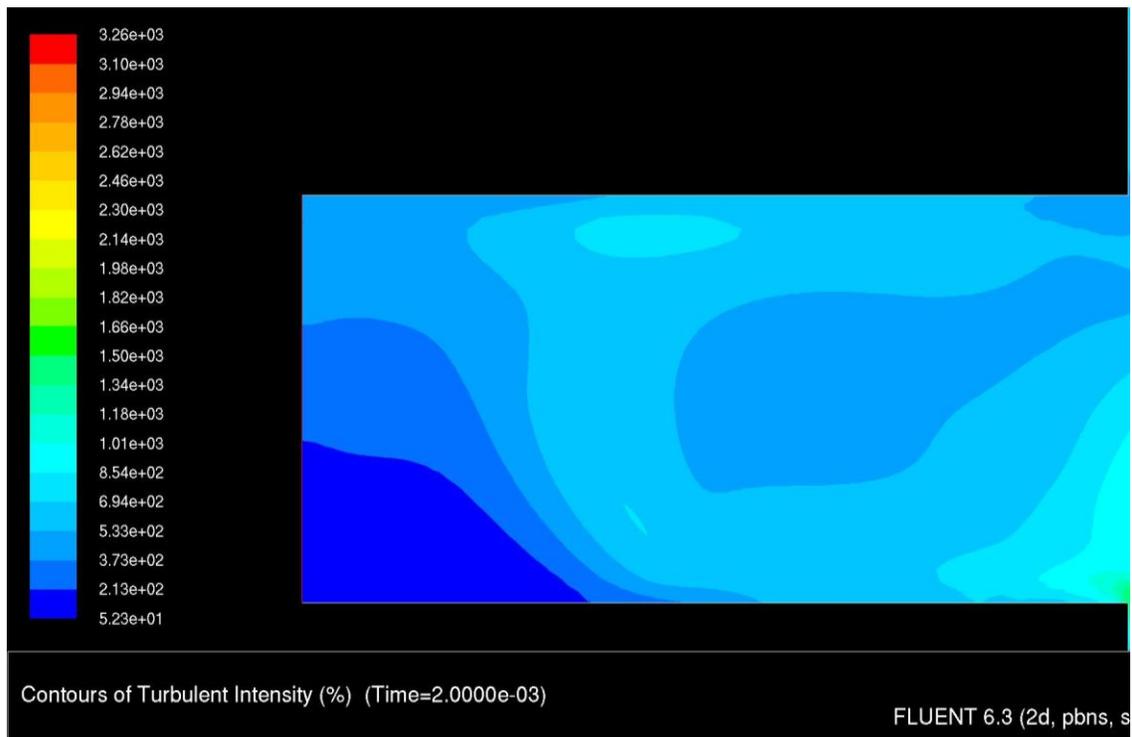


Figure 97: Turbulent Intensity in Single Port Injector

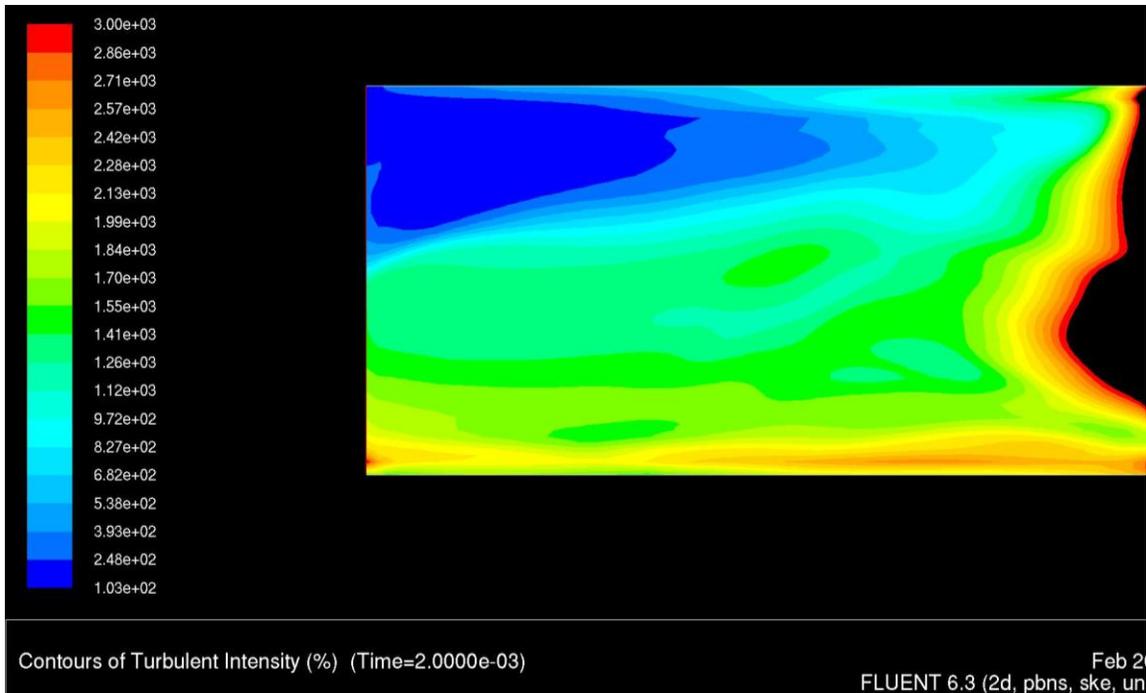


Figure 98: Turbulent Intensity in four-port Injector

Based on the turbulent intensity contours, these two figures show that the four-port injector outperforms the single port with respect to enhanced mixing. At the surface of the grain the single port injector model shows a turbulent intensity of around 700 %. The four port injector model registers up to around 2500% percent turbulent intensity at some points near the surface of the grain. Comparing these results, one sees that the four-port injector can achieve turbulent intensities approximately four times that of a single port injector.

5 Conclusions

This project satisfied the goals initially set. The following section details the conclusions from satisfying those goals.

5.1 Initial Test Vehicle

There were two main purposes in building the initial test vehicles. The first was to have a platform to test the flight computers and make sure that they were still operational and functioning correctly. The second purpose was to give everyone in the MQP group the experiences of building and flying at least 1 high powered rocket. This would expand their knowledge of model rockets in general and help determine

desired characteristics for the final flight vehicle. The first goal was met because it was determined that one of the flight computers was deemed non-functional. The second computer did work though, and recorded all the necessary data that would be required in the final launch vehicle. The second goal was met in the fact that every member of the MQP group participated in building one of the two initial launch vehicles. Every member of the group should now have at least a basic idea of how to construct and launch a high powered rocket.

5.2 Aerodynamics

No valuable information on fin flutter was gained from testing the wood fins because the wind tunnel did not achieve the speeds required for the fins to flutter. No flutter frequency was recorded in LabVIEW or MATLAB. Once this was determined, testing of the data acquisition system was performed to verify that the setup was working as designed. Using the bar test as described in Section 4.2.3.2, the DAQ was tested to ensure proper operation.

From testing the G-10 fins, data was obtained. The frequency signal was not strong enough to show through in the Fourier transform calculated in MATLAB. Following this a malfunctioning amplifier was discovered and replaced.

With a functional amplifier, testing continued on the G-10 fiberglass fins. Accurate data of the natural frequency of the fin geometries were gained. Data was collected from both the leading and trailing edges of the fin and Fourier transforms were performed on each of these trials. In using this experimental data to cross with the FinSim results, a discrepancy of 11.5% was found. The first geometry tested, the tapered swept fin, showed a small error of only 7%. The square fin had a higher error of 16%. No data of the third fin geometry, the clipped delta, was attained because flutter was not achieved in the wind tunnel.

5.3 Hybrid Rocket Propulsion

5.3.1 Ignition Delay and Chamber Design

Due to problems in producing consistent results with the 10lb paraffin motor, additional research needs to be performed to obtain a better understanding of how injector design affects performance characteristics. During testing it was found that when using the four port injector design, the vaporization delay of the nitrous oxide was reduced. To help solve the problem of stable combustion, two methods could be investigated. These suggested methods are shortening the vaporization delay or increasing grain length. Attempting to reduce the vaporization delay would require the addition of more injector ports

with smaller holes. Doing this would prove to be very time consuming due to the machining time associated with drilling the small injector ports. The option of increasing the grain lengths would be the next step. With this increase in grain length, the thrust the motor produces will also increase. The increased grain length could also cause regression rate to be non-uniform. This would be caused by the fuel regression rate being slower at the injector end of the grain.

Due to the time constraints associated with the project, the team was unable to manufacture the 25 lb motor that was designed. This motor had a grain length 3.5 inches longer than the grain on the 10 lb motor. With this increase of grain length, the nitrous would have a longer distance to vaporize. If test firing of this motor eventually proves to be successful, different injector designs could be implemented in an attempt to see how they affect performance characteristics. If this design still produced the same combustion problems associated with the 10 lb motor, the same process of shortening the vaporization delay with the addition of smaller injector ports could be used. If problems still persisted, changing the grain length would once again be the next option.

5.3.2 Injector Flow Modeling

Using the factors that were discussed in Section 4.3.2, it is clear that the four port injector outperforms the single port injector with respect to improved propellant mixing. More specifically, the four port injector should produce a higher regression rate of the fuel grain and more stable combustion. This conclusion is drawn from comparative analysis between the two injectors. The data found cannot be used as definitive due to the fact that this flow was modeled as a “Cold Flow” in that no combustion was modeled, however it can reveal typical flow patterns for both injector types.

5.3.3 Motor Operation

From an operational standpoint the test system worked as designed. The main reasons for failure in most of the tests were due to either some sort of human error, or the use of improper timing in the flow and igniter steps. Some of the human error included not properly configuring the control system, crossing wires in the operation box, and improper igniter creation. The timing configuration of the tests were one of the main factors that were constantly changed. The use of a pre-burn time, where the igniter was fired before the oxidizer valves were open, was used in some instances and a pre-flow time delay was used in others. Besides these slight changes, the test system was successful over the number of tests that were required.

5.4 Final Test Vehicle

After the design process of the final flight vehicle was completed, the components needed to manufacture the rocket were ordered. Construction of the rocket was in progress until it was halted due to time constraints. Plans to continue construction of the rocket in the future are in place with hopes of testing the FCDAQ and to see results of fin flutter.

6 Resources

- [1] Theordosen, Theodore, *A Theoretical and Experimental Investigation of the Flutter Problem*, NACA Report 685, 1940
- [2] McDonald, Duncan, *Fin Flutter*, <http://www.info-central.org/design_finflutter.shtml>
- [3] Milligan, Tim Van. "Basics of Dynamic Flight Analysis" Apogee peak of flight newsletter, Sept, 2007
< <http://www.apogeerockets.com/education/downloads/Newsletter193.pdf>>
- [4] Milligan, Tim Van. "Basics of Dynamic Flight Analysis" Apogee peak of flight newsletter, Sept, 2007
< <http://www.apogeerockets.com/education/downloads/Newsletter196.pdf>>
- [5] Gordon K. Mandell, George J. Caporaso and William P Bengan, topics in advanced model rocketry, the MIT Press, Cambridge Ma, September 1973
- [6] Babister A W: *Aircraft Dynamic Stability and Response*. Elsever 1980, ISBN 0-08-024768-7
- [7] Milligan, Tim Van. "Basics of Dynamic Flight Analysis" Apogee peak of flight newsletter, Sept, 2007
< <http://www.apogeerockets.com/education/downloads/Newsletter195.pdf>>
- [8] Chiaverini, M, and K Kuo. *Fundamentals of Hybrid Rocket Combustion and Propulsion*. Reston: American Institute of Aeronautics and Astronautics, 2007.
- [9] Sutton, G, and O. Biblarz. *Rocket Propulsion Elements*. 7th Edition. New York: Jonh Wiley & Sons, INC., 2001.
- [10] Dino, Jonas. "NASA TESTS ENVIRONMENTALLY FRIENDLY ROCKET FUEL." NASA.
http://www.nasa.gov/centers/ames/news/releases/2003/03_04AR.html (accessed 11/9/2007, 2007).
- [11] eAc. (2003). Retrieved 11/17/2007, 2007, from <http://www.hybrids.com/mulv.html>
- [12] Leonard, David. "Exclusive: Rules Set for \$50 Million 'America's Space Prize'" Space News. 20 Feb. 2008 <http://www.space.com/spacenews/businessmonday_bigelow_041108.html>.
- [13] Kudriavtsev, Vladimir, Brian Feeney, Max Buneta, James Porcher, Asier Ania, Michael Trauttmandorff, Ta-Liang Hsu, Marek Krzeminski, and Kalman Rooz. "DA VINCI X-PRIZE SPACE PROJECT: MISSION ANALYSIS". AAAF Paper #39-2.
http://www.davinciproject.com/documents/EADSpaper_n39_2.pdf

- [14] ERG Materials and Aerospace Corp. January 31, 2008.
<http://www.ergaerospace.com/img/projectGallery/rocketInsert.jpg>
- [15] Changjin Lee, Yang Na, Jae-Woo Lee, Yung-Hwan Byun, Effect of induced swirl flow on regression rate of hybrid rocket fuel by helical grain configuration. Aerospace Science and Technology (2007) pg 68-76.
- [16] Evens, Brian and Favorito, N. "Study of Solid Fuel Burning-Rate Enhancement Behavior in an X-ray translucent Hybrid Rocket Motor" AIAA, Virginia 2005.
- [17] Moor, Todd. www.Skyrippersystems.com, 2007.
- [18] C. Carmicino and A. Russo Sorge, Influence of a Conical Axial Injector on Hybrid Rocket Performance. Journal of Propulsion and Power, Vol 22, No. 5, September-October 2006.
- [19] C. Carmicino , A. Russo Sorge, Performance comparison between two different injector configurations in a hybrid rocket. Aerospace Science and Technology 11 (2007) 61–67.
- [20] mikroElektronika " Introduction to<http://www.mikroe.com/en/books/picbook/1_chapter.htm>
- [21] Zakirov, V.a, Ke Wan, Fan-Li Shan, Hai-Yun Zhang, and Lu-Ming Li. "Hybrid Rocket Motor." Tsinghua Space Center. Tsinghua University. 21 Feb. 2008
<http://www.tsinghua.edu.cn/docsn/lxx/mainpage/a/Web/index_files/page0008.htm>.
- [22] Fluent Incorporated. *Fluent 6.1 documentation: Tutorial Guide*.
- [23] Occupational Safety and Health Administration, "Paraffin Wax Fumes," 2007,
<http://www.osha.gov/dts/sltc/methods/partial/pv2047/2047.html>.
- [24] Microchip, "PIC18F87J50 Family Data Sheet 64/80-Pin High-Performance, 1-Mbit Flash USB Microcontrollers with nano Watt Technology" 2007
<http://ww1.microchip.com/downloads/en/DeviceDoc/39775b.pdf>

7 Appendices

7.1 Appendix A: LabVIEW Setup for Fin Flutter

Testing took place in three steps, the first was to setup up the virtual instrumentation, the next was to prepare the testing surface, or the fins, and the last step was to calibrate the instrument, run the program and store the data. The virtual instruments for this lab mimicked the function of an oscilloscope.

The next step was to set up the fin. This was done by carefully computing the geometric center of the desired shape, lightly sanding the surface and super-gluing a strain gauge to the desired location in the proper orientation to achieve the proper bridge. When this was accomplished the virtual instrument was powered on, the calibration procedure followed and data collected and stored for future analysis.

The VI in Figure 99: Sample VI Visualizer has controls to adjust the time base and the voltage per division, changing these dials resulted in the change of the resolution of the different signal inputs. This also allowed for each square to have an exact value for time and voltage. When the input signal from a strain gauge was measured the change of voltage was seen as a function of time. This was the data collected for each fin at different intervals.

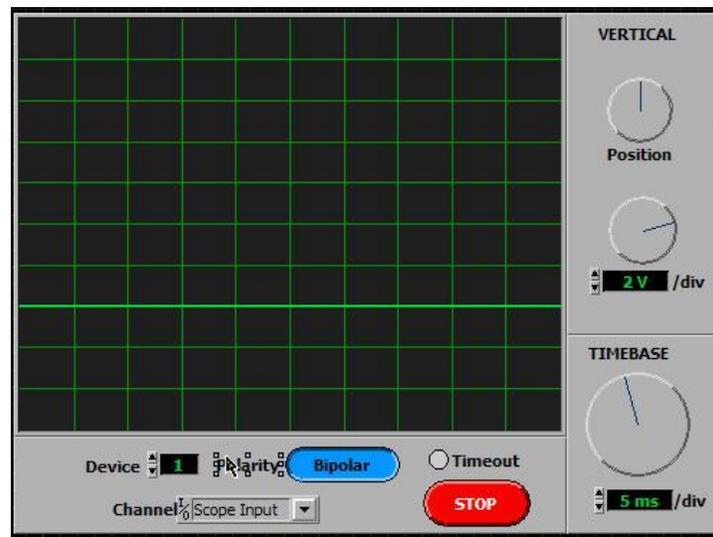


Figure 99: Sample VI Visualizer
© 2008 National Instruments Corporation

Figure 100 is a second oscilloscope VI, which left the display length as the time constant this set the display to a corresponding constant period of time, which also allowed for a choice of sampling rates. With the value of TBD selected, the same measurements were performed using the strain gauge. This data was then plotted in a similar fashion as done with the first example VI shown in Figure TBD.

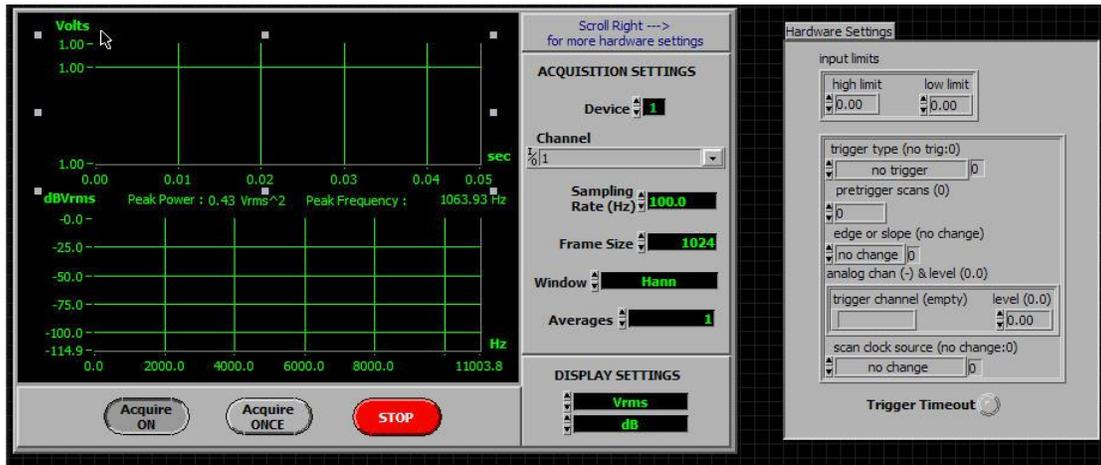
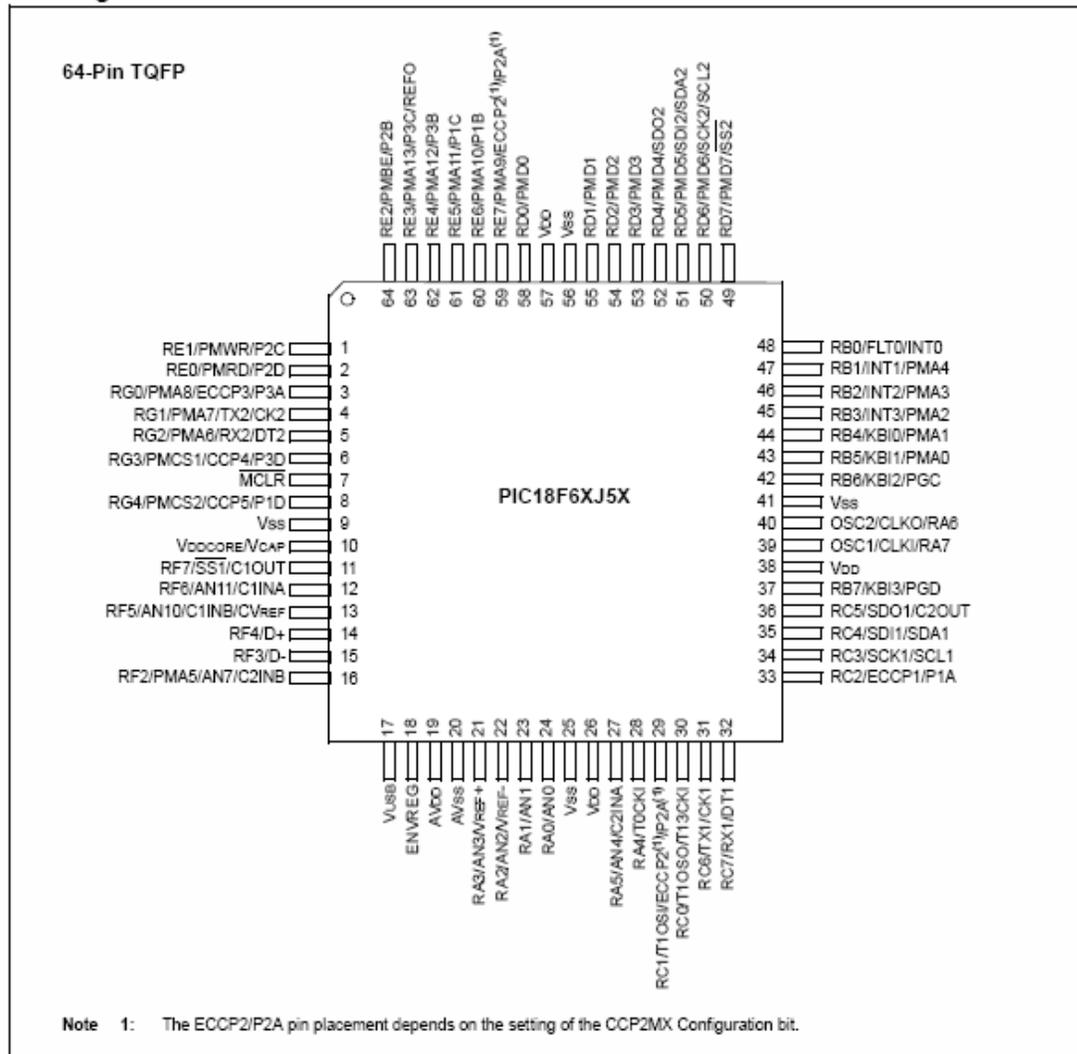


Figure 100 : Oscilloscope, with recording function VI
© 2008 National Instruments Corporation

7.2 Appendix B: Pin out of 18F87J50 PIC microprocessor

Pin Diagrams



7.3 Appendix C: Test Procedure Checklist

Hybrid Rocket Motor Test Procedure: Full Testing Checklist

Date: _____
Test Number: _____
Grain Number: _____
Injector Number: _____
Controller Manager: _____
Computer Manager: _____
Attendance: _____

PreTest: (to start once on site of testing facility)

- Weight of Tank before Testing: _____ g; Mass motor: _____ g
- Secure Test stand
- Place instrumentation cart at safety distance (10 ft)
- Boot computer and DAQ
- Connect solenoid controller to relay station
- Power up relay station
- Connect Sensors to Sensor power box
- Connect Power Supply to Sensor power box at 28.00V, 1A
- Power on Sensor power box
- Connect Oxidizer flow system lines to motor on test stand
- Ensure computer and DAQ are working properly by pressing on the load cell and verifying load.
- Inspect the motor for assembly
 - Check all fittings for grease or Teflon
 - Check to verify that o-rings are not visible and the casing is properly installed (If used)
 - Check the torque of the boilerplate connecting bolts (In a radial pattern) to be tight using standard wrench.
- Set up cameras and other visual devices.
 - Set for record and display a sheet of paper with the test number and date at the start.
- Take pictures of the setup for records.

Hot fire Testing:

- Ensure the go ahead for hot fire test from lab manager.
- Clear the area from any spectators or members not part of the startup procedure.
- The test area should now be clear of any flammable items or persons.
- Connect an igniter to the control box leads,
Place igniter in rocket motor
 - Make sure head is recessed from head of the injector.

- Tape the igniter away from the nozzle.
- Open the N₂O cylinder valve
- Open the N₂O Safety Valve
- Inspect for leaks. (If leaking sound is heard close tank, empty line and inspect fittings.)
- Move all away from testing area. (no personnel should be near the motor at this point)
- Arm the Controller
- Make a loud announcement that the test of a hybrid rocket motor will be commencing monetarily
- Ensure that the control program as the correct values.

Burn time: _____ PreFlow time: _____

- Turn the safety key to ON from the control box.
- Press the start button on the computer controlled ignition.

Time of Run test: _____

NOTE: IF YOU NOTICE A PROBLEM OR AN ANOMOLY ON THE DATA, PRESS ABORT. DATA WAS COLLECTED AND ANALYSIS CAN STILL TAKE PLACE.

Post Firing:

- Close Ball Valve at N₂O tank.
- Run Vent of N₂O Delivery Lines.
- Close valve at N₂O tank.
- Turn off the N₂O tank.
- Wait for hardware to cool. CAUTION: THE MOTOR IS VERY HOT, KEEP AWAY.
- Disassemble Test hardware and return to MQP lab.
- Weight of Tank after Testing: _____ g Mass motor: _____g

7.4 Appendix D: Fire Lab Safety Forms

Fire Science Laboratory
Sign-up and Summary Sheet

The goal of this Sign-up and Summary Sheet is to insure all students working in the Fire Science Laboratory have reviewed the Standard Operating Procedures, attended a Familiarization Tour and reviewed the Safety Guidelines. Also, to insure the student has reviewed either the appropriate Operating Procedures or developed an Experimental Plan for non-typical testing and understands all hazards and precautions to be taken.

Student Name: Theo McDonald

Project Partners Names	Buhler, Jared;	Renaud, Patrick
	Yocom, Megan	Fleming, Siobhan
	Cummings, Nicholas	Coverstone, Troy
	Huleatt, Thomas	

Advisor name: Prof. Blandino

Experiment Name and Number: _____

Hazards Identified:

Cold liquefied gas, hot exhaust gas, Possible over pressurization

Precautions to be understood by all involved (Attach experimental plan if non-standard testing):

Precautions are included in the testing plan attached. Precautions will be explained by the students and lab manager.

Advisor Signature Date

Cross Check Signature Date

(must be Professor Nicholas A. Dembsey if advisor is not an FPE Professor.)

I have read and will observe the Fire Science Laboratory Safety Guidelines. For non-typical testing, I will follow the procedures given in my approved Experimental Plan. I have attended a Fire Science Laboratory Familiarization Tour and feel comfortable with my responsibilities in the Fire Science Laboratory. I will ask my advisor, the lab staff or the Faculty Coordinator if I am unsure about anything in the Fire Science Laboratory.

Student Signature Date

Fire Science Laboratory
Short Form for Short Tests

The purpose of this form is to provide a simple way for students to develop an Experimental Plan for short, uncomplicated tests rather than using the Experimental Plan Template. The numbers listed below match the steps in How to Develop an Experimental Plan. You should read that first to learn how to develop a plan and then use the following format to present it. Short Form in Word Format.

Background

Reason for doing experiment:

This experiment will be a test of Paraffin wax/N₂O Hybrid rocket motor combustion. As cost and performance seem to be dueling aspects of rocketry, we are studying propulsion system where the fuel and oxidizer are inert and are unable to sustain burning unless initiated. Our study will primarily focus on the study of new mixing methods between the fuel and oxidizer once the combustion process has started.

Objective

Data/information you trying to obtain:

Performance of each rocket motor injector type that we design. All the data we are planning to collect will come from sensors on our test stand; as such we will not need to use any measuring devices from the Fire Science Lab.

Process flow and instrumentation diagram (PID)

Draw your PID:

Attached on following page

PID components

List components of PID: Exhaust hood in back room, MQP provided parts: Blast shield, N₂O cylinder, Thruster on Test Stand, Blast walls (sandwich of plywood, sand and plywood).

Safety issues

List materials and chemicals to be used, required use and hazards:

N₂O- Liquefied gas, used as oxidizer in test, however at standard air temp this chemical is not an oxidizer. It must be decomposed before it will initiate burning.

Paraffin Wax fuel (with additives) – Non toxic fuel used. Inert in nature, this will not sustain burning when the oxidizer flow is cut.

MSDS of both will be available at both testing location and storage location

List Protective Personal Equipment to be worn to protect against those hazards:

Whenever working with the N₂O tank, the liquid is extremely cold during evaporation, thus gloves must be worn when handling fittings on the N₂O cylinder, and eye protection when delivery lines are pressurized.

Full face masks and hearing protection will be used while in the testing room if the motor is running, if personnel are not in the room while hot fire testing, hearing protection and eye protection will still be worn.

Failure possibilities

List failure possibilities of EACH of the PID components and how you will minimize these: Thruster Failure: High Safety Factor in design as well as inspection of parts before testing occurs.

N₂O Delivery System Failure (burst in flow lines): Inspection before use, proper tightening of fittings.

Safety Devices: (Blast Shield and containment shield) Inspection before use and proper placement and secure attachment and weight.

Checklist

Write a checklist which encompasses all your procedures: Include pre-test, test and post-test activities Use additional/separate sheet if needed:

See attached sheet.

Checklist specifics

Plan to inform others of activities: Lab Manager Randy Harris will take this responsibility

Number of people needed to run: One for controller and one for data collection

Emergency shutdown

List events that necessitate an immediate shutdown:

Over pressurization of the combustion chamber, radical or highly progressive thrust monitored on the testing laptop, breach of combustion chamber

& 11. Apparatus checkout

How will you know if all the components work properly:

All components will be inspected and quickly testing for functionality before testing, if this is not done so, we could have poor data or a motor failure.

Team review

Explain how all team members will know what is going on: All team members will be following the checklist as well, to ensure that all sections are followed.

& 14. Pre-test and test

Explain how you will know the information obtained is valid (i.e., reasonable or makes sense):

The data will be collected on our own DAQ and computer, this will be providing live results that will be easy to tell whether the motor performed as we expected.

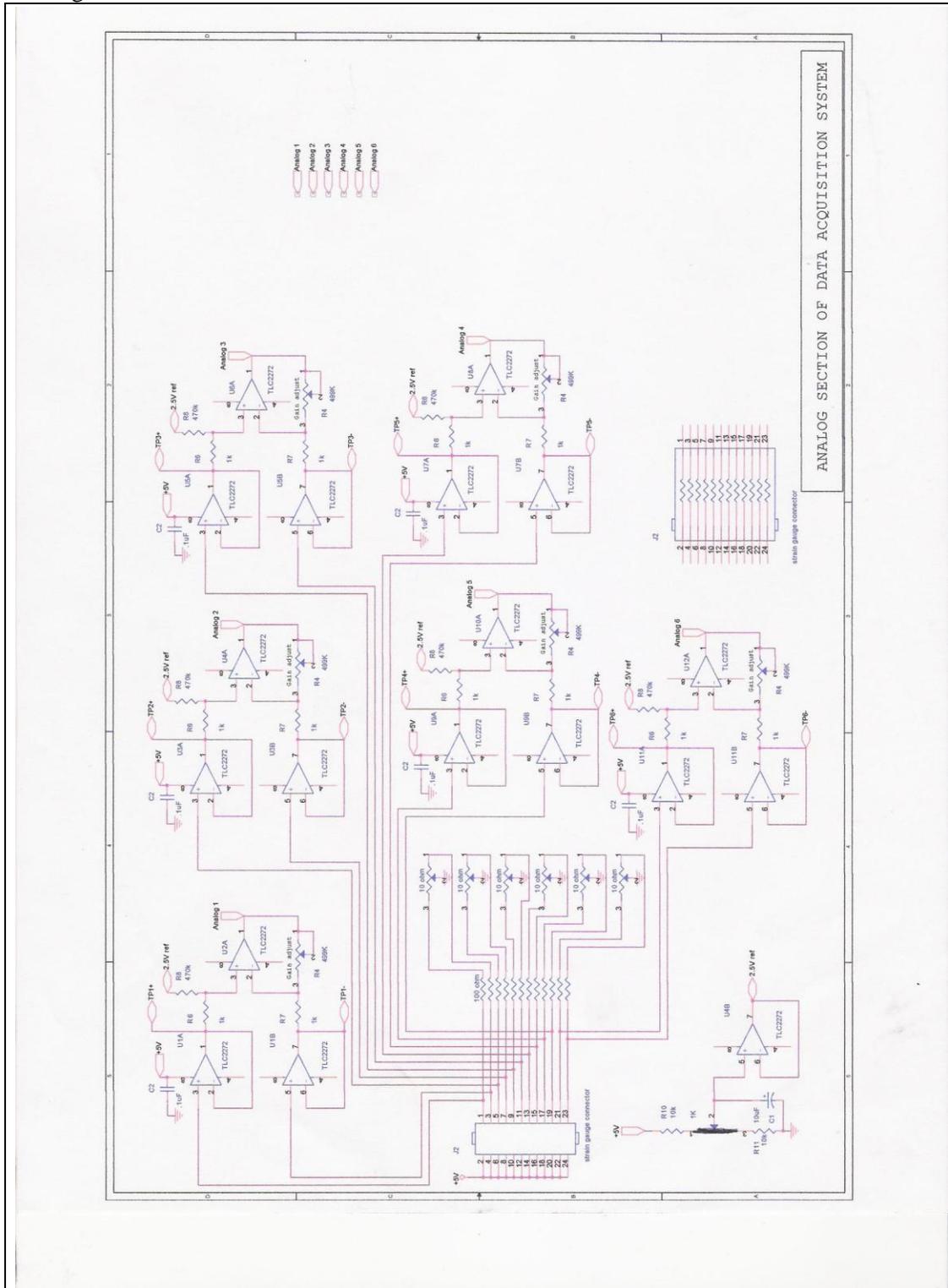
Post-test

List what/how you will clean post-test:

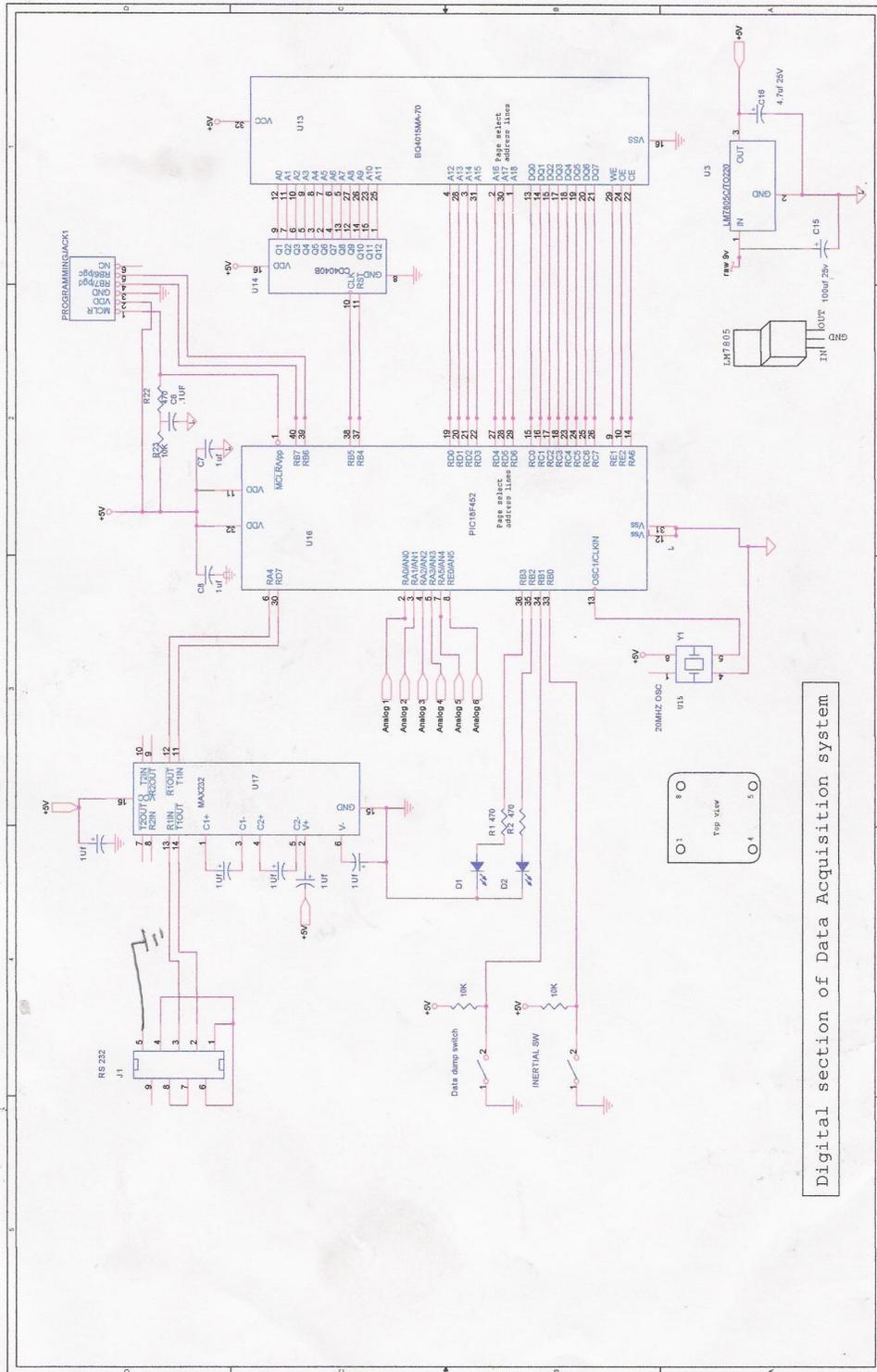
We will return our equipment to our own lab and clean any possible exhaust residue that could be on the floor directly under the motor.

7.5 Appendix E: FCDAQ Wiring Schematics.

Analog Schematic:



Digital Schematic:



Digital section of Data Acquisition system

7.6 Appendix F: FCDAQ Software Coding

Code written for use of FCDAQ with wind tunnel to simulate flight:

```
;*****
;MQP PROJECT
;MEASURING THE VIBRATION FREQUENCY AND AMPLITUDE OF FIN WARP ON ROCKET
;*****
; A NON VOLATILE MEMORY CHIP WITH BUILT IN DATA RETENTION BATTERY IS COUPLED
WITH A 20MHZ
;MICROCONTROLLER TO ACQUIRE STRAIN DATA FROM TWO PLACES ON EACH OF THE 3 FINS
;WIND TUNNEL TEST HAVE DETERMINED THE RESONANT FREQUENCY OF THE FINS TO BE
APROXIMATELY 100HZ
; THE DATA IS ACQUIRED ON ALL SIX STRAIN QUAGES IN EACH ACQUISITION PASS.
; THE DATA IS THEN STORED IN SIX PAGES OF MEMORY WITH 65535 DATA POINTS EACH
;THE SCAN FREQUENCY IS ADJUSTABLE,WITH CONSTANT"ACQDELAY" AND THE ACQUISITION
IS STARTED WHEN AN ACCELERATION SENSOR IS TRIGGERED BY THE BLASTOFF
;AN RS232 SERIAL CONNECTION PROVIDES A MEANS TO GET THE DATA INTO A PC, WHERE
LABVIEW WILL ALLOW HIGH LEVEL ANALISYS OF THE DATA.
;
;*****

    list p=18f452

    ; Include file, change directory if needed
    include "P18F452.inc"
;Port A assignments:    <---- in    -----> out

;RA0/AN0    <-----Analog        strain guage 1
;RA1/AN1    <-----Analog        Strain guage 2
;RA2/AN2    <-----Analog        Strain guage 3
;RA3/AN3    <-----Analog        Strain guage 4
;RA4/TOCK1  <-----DIN            SERIAL IN
;RA5/AN4    <-----ANALOG        Strain guage 5
;RA6        ----->DOUT        Memory CE
;
;

;Port B assignments:    <---- in    -----> out

;RB0        ----->DOut        NC
;RB1        ----->DOut        NC
;RB2        ----->DOut        NC
;RB3        ----->DOut        NC
;RB4        ----->DOut        COUNTER RESET
;RB5        ----->DOut        COUNTER CLOCK
;RB6        ----->DOut        PGC PROGRAMMER
;RB7        ----->DOut        PGD PROGRAMMER
;

;Port C assignments:    <---- in    -----> out
;RC0        <----->DOut        Data port 0
;RC1        <----->DOut        Data port 1

;RC2        <----->DOut        Data port 2
```



```

;RC3          <----->DOut          Data port 3

;RC4          <----->DOut          Data port 4
;RC5          <----->DOut          Data port 5
;RC6          <----->DOut          Data port 6
;RC7          <----->DOut          Data port 7
;
;Port D assignments:
;RD0          ----->DOut          Address A12
;RD1          ----->DOut          Address A13
;RD2          ----->DOut          Address A14
;RD3          ----->DOut          Address A15
;RD4          ----->DOut          Address A16
;RD5          ----->DOut          Address A16
;RD6          ----->DOut          Address A18
;RD7          ----->DOut          SERIAL OUT
;
;Port E assignments:
;RE0/AN5      <-----ANALOG        Strain guage 6
;RE1/AN6      ----->DOut          MEMORY WE
;RE2/AN7      ----->DOut          MEMORY OE

;Variables
      cblock 0x000
      FLAG1
;FLAG1 BIT0 =      END OF MEMORY LO= OK      HI = END
;FLAG1 BIT1 =
;FLAG1 BIT2 =
;FLAG1 BIT3 =
;FLAG1 BIT4 =
;FLAG1 BIT5 =
;FLAG1 BIT6 =
;FLAG1 BIT7 =
ADDINTLO          ;HOLDS LOCAL LOW 8 BITS OF ADDRESS
ADDINTMID         ;HOLDS LOCAL MID BITS. THE HIGH NIBBLE HOLDS
A12,A13,A14,A15 FOR EXTERNAL RAM
ADDPAGE           ;HOLDS CURRENT PAGE BITS OF ADDRESS A16,A17,A18
DELAYLO           ;LOW REGISTER FOR COUNTER
DELAYMID         ;MID REGISTER FOR COUNTER
DELAYHI          ;HIGH REGISTER FOR COUNTER
STRAIN1RES       ;8 BIT RESULTS STRAINGUAGE1
STRAIN2RES       ;8 BIT RESULTS STRAINGUAGE2
STRAIN3RES       ;8 BIT RESULTS STRAINGUAGE3
STRAIN4RES       ;8 BIT RESULTS STRAINGUAGE4
STRAIN5RES       ;8 BIT RESULTS STRAINGUAGE5
STRAIN6RES       ;8 BIT RESULTS STRAINGUAGE6
TXTCNTR          ;COUNTS BYTE OUTPUT
Xmit_Byte        ;holds byte to xmit
Rcv_Byte         ;holds received byte
Bit_Cntr         ;bit counter for RS232
Delay_Count      ;delay loop counter
TEMP1            ;TEMPORARY REGISTER
FAKEAD           ;COUNTER FOR GENERATING FAKE ANALOG INPUT
BINDALO
BINAHI
BCD1

```

```

BCD2
COUNT
TMP1
DIGIT100
DIGIT10
DIGITUNITS
Digit
READTSTCOUNT
    endc

```

```

;*****Text replacements

```

```

#DEFINE          WRITEENABLE          PORTE,1          ;WRITE STROBE FOR
MEMORY
#DEFINE          CHIPENABLE           PORTA,6           ;CHIP ENABLE FOR
MEMORY
#DEFINE          OUTPUTENABLE         PORTE,2           ;OUTPUT ENABLE
FOR MEMORY
#DEFINE          MCOUNTRESET         PORTB,4           ;RESET FOR CD4040
COUNTER RESET WHEN HI
#DEFINE          MCOUNTCLOCK         PORTB,5           ;INCREMENTS
MEMORY COUNTER ON LOW TRANSITION
#DEFINE          ADGO                  ADCON0,2          ;AD START BIT
#DEFINE          EOMFLAG                FLAG1,0          ;SIGNALS
FULL MEMORY
#DEFINE          INERTIASW             PORTB,0           ;STARTS THE DATA
ACQUISITION
#DEFINE          DATADUMPSW           PORTB,1           ;STARTS THE DATA
SERIAL UPLOAD
#DEFINE          LED1                  PORTB,2           ;
#DEFINE          LED2                  PORTB,3           ;
#DEFINE          SERIN                 PORTA,4
#DEFINE          SEROUT                PORTD,7

```

```

;*****CONSTANTS*****
ADTIMEDELAY      EQU          D'240'          ;DELAYAD
TIMEDELAY        EQU          D'255'
MEMDELAY         EQU          D'255'
CLOCKDELAY      EQU          D'255'          ; counter is up count so 255=
2.8uS

```

```

; 0= 206.8 uS each
count = .8uS
ACQDELAY        EQU          D'1'            ;EACH COUNT LESS THAN 255 HERE =
155 uS 249 = 1.086 MS 1=.0397 SEC
ANALOGCH0       EQU          B'10000001'
ANALOGCH1       EQU          B'10001001'
ANALOGCH2       EQU          B'10010001'
ANALOGCH3       EQU          B'10011001'
ANALOGCH4       EQU          B'10100001'
ANALOGCH5       EQU          B'10101001'

```

```

; Start at the reset vector
org          0x000

```

```

        goto Main

INIT    org          0x020

        call SER_INIT          ;initialise serial port

        clrf porta
        clrf PORTB           ;Clear ports
        clrf portc
        clrf portd
        movlw b'00011111'
        MOVWF TRISA
        MOVLW B'00000011'
        MOVWF TRISB          ;PORTB all outputs, DATA PORT FOR MEMORY START AS

OUTPUT
        MOVLW B'11111111'
        MOVWF TRISC          ;PORT C ALL INPUTS (MEMORY DATA BUSS)
        CLRF TRISD           ;CLEAR ADDRESS PORT
        MOVLW B'00000001' ;PORT E BIT 0 IS ANALOG IN
        MOVWF TRISE
        CLRF FLAG1
        CLRF ADDINTLO
        CLRF ADDINTLO
        ;SET UP MEMORY CONTROL LINES TO INIT CONDITION
        BSF          CHIPENABLE
        BSF          WRITEENABLE
        BSF          OUTPUTENABLE
        CLRF DELAYLO
        CLRF DELAYMID
        CLRF DELAYHI
        CLRF INTCON2
        BSF          MCOUNTRESET          ;RESETS THE CD4040 COUNTER
        BSF          MCOUNTCLOCK         ; GETS CLOCH READY TO COUNT
        BCF          MCOUNTRESET         ;FINISH RESET READY TO COUNT
        movlw B'10000001' ;Fosc/32, A/D enabled
        movwf ADCON0
        movlw B'00001001' ;left justify,6 analog channels
        movwf ADCON1          ;this places the 8 bit answer into ADRESHI

        movlw B'11000111' ;TMR0 prescaler, 1:256
        movwf T0CON
        CLRF ADDINTLO        ;CLEAR INTERNAL MEMORY COUNTERS
        CLRF ADDINTMID
        MOVLW B'00000110'
        MOVWF PORTE
        CLRF FAKEAD

        RETURN
;*****
;MAIN LOOP OF ACQUISITION
;*****
**
Main

        call init

```

```

WAITSTART    ;BTFSS      INERTIASW          ;CHECK THE INERTIA SWITCH HI=NOT
MOVING, LOW = ROCKET HAS BLASTED OFF
              ;GOTO MAINLOOP              ;IF INERTIAL SW IS
              BTFSS DATADUMPSW
              GOTO  MAINLOOP
              GOTO  WAITSTART
MAINLOOP     BTFSS DATADUMPSW
              GOTO  mainloop
              CALL  ANALOGIN                ;READ THE 6 STRAIN
GUAGES AND STORE RESULTS
              call  LIVE_RS232
              GOTO  WAITABIT

              CALL  ANALOGOUT                ;SEND THE DATA TO THE
MEMORY CHIP
              CALL  ADDRESSGENERATOR        ;INCREMENT BASE MEMORY
ADDRESS TO NEXT
              BTFSC EOMFLAG                  ;CHECK IF MEMORY
FLAG SHOWS THAT WE HAVE REACHED THE END OF MEMORY
              GOTO  FINISHED                ;FINISHED WITH
ACQUISITIONJUMP TO BIT SCAN ROUTINE AND JUST WAIT
                                                    ;WE NEED TO
SEND THE 4 MS BITS OF THE ADDRESS OUT TO RD0,RD1,RD2,RD3, SO WE MUST
                                                    ;SWAP THE
UPPER AND LOWER NIBBLES OF
                                                    ;THE
ADDINTMID COUNT, AND CLEAR THE PAGE BITS

              CALL  OUTADDDHIGH             ;FINISHES THE ADDRESS UPDATE OUTPUT

WAITABIT     CALL  READWAIT                  ;DELAY BETWEEN EACH SCAN OF
SENSORS. WE WANT TO DELAY ENOUGH SO THAT THE
                                                    ;WE DONT
GET MULTIPLE SCANS ON EACH PRESS THIS IS ABOUT 400 mS
              CALL  READWAIT
              CALL  READWAIT
              CALL  READWAIT
              CALL  READWAIT
              CALL  READWAIT
              CALL  READWAIT
              CALL  READWAIT
              CALL  READWAIT
              CALL  READWAIT
              goto  MAINLOOP                ;Do it again

FINISHED     BTG      LED1
              CALL  READLOOP
              CALL  READLOOP
              CALL  READLOOP
              CALL  READLOOP
              CALL  READLOOP
              GOTO  FINISHED

```

```

;*****
*****
; DELAYAD ROUTINE #1
; counter is up count so 255= 2.8uS
; 0= 206.8 uS each count = .8uS
;*****
*****
DELAYAD          MOVLW ADTIMEDELAY
                MOVWF DELAYLO

LOWCOUNT
    INCFSZ      DELAYLO
    GOTO LOWCOUNT
    RETURN

DELAY           MOVLW TIMEDELAY
                MOVWF DELAYLO
                GOTO LOWCOUNT

DELAYMEM        MOVLW MEMDELAY
                MOVWF DELAYLO
                GOTO LOWCOUNT

DELAYADDRESSCLOCK MOVLW CLOCKDELAY
                MOVWF DELAYLO
                GOTO LOWCOUNT

;*****
*****
READWAIT        MOVLW ACQDELAY
                MOVWF DELAYMID
READLOOP        CALL LOWCOUNT
                INCFSZ      DELAYMID
                GOTO READLOOP
                RETURN

;*****
;
; EACH PRESS OF THE READ BUTTON WILL ACQUIRE THE SIX STRAIN SIGNALS, AND SEND
THE RESULTS TO THE TERMINAL
;
;*****
LIVE_RS232      MOVF STRAIN1RES,W
                MOVWF      BINDALO                ;PUT DATA IN
BINDALO
                CALL BCDCNV                        ;CONVERT TO PACKED BCD
                CALL SENDDATA
                MOVF STRAIN2RES,W
                MOVWF      BINDALO                ;PUT DATA IN
BINDALO
                CALL BCDCNV                        ;CONVERT TO PACKED BCD
                CALL SENDDATA
                MOVF STRAIN3RES,W

```

```

MOVWF      BINDALO          ;PUT DATA IN
BINDALO
CALL  BCDCNV          ;CONVERT TO PACKED BCD
CALL  SENDDATA
MOVF  STRAIN4RES,W
MOVWF      BINDALO          ;PUT DATA IN
BINDALO
CALL  BCDCNV          ;CONVERT TO PACKED BCD
CALL  SENDDATA
MOVF  STRAIN5RES,W
MOVWF      BINDALO          ;PUT DATA IN
BINDALO
CALL  BCDCNV          ;CONVERT TO PACKED BCD
CALL  SENDDATA
MOVF  STRAIN6RES,W
MOVWF      BINDALO          ;PUT DATA IN
BINDALO
CALL  BCDCNV          ;CONVERT TO PACKED BCD
CALL  SENDDATA
RETURN
;*****
;*****
; THIS SUBROUTINE GENERATES THE 19 BIT ADDRESS. The first 12 bits of address
(A0-A11) is generated by the cd4040 ripple counter.
;A12,A13,A14,A15 ARE GENERATED IN SOFTWARE COUNTER ADDINTMID
;ADDINTLO+ ADDINTMID HOLDS A COPY OF WHERE THE CD4040 COUNTER SHOULD BE
;ADDINTMID HOLDS THE REST OF THE 12 BIT CD4040 COUNT, AS WELL AS THE LAST 4
BITS OF ACTUAL ADDRESS WHICH WILL BE SENT TO
;PORT C BIT 0,1,2,3.
;*****
;*****
ADDRESSGENERATOR
      BCF      MCOUNTCLOCK      ;increments the external counter
      CALL    DELAYADDRESSCLOCK
      BSF      MCOUNTCLOCK      ;FINISH PULSE GET READY FOR NEXT
COUNT
      INCFSZ   ADDINTLO      ; Increment internal address TO MATCH
EXTERNAL
      RETURN          ;COUNTER DID NOT ROLL OVER
      INCFSZ   ADDINTMID      ;INCREMENT MID COUNTER
      RETURN
      BSF      EOMFLAG      ;SET END OF MEMORY FLAG
RETURN
;*****
;*****
;READDATA DUMP WILL GENERATE THE REQUIRED ADDRESS TO READ THE DATA BACK.
;IT WILL READ EACH FULL PAGE THEN MOVE TO THE NEXT

```

```

;ONCE THE ADDRESS IS VALID, THE DATA WILL BE ON PORT D NO OTHER CONTROLL
SIGNALS NEED TO BE CHANGED
;SINCE THE OUTPUT WILL REMAIN ACTIVE
;*****
*****
READDATADUMP
    MOVLW B'11111111'
    MOVWF TRISC          ;SET UP PORT C AS INPUT
    BCF      CHIPENABLE  ;CHIP IS SELECTED
    BSF      WRITEENABLE ;KEPT HIGH FOR READ
    BCF      OUTPUTENABLE ;ENABLES OUTPUT
    CALL    RESETADDRESS
;THE ADDRESS COUNTER SHOULD NOW BE AT ZERO, AND THE INTERNAL COUNTERS ALSO AT
ZERO, AND THE PAGE 0 DATA WILL BE ON PORTD
READPAGE0    CALL    PAGE0
              MOVF   PORTC,W          ;GET DATA INTO W
              MOVWF  BINDALO         ;PUT DATA IN BINDALO
              CALL   BCDCNV          ;CONVERT TO PACKED BCD
              CALL   SENDDATA

              CALL   PAGE1
              MOVF   PORTC,W          ;GET DATA INTO W
              MOVWF  BINDALO         ;PUT DATA IN BINDALO
              CALL   BCDCNV          ;CONVERT TO PACKED BCD
              CALL   SENDDATA

              CALL   PAGE2
              MOVF   PORTC,W          ;GET DATA INTO W
              MOVWF  BINDALO         ;PUT DATA IN BINDALO
              CALL   BCDCNV          ;CONVERT TO PACKED BCD
              CALL   SENDDATA

              CALL   PAGE3
              MOVF   PORTC,W          ;GET DATA INTO W
              MOVWF  BINDALO         ;PUT DATA IN BINDALO
              CALL   BCDCNV          ;CONVERT TO PACKED BCD
              CALL   SENDDATA

              CALL   PAGE4
              MOVF   PORTC,W          ;GET DATA INTO W
              MOVWF  BINDALO         ;PUT DATA IN BINDALO
              CALL   BCDCNV          ;CONVERT TO PACKED BCD
              CALL   SENDDATA

              CALL   PAGE5
              MOVF   PORTC,W          ;GET DATA INTO W
              MOVWF  BINDALO         ;PUT DATA IN BINDALO
              CALL   BCDCNV          ;CONVERT TO PACKED BCD
              CALL   SENDDATA

    MOVLW D'13'          ;SEND CARRIGE RETURN
    CALL   XMIT_RS232
    CALL   LOWCOUNT
    MOVLW D'10'          ;SEND LINE FEED
    CALL   XMIT_RS232
    CALL   LOWCOUNT

```

```

; THE 6 DATAPOINTS ARE NOW
UPLOADED, SO WE INCREMENT TO THE NEXT MEMORY LOCATION
    CALL ADDRESSGENERATOR
    CALL OUTADDHIGH
    INCF READTSTCOUNT
    BTFSS EOMFLAG
    GOTO READPAGE0
    CALL RESETADDRESS

WAITHERE    BTG            LED2
            CALL READLOOP
            CALL READLOOP
            CALL READLOOP
            CALL READLOOP
            CALL READLOOP
            GOTO WAITHERE

;*****
;*****
RESETADDRESS    BSF            MCOUNTRESET            ;RESETS THE CD4040 COUNTER
                BSF            MCOUNTCLOCK            ; GETS CLOCK READY TO COUNT
                BCF            MCOUNTRESET            ;FINISH RESET READY TO COUNT
                CLRF ADDINTLO                ;CLEAR LOCAL LOW 8 BITS OF ADDRESS
                CLRF ADDINTMID            ;CLEAR LOCAL MID BITS. THE HIGH NIBBLE HOLDS
A12,A13,A14,A15 FOR EXTERNAL RAM
                CALL OUTADDHIGH            ;FINISHES THE OUTPUT TO THE ADDRESS PORT CLEARS
PAGE BITS
                CLRF FLAG1
                RETURN

;*****
;*****
;TAKES ADDINTMID AND MOVES IT TO PORD D IN THE CORRECT CONFIGURATION, THIS
ALSO CLEARS THE PAGE BITS
;*****
;*****
OUTADDHIGH
    MOVFF ADDINTMID,TEMP1            ;PUT CURRENT ADDRESS HIGH COUNT
    INTO TEMP1
    SWAPF TEMP1                ;SWAP HIGH NIBBLE WITH LOW
    BSF            TEMP1,7
    MOVLW B'10001111'            ;SET UP THE "AND" MASK
    ANDWF TEMP1,W                ;W NOW HOLDS THE CORRECT 4
ADDRESS BITS IN THE LOW NIBBLE, AND THE HIGH NIBBLE IS CLEARED
    MOVWF PORTD                ;SEND TO THE PORT WHICH SETS UP
RD0,1,2,3 TO THE ADDRESS, AND CLEARS THE PAGE BITS
    RETURN

;*****
;*****
; THIS SUBROUTINE GETS THE 4 ANALOG CHANELS INTO THEIR RESPECTIVE RAM
REGISTERS

```



```

CALL DELAYAD
BSF ADGO ;SET GO BIT TO START CONVERSIONS
ADWAIT5 BTFSC ADGO ;CHECK GO BIT TO SEE IF FINISHED
GOTO ADWAIT5
MOVFF ADRESH,STRAIN6RES ;8 BIT ANSWER INTO STRAIN4RES
BTG LED1
RETURN

```

```

;*****
*****
; THIS SUBROUTINE WRITES THE 6 ANALOG CHANNELS INTO THEIR RESPECTIVE DATA
MEMORY ADDRESSES
;*****
*****

```

ANALOGOUT

```

CLRF TRISC ;SET UP DATA PORT TO OUTPUT SO WE CAN SEND DATA TO
THE MEMORY CHIP
CALL PAGE0
MOVFF STRAIN1RES,PORTC
CALL WRITE1 ;FIRST WRITE WILL BE TO PAGE 0
WHICH IS ALREADY SET
MOVFF STRAIN2RES,PORTC
CALL PAGE1
CALL WRITE1
MOVFF STRAIN3RES,PORTC
CALL PAGE2
CALL WRITE1
MOVFF STRAIN4RES,PORTC
CALL PAGE3
CALL WRITE1
MOVFF STRAIN5RES,PORTC
CALL PAGE4
CALL WRITE1
MOVFF STRAIN6RES,PORTC
CALL PAGE5
CALL WRITE1

RETURN

```

```

;*****
*****
;PAGE SWAPING ROUTINES
;*****
*****

```

```

PAGE0 BCF PORTD,4
BCF PORTD,5
BCF PORTD,6
RETURN

```

```

PAGE1 BSF PORTD,4
BCF PORTD,5
BCF PORTD,6
RETURN

```

```
PAGE2 BCF          PORTD, 4
          BSF          PORTD, 5
          BCF          PORTD, 6
          RETURN
```

```
PAGE3 BSF          PORTD, 4
          BSF          PORTD, 5
          BCF          PORTD, 6
          RETURN
```

```
PAGE4 BCF          PORTD, 4
          BCF          PORTD, 5
          BSF          PORTD, 6
          RETURN
```

```
PAGE5 BSF          PORTD, 4
          BCF          PORTD, 5
          BSF          PORTD, 6
          RETURN
```

```
;*****
*****
; WRITES DATA TO MEMORY CHIP ADDRESS IS ALREADY SETUP
; LOOK AT PAGE 9 AND 10 OF THE DATA SHEET FOR THE WRITE CYCLE TIMING
DIAGRAMS. THE TIMING DIAGRAMS TELL IT ALL
; READ IT OVER A FEW TIMES, AND BE SURE YOU UNDERSTAND ALL THE TERMS USED
; IT LOOKS LIKE THE WRITES CAN BE DONE TWO WAYS, THE CE, AND THE WE. BE
PARTICULARLY CAREFUL OF THE WARNING ABOUT THE DATA LINES BEING UNDEFINED.
;*****
*****
```

```
WRITE1  BCF          WRITEENABLE
          BCF          CHIPENABLE
          NOP
          BSF          WRITEENABLE
          BSF          CHIPENABLE
          RETURN
```

```
;*****
*****
;THE SERIAL ROUTINES WILL BE USED TO SEND COMMANDS TO THE SERIAL DATA DUMP
PART OF THE PROGRAM.
;WE NEED SOME WAY TO TELL THE CHIP TO START SENDING DATA, AND WE DONT WANT IT
TO START UNTILL WE HAVE CONFIRMED COMMUNICATION WITH THE PC, SO WE MAY AS
WELL USE THE SMART LINK TO
;GIVE US SOME UPLOAD OPTIONS. JUST LIKE WE MADE A SERIAL COMMAND FROM THE PC
TRIGGER THE SENDING OF THE MIKE & TROY MESSAGE.
;BECAUSE WE WANT TO GET ALL OF THE SENSOR DATA FOR EACH SENSOR OUT OF MEMORY
IN ONE PIECE, A NEW ADDRESS GENERATOR PROGRAM WILL NEED TO BE WRITTEN
;LOOK AT HOW I DID THE WRITE ADDRESS GENERATOR, TO GET SOME IDEAS AS TO HOW
TO DO THAT.
;*****
*****
```

```
;Serial routines
```

```
SER_INIT
```

```

BCF      TRISC,D'6'      ;set B6 as an output
BSF      TRISC, D'7'     ;set B7 as an input
BSF      PORTC, D'6'     ;set SER_OUT high
RETURN

XMIT_RS232 MOVWF  Xmit_Byte      ;move W to Xmit_Byte
           MOVLW  0x08        ;set 8 bits out
           MOVWF  Bit_Cntr
           BCF    SEROUT
           CALL   Bit_Delay
Ser_Loop   RRCF    Xmit_Byte , f      ;send one bit

           BTFSS  STATUS      , C
           BCF    SEROUT
           BTFSC  STATUS      , C
           BSF    SEROUT
           CALL   Bit_Delay
           DECFSZ Bit_Cntr , f      ;test if all done
           GOTO   Ser_Loop
           BSF    SEROUT
           CALL   Bit_Delay
           RETURN

Rcv_RS232  BTFSC  SERIN      ;wait for start bit
           GOTO   Rcv_RS232
           CALL   Start_Delay      ;do half bit time delay
           BTFSC  SERIN      ;check still in start bit
           GOTO   Rcv_RS232
           MOVLW  0x08        ;set up to read 8 bits
           MOVWF  Bit_Cntr
           CLRF   Rcv_Byte
Next_RcvBit CALL   Bit_Delay
           BTFSS  SERIN
           BCF    STATUS      , C
           BTFSC  SERIN
           BSF    STATUS      , C
           RRCF   Rcv_Byte , f
           DECFSZ Bit_Cntr , f      ;test if all done
           GOTO   Next_RcvBit
           CALL   Bit_Delay
           MOVF   Rcv_Byte, W
           RETURN

Start_Delay MOVLW  D'32'      ;0x0C
           MOVWF  Delay_Count

Start_Wait  NOP
           DECFSZ Delay_Count , f
           GOTO   Start_Wait
           RETURN

Bit_Delay  MOVLW  D'64'        ;0x18
           MOVWF  Delay_Count

Bit_Wait   NOP

```



```

UNPACKBCD      MOVLW B'00001111'      ;SET MASK
                ANDWF BCD2,W          ;AND WITH MSD IN BCD2
LSNIBBLE
                MOVWF DIGIT100        ;SAVE UNPACKED BCD 100'S
DIGIT
                MOVFF BCD1,TMP1       ;GET PACKED BCD UNITS AND
TENS IN TEMP1 FOR STRIPPING
                MOVLW B'00001111'    ;SET MASK
                ANDWF TMP1,W
                MOVWF DIGITUNITS
                SWAPF TMP1            ;SWAP UPPER NIBBLE WITH LOWER
NIBBLE
                MOVLW B'00001111'    ;SET MASK
                ANDWF TMP1,W
                MOVWF DIGIT10        ;SAVE TEN'S DIGIT
;*****
**
;LEADING ZERO SUPRESSION, ASCII CONVERSION
;*****
ZEROSUPRESSION      MOVLW D'1'        ;WILL ALWAYS OUTPUT AT
LEAST ONE DIGIT
                MOVWF COUNT
                MOVF  DIGIT100,W
                BTFSS STATUS,Z        ;TEST IF DIGIT 100 IS A ZERO
                GOTO  D100NZ
                MOVF  DIGIT10,W
                BTFSS STATUS,N        ;TEST IF DIGIT 10 IS A ZERO
                GOTO  D10NZ
                MOVF  DIGITUNITS,W
                ADDLW H'30'           ;HEX 30 IS ADDED TO CONVERT
                MOVWF DIGITUNITS
                RETURN

D100NZ
BE SENT            MOVLW D'3'          ;MSD NOT ZERO SO ALL 3 DIGITS WILL

                MOVWF COUNT

                MOVF  DIGIT100,W
                ADDLW H'30'           ;HEX 30 IS ADDED TO CONVERT
                MOVWF DIGIT100

                MOVF  DIGIT10,W
                ADDLW H'30'           ;HEX 30 IS ADDED TO CONVERT
                MOVWF DIGIT10

                MOVF  DIGITUNITS,W
                ADDLW H'30'           ;HEX 30 IS ADDED TO CONVERT
                MOVWF DIGITUNITS
                RETURN

D10NZ
SENT            MOVLW D'2'          ;MSD NOT ZERO SO ALL 3 DIGITS WILL BE

                MOVWF COUNT

                MOVF  DIGIT10,W

```

```

ADDLW H'30'          ;HEX 30 IS ADDED TO CONVERT
MOVWF DIGIT10

MOVF  DIGITUNITS,W
ADDLW H'30'          ;HEX 30 IS ADDED TO CONVERT
MOVWF DIGITUNITS
RETURN

;*****
;*****
;SEND 3,2, OR 1 DIGIT TO SERIAL PORT
;*****
;*****

SENDDATA    MOVLW D'3'
             CPFSEQ      COUNT          ;IS THE COUNT 3?
             GOTO  CHECK42
             MOVF  DIGIT100,W
             CALL  XMIT_RS232  ;SEND 100'S
             MOVF  DIGIT10,W
             CALL  XMIT_RS232  ;SEND 10'S
             MOVF  DIGITUNITS,W
             CALL  XMIT_RS232  ;SEND UNITS
             MOVLW D'44'      ; COMMA
             CALL  XMIT_RS232  ;SEND COMMA
             RETURN

CHECK42     MOVLW D'2'
             CPFSEQ      COUNT          ;IS THE COUNT 3?
             GOTO  OUT1
             MOVF  DIGIT10,W
             CALL  XMIT_RS232  ;SEND 10'S
             MOVF  DIGITUNITS,W
             CALL  XMIT_RS232  ;SEND UNITS
             MOVLW D'44'      ; COMMA
             CALL  XMIT_RS232  ;SEND COMMA
             RETURN

OUT1       MOVF  DIGITUNITS,W
             CALL  XMIT_RS232  ;SEND UNITS
             MOVLW D'44'      ; COMMA
             CALL  XMIT_RS232  ;SEND COMMA
             RETURN

end

```

Code written for use of FCDAQ with Flight Test Vehicle

```
;*****
;MQP PROJECT
;MEASURING THE VIBRATION FREQUENCY AND AMPLITUDE OF FIN WARP ON ROCKET
;*****
; A NON VOLITILE MEMORY CHIP WITH BUILT IN DATA RETENTION BATTERY IS COUPLED
WITH A 20MHZ
;MICROCONTROLLER TO ACQUIRE STRAIN DATA FROM TWO PLACES ON EACH OF THE 3 FINS
;WIND TUNNEL TEST HAVE DETERMINED THE RESONANT FREQUENCY OF THE FINS TO BE
APROXIMATELY 100HZ
; THE DATA IS ACQUIRED ON ALL SIX STRAIN QUAGES IN EACH ACQUISITION PASS.
; THE DATA IS THEN STORED IN SIX PAGES OF MEMORY WITH 65535 DATA POINTS EACH
;THE SCAN FREQUENCY IS ADJUSTABLE,WITH CONSTANT"ACQDELAY" AND THE ACQUISITION
IS STARTED WHEN AN ACCELERATION SENSOR IS TRIGGERED BY THE BLASTOFF
;AN RS232 SERIAL CONNECTION PROVIDES A MEANS TO GET THE DATA INTO A PC, WHERE
LABVIEW WILL ALLOW HIGH LEVEL ANALISYS OF THE DATA.
;
;*****

    list p=18f452

    ; Include file, change directory if needed
    include "P18F452.inc"
;Port A assignments:    <---- in    -----> out

;RA0/AN0    <-----Analog        Strain guage 1
;RA1/AN1    <-----Analog        Strain guage 2
;RA2/AN2    <-----Analog        Strain guage 3
;RA3/AN3    <-----Analog        Strain guage 4
;RA4/TOCK1  <-----DIN            SERIAL IN
;RA5/AN4    <-----ANALOG        Strain guage 5
;RA6        ----->DOUT          Memory CE
;
;

;Port B assignments:    <---- in    -----> out

;RB0        ----->DOut          NC
;RB1        ----->DOut          NC
;RB2        ----->DOut          NC
;RB3        ----->DOut          NC
;RB4        ----->DOut          COUNTER RESET
;RB5        ----->DOut          COUNTER CLOCK
;RB6        ----->DOut          PGC PROGRAMMER
;RB7        ----->DOut          PGD PROGRAMMER
;
;Port C assignments:    <---- in    -----> out
;RC0        <----->DOut          Data port 0
;RC1        <----->DOut          Data port 1

;RC2        <----->DOut          Data port 2
;RC3        <----->DOut          Data port 3

;RC4        <----->DOut          Data port 4
;RC5        <----->DOut          Data port 5
```



```

;RC6          <----->DOut          Data port 6
;RC7          <----->DOut          Data port 7
;
;Port D assignments:
;RD0  ----->DOut          Address A12
;RD1  ----->DOut          Address A13
;RD2  ----->DOut          Address A14
;RD3  ----->DOut          Address A15
;RD4  ----->DOut          Address A16
;RD5  ----->DOut          Address A16
;RD6  ----->DOut          Address A18
;RD7  ----->DOut          SERIAL OUT
;
;Port E assignments:
;RE0/AN5     <-----ANALOG        Strain guage 6
;RE1/AN6     ----->DOut          MEMORY WE
;RE2/AN7     ----->DOut          MEMORY OE

;Variables
    cblock 0x000
    FLAG1
;FLAG1 BIT0 =      END OF MEMORY LO= OK      HI = END
;FLAG1 BIT1 =
;FLAG1 BIT2 =
;FLAG1 BIT3 =
;FLAG1 BIT4 =
;FLAG1 BIT5 =
;FLAG1 BIT6 =
;FLAG1 BIT7 =
ADDINTLO          ;HOLDS LOCAL LOW 8 BITS OF ADDRESS
ADDINTMID         ;HOLDS LOCAL MID BITS. THE HIGH NIBBLE HOLDS
A12,A13,A14,A15 FOR EXTERNAL RAM
ADDPAGE           ;HOLDS CURRENT PAGE BITS OF ADDRESS A16,A17,A18
DELAYLO           ;LOW REGISTER FOR COUNTER
DELAYMID          ;MID REGISTER FOR COUNTER
DELAYHI           ;HIGH REGISTER FOR COUNTER
STRAIN1RES        ;8 BIT RESULTS STRAINGUAGE1
STRAIN2RES        ;8 BIT RESULTS STRAINGUAGE2
STRAIN3RES        ;8 BIT RESULTS STRAINGUAGE3
STRAIN4RES        ;8 BIT RESULTS STRAINGUAGE4
STRAIN5RES        ;8 BIT RESULTS STRAINGUAGE5
STRAIN6RES        ;8 BIT RESULTS STRAINGUAGE6
TXTCNTR           ;COUNTS BYTE OUTPUT
Xmit_Byte         ;holds byte to xmit
Rcv_Byte          ;holds received byte
Bit_Cntr          ;bit counter for RS232
Delay_Count       ;delay loop counter
TEMP1             ;TEMPORARY REGISTER
FAKEAD            ;COUNTER FOR GENERATING FAKE ANALOG INPUT
BINDALO
BINDAHI
BCD1
BCD2
COUNT
TMP1
DIGIT100

```

```

DIGIT10
DIGITUNITS
Digit
READTSTCOUNT
    endc

```

```

;*****Text replacements

```

```

#DEFINE          WRITEENABLE          PORTE,1          ;WRITE STROBE FOR
MEMORY
#DEFINE          CHIPENABLE           PORTA,6           ;CHIP ENABLE FOR
MEMORY
#DEFINE          OUTPUTENABLE         PORTE,2           ;OUTPUT ENABLE
FOR MEMORY
#DEFINE          MCOUNTRESET         PORTB,4           ;RESET FOR CD4040
COUNTER RESET WHEN HI
#DEFINE          MCOUNTCLOCK        PORTB,5           ;INCREMENTS
MEMORY COUNTER ON LOW TRANSITION
#DEFINE          ADGO                 ADCON0,2          ;AD START BIT
#DEFINE          EOMFLAG              FLAG1,0           ;SIGNALS
FULL MEMORY
#DEFINE          INERTIASW            PORTB,0           ;STARTS THE DATA
ACQUISITION
#DEFINE          DATADUMPSW          PORTB,1           ;STARTS THE DATA
SERIAL UPLOAD
#DEFINE          LED1                 PORTB,2           ;
#DEFINE          LED2                 PORTB,3           ;
#DEFINE          SERIN                PORTA,4
#DEFINE          SEROUT               PORTD,7

```

```

;*****CONSTANTS*****
ADTIMEDELAY     EQU          D'240'          ;DELAYAD
TIMEDELAY       EQU          D'255'
MEMDELAY       EQU          D'255'
CLOCKDELAY     EQU          D'255'          ; counter is up count so 255=
2.8uS
                                     ; 0= 206.8 uS each
count = .8uS
ACQDELAY       EQU          D'250'          ;EACH COUNT LESS THAN 255
HERE = 155 uS 249 = 1.086 MS
ANALOGCH0     EQU          B'10000001'
ANALOGCH1     EQU          B'10001001'
ANALOGCH2     EQU          B'10010001'
ANALOGCH3     EQU          B'10011001'
ANALOGCH4     EQU          B'10100001'
ANALOGCH5     EQU          B'10101001'

```

```

    ; Start at the reset vector
org      0x000
    goto Main

org      0x020

```

```

INIT
    call  SER_INIT          ;initialise serial port

    clrf  porta
    clrf  PORTB            ;Clear ports
    clrf  portc
    clrf  portd
    movlw b'00011111'
    MOVWF TRISA
    MOVLW B'00000011'
    MOVWF TRISB            ;PORTB all outputs, DATA PORT FOR MEMORY START AS

OUTPUT
    MOVLW B'11111111'
    MOVWF TRISC            ;PORT C ALL INPUTS(MEMORY DATA BUSS)
    CLRF  TRISD            ;CLEAR ADDRESS PORT
    MOVLW B'00000001' ;PORT E BIT 0 IS ANALOG IN
    MOVWF TRISE
    CLRF  FLAG1
    CLRF  ADDINTLO
    CLRF  ADDINTLO
    ;SET UP MEMORY CONTROL LINES TO INIT CONDITION
    BSF   CHIPENABLE
    BSF   WRITEENABLE
    BSF   OUTPUTENABLE
    CLRF  DELAYLO
    CLRF  DELAYMID
    CLRF  DELAYHI
    CLRF  INTCON2
    BSF   MCOUNTRESET    ;RESETS THE CD4040 COUNTER
    BSF   MCOUNTCLOCK    ; GETS CLOCH READY TO COUNT
    BCF   MCOUNTRESET    ;FINISH RESET READY TO COUNT
    movlw B'10000001' ;Fosc/32, A/D enabled
    movwf ADCON0
    movlw B'00001001' ;left justify,6 analog channels
    movwf ADCON1          ;this places the 8 bit answer into ADRESHI

    movlw B'11000111' ;TMR0 prescaler, 1:256
    movwf T0CON
    CLRF  ADDINTLO        ;CLEAR INTERNAL MEMORY COUNTERS
    CLRF  ADDINTMID
    MOVLW B'00000110'
    MOVWF PORTE
    CLRF  FAKEAD

    RETURN
;*****
;MAIN LOOP OF ACQUISITION
;*****
**
Main

    call  init

WAITSTART  BTSS INERTIASW          ;CHECK THE INERTIA SWITCH HI=NOT MOVING,
LOW = ROCKET HAS BLASTED OFF
            GOTO  MAINLOOP          ;IF INERTIAL SW IS
            BTSS DATADUMPSW

```

```

                GOTO  READDATADUMP
                GOTO  WAITSTART

MAINLOOP
                CALL  ANALOGIN                ;READ THE 6 STRAIN
GUAGES AND STORE RESULTS
                CALL  ANALOGOUT              ;SEND THE DATA TO THE
MEMORY CHIP
                CALL  ADDRESSGENERATOR      ;INCREMENT BASE MEMORY
ADDRESS TO NEXT
                BTFSC EOMFLAG                ;CHECK IF MEMORY
FLAG SHOWS THAT WE HAVE REACHED THE END OF MEMORY
                GOTO  FINISHED              ;FINISHED WITH
ACQUISITIONJUMP TO BIT SCAN ROUTINE AND JUST WAIT
                                                    ;WE NEED TO
SEND THE 4 MS BITS OF THE ADDRESS OUT TO RD0,RD1,RD2,RD3, SO WE MUST
                                                    ;SWAP THE
UPPER AND LOWER NIBBLES OF
                                                    ;THE
ADDINTMID COUNT, AND CLEAR THE PAGE BITS

                CALL  OUTADDHIGH            ;FINISHES THE ADDRESS UPDATE OUTPUT
                CALL  READWAIT              ;DELAY BETWEEN EACH SCAN OF SENSORS

                goto  MAINLOOP              ;Do it again
FINISHED      BTG          LED1
                CALL  READLOOP
                CALL  READLOOP
                CALL  READLOOP
                CALL  READLOOP
                CALL  READLOOP
                GOTO  FINISHED

;*****
;*****
; DELAYAD ROUTINE #1
; counter is up count so 255= 2.8uS
; 0= 206.8 uS each count = .8uS
;*****
;*****
DELAYAD      MOVLW ADTIMEDELAY
                MOVWF DELAYLO

LOWCOUNT
                INCFSZ    DELAYLO
                GOTO  LOWCOUNT
                RETURN

DELAY      MOVLW TIMEDELAY
                MOVWF DELAYLO
                GOTO  LOWCOUNT

DELAYMEM    MOVLW MEMDELAY
                MOVWF DELAYLO
                GOTO  LOWCOUNT

DELAYADDRESSCLOCK MOVLW CLOCKDELAY
                MOVWF DELAYLO

```

```

                                GOTO  LOWCOUNT

;*****
*****
READWAIT          MOVLW  ACQDELAY
                  MOVWF  DELAYMID
READLOOP         CALL   LOWCOUNT
                  INCFSZ        DELAYMID
                  GOTO  READLOOP
                  RETURN
;*****

;*****
*****
; THIS SUBROUTINE GENERATES THE 19 BIT ADDRESS. The first 12 bits of address
(A0-A11)is generated by the cd4040 ripple counter.
;A12,A13,A14,A15 ARE GENERATED IN SOFTWARE COUNTER ADDINTMID
;ADDINTLO+ ADDINTMID HOLDS A COPY OF WHERE THE CD4040 COUNTER SHOULD BE
;ADDINTMID HOLDS THE REST OF THE 12 BIT CD4040 COUNT, AS WELL AS THE LAST 4
BITS OF ACTUAL ADDRESS WHICH WILL BE SENT TO
;PORT C BIT 0,1,2,3.
;*****
*****
ADDRESSGENERATOR
          BCF      MCOUNTCLOCK      ;increments the external counter
          CALL    DELAYADDRESSCLOCK
          BSF      MCOUNTCLOCK      ;FINISH PULSE GET READY FOR NEXT
COUNT

          INCFSZ   ADDINTLO           ; Increment internal address TO MATCH
EXTERNAL
          RETURN                                     ;COUNTER DID NOT ROLL OVER
          INCFSZ   ADDINTMID         ;INCREMENT MID COUNTER
          RETURN
          BSF      EOMFLAG           ;SET END OF MEMORY FLAG

          RETURN

;*****
*****
;READDATA DUMP WILL GENERATE THE REQUIRED ADDRESS TO READ THE DATA BACK.
;IT WILL READ EACH FULL PAGE THEN MOVE TO THE NEXT
;ONCE THE ADDRESS IS VALID, THE DATA WILL BE ON PORT D  NO OTHER CONTROLL
SIGNALS NEED TO BE CHANGED
;SINCE THE OUTPUT WILL REMAIN ACTIVE
;*****
*****
READDATADUMP
          MOVLW  B'11111111'
          MOVWF  TRISC                ;SET UP PORT C AS INPUT
          BCF   CHIPENABLE            ;CHIP IS SELECTED
          BSF   WRITEENABLE           ;KEPT HIGH FOR READ
          BCF   OUTPUTENABLE         ;ENABLES OUTPUT
          CALL  RESETADDRESS
;THE ADDRESS COUNTER SHOULD NOW BE AT ZERO, AND THE INTERNAL COUNTERS ALSO AT
ZERO, AND THE PAGE 0 DATA WILL BE ON PORTD

```

```

READPAGE0  CALL  PAGE0
           MOVF  PORTC,W           ;GET DATA INTO W
           MOVWF BINDALO          ;PUT DATA IN BINDALO
           CALL  BCDCNV           ;CONVERT TO PACKED BCD
           CALL  SENDDATA

           CALL  PAGE1
           MOVF  PORTC,W           ;GET DATA INTO W
           MOVWF BINDALO          ;PUT DATA IN BINDALO
           CALL  BCDCNV           ;CONVERT TO PACKED BCD
           CALL  SENDDATA

           CALL  PAGE2
           MOVF  PORTC,W           ;GET DATA INTO W
           MOVWF BINDALO          ;PUT DATA IN BINDALO
           CALL  BCDCNV           ;CONVERT TO PACKED BCD
           CALL  SENDDATA

           CALL  PAGE3
           MOVF  PORTC,W           ;GET DATA INTO W
           MOVWF BINDALO          ;PUT DATA IN BINDALO
           CALL  BCDCNV           ;CONVERT TO PACKED BCD
           CALL  SENDDATA

           CALL  PAGE4
           MOVF  PORTC,W           ;GET DATA INTO W
           MOVWF BINDALO          ;PUT DATA IN BINDALO
           CALL  BCDCNV           ;CONVERT TO PACKED BCD
           CALL  SENDDATA

           CALL  PAGE5
           MOVF  PORTC,W           ;GET DATA INTO W
           MOVWF BINDALO          ;PUT DATA IN BINDALO
           CALL  BCDCNV           ;CONVERT TO PACKED BCD
           CALL  SENDDATA

           MOVLW D'13'             ;SEND CARRIGE RETURN
           CALL  XMIT_RS232
           CALL  LOWCOUNT
           MOVLW D'10'             ;SEND LINE FEED
           CALL  XMIT_RS232
           CALL  LOWCOUNT
                                           ; THE 6 DATAPOINTS ARE NOW
UPLOADED, SO WE INCREMENT TO THE NEXT MEMORY LOCATION
           CALL  ADDRESSGENERATOR
           CALL  OUTADDHIGH
           INCF  READTSTCOUNT
           BTFSS EOMFLAG
           GOTO  READPAGE0
           CALL  RESETADDRESS

WAITHERE  BTG      LED2
           CALL  READLOOP
           CALL  READLOOP
           CALL  READLOOP
           CALL  READLOOP

```



```

MOVWF ADCON0 ;PUT INTO CONTROL REGISTER
CALL DELAYAD
BSF ADGO ;SET GO BIT TO START CONVERSIONS
ADWAIT BTFSF ADGO ;CHECK GO BIT TO SEE IF FINISHED
GOTO ADWAIT
MOVFF ADRESH,STRAIN1RES ;8 BIT ANSWER INTO STRAIN1RES

MOVLW ANALOGCH1 ;SETS UP ADCON0 FOR SECOND CHANNEL
MOVWF ADCON0 ;PUT INTO CONTROL REGISTER
CALL DELAYAD
BSF ADGO ;SET GO BIT TO START CONVERSIONS
ADWAIT1 BTFSF ADGO ;CHECK GO BIT TO SEE IF FINISHED
GOTO ADWAIT1
MOVFF ADRESH,STRAIN2RES ;8 BIT ANSWER INTO STRAIN2RES

MOVLW ANALOGCH2 ;SETS UP ADCON0 FOR SECOND CHANNEL
MOVWF ADCON0 ;PUT INTO CONTROL REGISTER
CALL DELAYAD
BSF ADGO ;SET GO BIT TO START CONVERSIONS
ADWAIT2 BTFSF ADGO ;CHECK GO BIT TO SEE IF FINISHED
GOTO ADWAIT2
MOVFF ADRESH,STRAIN3RES ;8 BIT ANSWER INTO STRAIN3RES

MOVLW ANALOGCH3 ;SETS UP ADCON0 FOR THIRD CHANNEL
MOVWF ADCON0 ;PUT INTO CONTROL REGISTER
CALL DELAYAD
BSF ADGO ;SET GO BIT TO START CONVERSIONS
ADWAIT3 BTFSF ADGO ;CHECK GO BIT TO SEE IF FINISHED
GOTO ADWAIT3
MOVFF ADRESH,STRAIN4RES ;8 BIT ANSWER INTO STRAIN4RES

MOVLW ANALOGCH4 ;SETS UP ADCON0 FOR THIRD CHANNEL
MOVWF ADCON0 ;PUT INTO CONTROL REGISTER
CALL DELAYAD
BSF ADGO ;SET GO BIT TO START CONVERSIONS
ADWAIT4 BTFSF ADGO ;CHECK GO BIT TO SEE IF FINISHED
GOTO ADWAIT4
MOVFF ADRESH,STRAIN5RES ;8 BIT ANSWER INTO STRAIN4RES

MOVLW ANALOGCH5 ;SETS UP ADCON0 FOR THIRD CHANNEL
MOVWF ADCON0 ;PUT INTO CONTROL REGISTER
CALL DELAYAD
BSF ADGO ;SET GO BIT TO START CONVERSIONS
ADWAIT5 BTFSF ADGO ;CHECK GO BIT TO SEE IF FINISHED
GOTO ADWAIT5
MOVFF ADRESH,STRAIN6RES ;8 BIT ANSWER INTO STRAIN4RES
BTG LED1
RETURN

```

```

;*****
*****
; THIS SUBROUTINE WRITES THE 6 ANALOG CHANELS INTO THEIR RESPECTIVE DATA
MEMORY ADDRESSES
;*****
*****

```



```

ANALOGOUT
    CLR    TRISC ;SET UP DATA PORT TO OUTPUT SO WE CAN SEND DATA TO
THE MEMORY CHIP
    CALL  PAGE0
    MOVFF STRAIN1RES,PORTC
    CALL  WRITE1 ;FIRST WRITE WILL BE TO PAGE 0
WHICH IS ALREADY SET
    MOVFF STRAIN2RES,PORTC
    CALL  PAGE1
    CALL  WRITE1
    MOVFF STRAIN3RES,PORTC
    CALL  PAGE2
    CALL  WRITE1
    MOVFF STRAIN4RES,PORTC
    CALL  PAGE3
    CALL  WRITE1
    MOVFF STRAIN5RES,PORTC
    CALL  PAGE4
    CALL  WRITE1
    MOVFF STRAIN6RES,PORTC
    CALL  PAGE5
    CALL  WRITE1

    RETURN

```

```

;*****
;*****
;PAGE SWAPING ROUTINES
;*****
;*****
PAGE0 BCF          PORTD, 4
      BCF          PORTD, 5
      BCF          PORTD, 6
      RETURN

PAGE1 BSF          PORTD, 4
      BCF          PORTD, 5
      BCF          PORTD, 6
      RETURN

PAGE2 BCF          PORTD, 4
      BSF          PORTD, 5
      BCF          PORTD, 6
      RETURN

PAGE3 BSF          PORTD, 4
      BSF          PORTD, 5
      BCF          PORTD, 6
      RETURN

PAGE4 BCF          PORTD, 4
      BCF          PORTD, 5
      BSF          PORTD, 6
      RETURN

```

```

PAGE5 BSF          PORTD,4
          BCF          PORTD,5
          BSF          PORTD,6
          RETURN

;*****
;*****
; WRITES DATA TO MEMORY CHIP ADDRESS IS ALREADY SETUP
; LOOK AT PAGE 9 AND 10 OF THE DATA SHEET FOR THE WRITE CYCLE TIMING
DIAGRAMS. THE TIMING DIAGRAMS TELL IT ALL
; READ IT OVER A FEW TIMES, AND BE SURE YOU UNDERSTAND ALL THE TERMS USED
; IT LOOKS LIKE THE WRITES CAN BE DONE TWO WAYS, THE CE, AND THE WE. BE
PARTICULARLY CAREFUL OF THE WARNING ABOUT THE DATA LINES BEING UNDEFINED.
;*****
;*****

WRITE1      BCF          WRITEENABLE
          BCF          CHIPENABLE
          NOP
          BSF          WRITEENABLE
          BSF          CHIPENABLE
          RETURN

;*****
;*****
;THE SERIAL ROUTINES WILL BE USED TO SEND COMMANDS TO THE SERIAL DATA DUMP
PART OF THE PROGRAM.
;WE NEED SOME WAY TO TELL THE CHIP TO START SENDING DATA, AND WE DONT WANT IT
TO START UNTILL WE HAVE CONFIRMED COMMUNICATION WITH THE PC, SO WE MAY AS
WELL USE THE SMART LINK TO
;GIVE US SOME UPLOAD OPTIONS. JUST LIKE WE MADE A SERIAL COMMAND FROM THE PC
TRIGGER THE SENDING OF THE MIKE & TROY MESSAGE.
;BECAUSE WE WANT TO GET ALL OF THE SENSOR DATA FOR EACH SENSOR OUT OF MEMORY
IN ONE PIECE, A NEW ADDRESS GENERATOR PROGRAM WILL NEED TO BE WRITTEN
;LOOK AT HOW I DID THE WRITE ADDRESS GENERATOR, TO GET SOME IDEAS AS TO HOW
TO DO THAT.
;*****
;*****

;Serial routines

SER_INIT
          BCF          TRISC,D'6'      ;set B6 as an output
          BSF          TRISC, D'7'     ;set B7 as an input
          BSF          PORTC, D'6'     ;set SER_OUT high
          RETURN

XMIT_RS232  MOVWF      Xmit_Byte      ;move W to Xmit_Byte
          MOVLW      0x08            ;set 8 bits out
          MOVWF      Bit_Cntr
          BCF          SEROUT
          CALL       Bit_Delay
Ser_Loop    RRCF          Xmit_Byte , f      ;send one bit

```



```

                                CLRf  BCD1          ; BCD2 (100'S AND 1000'S)
                                CLRf  BCD2
LOOP16                          RLCf  BINDALO,1
                                RLCf  BINDAHI,1
                                RLCf  BCD1,1
                                RLCf  BCD2,1
                                DECFSZ COUNT,1
                                GOTO  ADJDEC
                                GOTO  UNPACKBCD

ADJDEC                          CALL  ADJBCD1
                                CALL  ADJBCD2
                                GOTO  LOOP16

ADJBCD1                        MOVLW 3
                                ADDWF BCD1,W
                                MOVWF TMP1
                                BTFSC TMP1,3
                                MOVWF BCD1
                                MOVLW 30
                                ADDWF BCD1,W
                                MOVWF TMP1
                                BTFSC TMP1,7
                                MOVWF BCD1
                                RETURN

ADJBCD2                        MOVLW 3
                                ADDWF BCD2,W
                                MOVWF TMP1
                                BTFSC TMP1,3
                                MOVWF BCD2
                                MOVLW 30
                                ADDWF BCD2,W
                                MOVWF TMP1
                                BTFSC TMP1,7
                                MOVWF BCD2
                                RETURN
;*****
;the three BCD digits are packed into BCD1 and BCD2 we want to unpack them
into
;DIGIT100, DIGIT10, DIGITUNITS
;*****

UNPACKBCD                      MOVLW B'00001111'      ;SET MASK
                                ANDWF BCD2,W           ;AND WITH MSD IN BCD2
LSNIBBLE                        MOVWF DIGIT100         ;SAVE UNPACKED BCD 100'S
DIGIT                            MOVFF BCD1,TMP1      ;GET PACKED BCD UNITS AND
TENS IN TEMP1 FOR STRIPPING
                                MOVLW B'00001111'    ;SET MASK
                                ANDWF TMP1,W
                                MOVWF DIGITUNITS
                                SWAPF TMP1           ;SWAP UPPER NIBBLE WITH LOWER
NIBBLE                          MOVLW B'00001111'    ;SET MASK
                                ANDWF TMP1,W
                                MOVWF DIGIT10        ;SAVE TEN'S DIGIT

```

```

;*****
**
;LEADING ZERO SUPPRESSION, ASCII CONVERSION
;*****
ZEROSUPPRESSION          MOVLW D'1'          ;WILL ALWAYS OUTPUT AT
LEAST ONE DIGIT

        MOVWF COUNT
        MOVF  DIGIT100,W
        BTFSS STATUS,Z          ;TEST IF DIGIT 100 IS A ZERO
        GOTO  D100NZ
        MOVF  DIGIT10,W
        BTFSS STATUS,N          ;TEST IF DIGIT 10 IS A ZERO
        GOTO  D10NZ
        MOVF  DIGITUNITS,W
        ADDLW H'30'          ;HEX 30 IS ADDED TO CONVERT
        MOVWF DIGITUNITS
        RETURN

D100NZ          MOVLW D'3'          ;MSD NOT ZERO SO ALL 3 DIGITS WILL
BE SENT

        MOVWF COUNT

        MOVF  DIGIT100,W
        ADDLW H'30'          ;HEX 30 IS ADDED TO CONVERT
        MOVWF DIGIT100

        MOVF  DIGIT10,W
        ADDLW H'30'          ;HEX 30 IS ADDED TO CONVERT
        MOVWF DIGIT10

        MOVF  DIGITUNITS,W
        ADDLW H'30'          ;HEX 30 IS ADDED TO CONVERT
        MOVWF DIGITUNITS
        RETURN

D10NZ          MOVLW D'2'          ;MSD NOT ZERO SO ALL 3 DIGITS WILL BE
SENT

        MOVWF COUNT

        MOVF  DIGIT10,W
        ADDLW H'30'          ;HEX 30 IS ADDED TO CONVERT
        MOVWF DIGIT10

        MOVF  DIGITUNITS,W
        ADDLW H'30'          ;HEX 30 IS ADDED TO CONVERT
        MOVWF DIGITUNITS
        RETURN
;*****
*****
;SEND 3,2, OR 1 DIGIT TO SERIAL PORT
;*****
*****

SENDDATA      MOVLW D'3'
               CPFSEQ      COUNT          ;IS THE COUNT 3?
               GOTO  CHECK42

```

```

MOVF  DIGIT100,W
CALL  XMIT_RS232  ;SEND 100'S
MOVF  DIGIT10,W
CALL  XMIT_RS232  ;SEND 10'S
MOVF  DIGITUNITS,W
CALL  XMIT_RS232  ;SEND UNITS
MOVLW D'44'      ; COMMA
CALL  XMIT_RS232  ;SEND COMMA
RETURN

```

```

CHECK42      MOVLW D'2'
              CPFSEQ      COUNT      ;IS THE COUNT 3?
              GOTO  OUT1
              MOVF  DIGIT10,W
              CALL  XMIT_RS232  ;SEND 10'S
              MOVF  DIGITUNITS,W
              CALL  XMIT_RS232  ;SEND UNITS
              MOVLW D'44'      ; COMMA
              CALL  XMIT_RS232  ;SEND COMMA
              RETURN

```

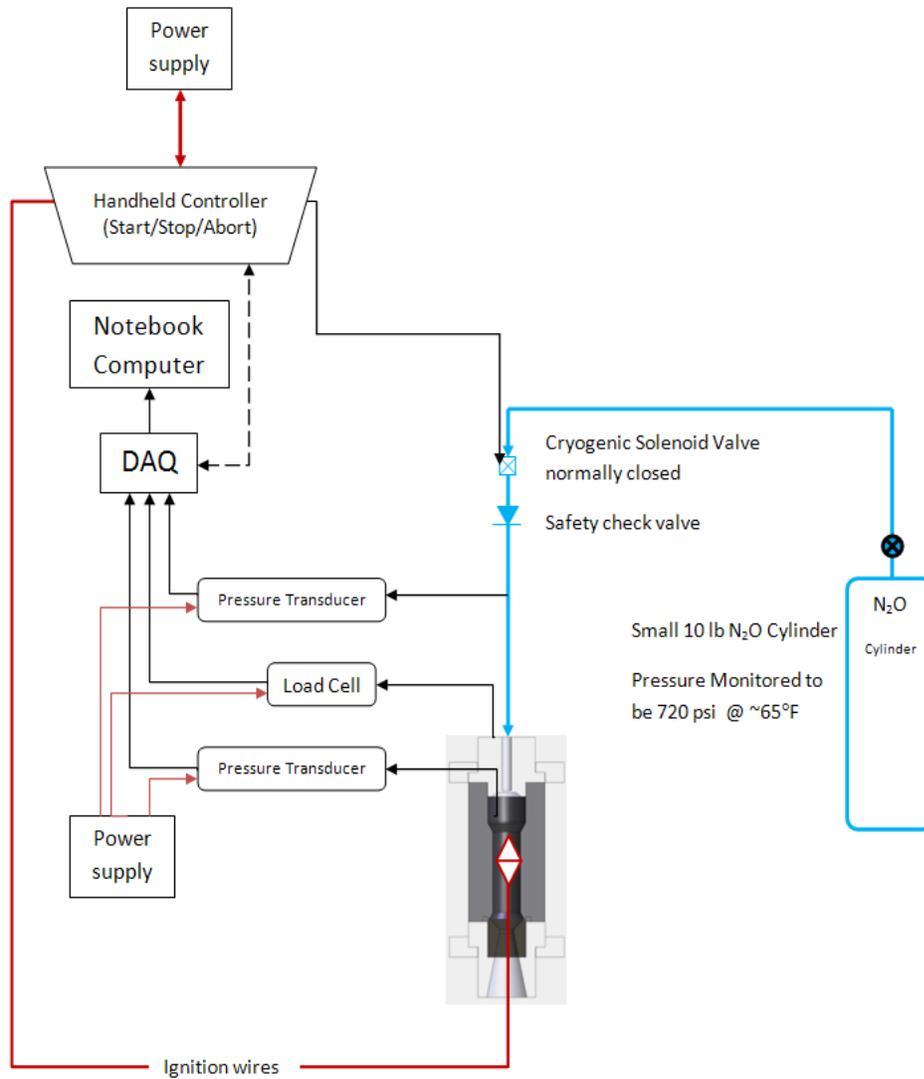
```

OUT1      MOVF  DIGITUNITS,W
           CALL  XMIT_RS232  ;SEND UNITS
           MOVLW D'44'      ; COMMA
           CALL  XMIT_RS232  ;SEND COMMA
           RETURN

```

end

7.7 Appendix G : Nitrous Oxide Flow Diagram



7.8 Appendix H : Manual Controller



Figure 101: Hybrid Motor Manual Controller

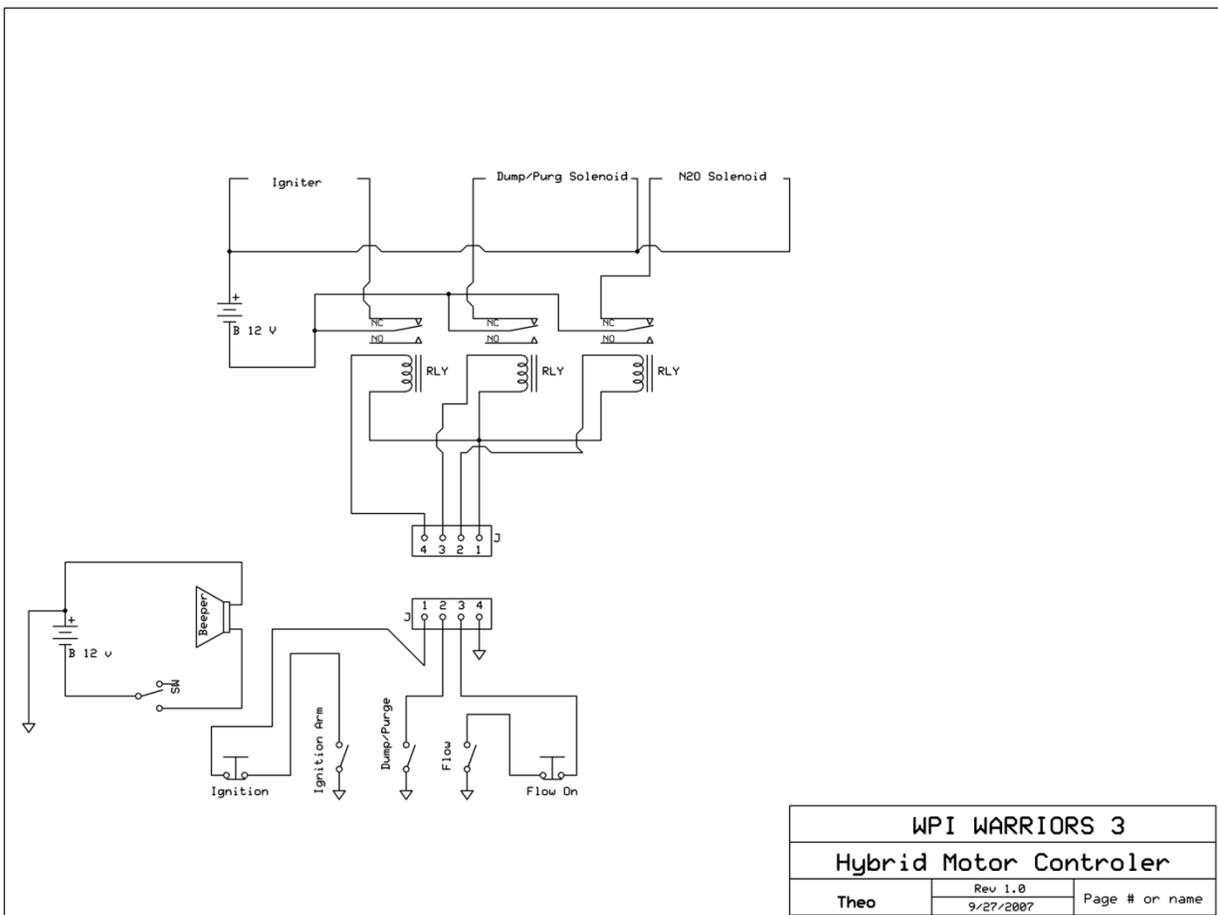


Figure 102: Hybrid Motor Controller wiring diagram