



**Michigan  
Technological  
University**

Michigan Technological University  
**Digital Commons @ Michigan Tech**

---

Dissertations, Master's Theses and Master's Reports

---

2015

## **DEVELOPMENT AND OPERATION OF A MOBILE TEST FACILITY FOR EDUCATION**

Christopher Davis

*Michigan Technological University, [ctdavis@mtu.edu](mailto:ctdavis@mtu.edu)*

Copyright 2015 Christopher Davis

---

### **Recommended Citation**

Davis, Christopher, "DEVELOPMENT AND OPERATION OF A MOBILE TEST FACILITY FOR EDUCATION",  
Open Access Master's Thesis, Michigan Technological University, 2015.  
<https://digitalcommons.mtu.edu/etdr/13>

Follow this and additional works at: <https://digitalcommons.mtu.edu/etdr>



Part of the [Controls and Control Theory Commons](#), [Electro-Mechanical Systems Commons](#), and the  
[Engineering Education Commons](#)

# DEVELOPMENT AND OPERATION OF A MOBILE TEST FACILITY FOR EDUCATION

By

Christopher T. Davis

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Mechanical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2015

© 2015 Christopher T. Davis

This thesis has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Mechanical Engineering

Department of Mechanical Engineering – Engineering Mechanics

Thesis Co-Advisor: *Jeffrey Naber*

Thesis Co-Advisor: *Jeremy Worm*

Committee Member: *Wayne Weaver*

Department Chair: *William W. Predebon*

# Table of Contents

Table of Figures .....	vi
Table of Tables .....	xi
Acronyms.....	xii
Abstract.....	xiii
1 Introduction .....	1
2 Literature Review .....	3
3 Mobile Lab Overview and Outline.....	4
4 Test Facility Hardware.....	7
4.1 Fuel System .....	8
4.2 Ventilation System .....	15
4.3 Dyno and Drive .....	18
4.4 Powertrain Options.....	25
4.4.1 Kohler Engine System .....	25
4.4.2 D&D Motors System .....	31
4.4.3 Yanmar Engine System.....	32
5 Test Facility Data Acquisition and Control .....	34
5.1 Test Cell Control and DAQ System.....	34
5.1.1 PXI Hardware and Supporting Systems .....	34
5.1.1.1 PXI 6514 Module .....	37
5.1.1.2 PXI 6259 Module .....	37
5.1.1.3 PXI 8513/2 Module.....	39
5.1.2 WTI Inerita Software Package.....	40
5.1.2.1 System Explorer and Configuration File .....	41
5.1.2.2 Inertia Workspace.....	47
5.1.2.3 Data Logging.....	48
5.1.2.4 Automated Testing .....	49
5.1.3 NI Veristand with Inertia Plugin Software Package.....	52
5.1.3.1 Veristand Workspace.....	53
5.1.3.2 Project Explorer .....	54



5.1.3.3	System Explorer .....	55
5.1.3.4	CAN Control and Messaging .....	56
5.1.3.5	Data Logging.....	59
5.1.3.6	Automated Testing .....	59
5.2	Hybrid Control System .....	62
5.3	SI Engine Control System .....	63
5.4	CI Engine Control System .....	67
5.5	Combustion Analysis .....	68
5.5.1	NI Combustion Analysis Hardware .....	68
5.5.2	PCB Charge Amplifiers .....	69
5.5.3	$\mu$ DCAT Software Package .....	71
5.5.3.1	Standard Operating Procedure .....	73
5.5.3.2	Data Logging and Post Processing .....	74
6	Course and Experiment Development.....	77
6.1	Electric Machines Experiment .....	77
6.2	Batteries Experiment .....	78
6.3	Thermal Modeling Experiment .....	79
6.4	CAN Communication Experiment.....	83
6.5	Throttle Body Experiment.....	87
6.6	MEEM 5296 Final Project .....	88
6.6.1	AVL Cruise Simulation .....	88
6.6.2	Chassis Dyno Setup .....	88
6.6.3	DynoMax Setup.....	89
6.6.4	Operating the CHEV on the Dyno.....	94
6.7	Electric Machines Seminar Activity.....	95
6.8	Power Electronics Seminar Activity.....	96
6.9	Battery Fundamentals Seminar Activity .....	96
6.10	Battery Applications Seminar Activity .....	98
6.11	Vehicle Dynamics Seminar Activity .....	99
7	Supporting Systems .....	102
7.1	Wireless Router and Network .....	102

7.2	CCTV System.....	104
7.3	Wireless HDMI System .....	104
8	Summary.....	106
9	Future Work.....	107
10	References .....	108
	Appendix A.....	109
	Appendix B.....	111
	Appendix C.....	127

# Table of Figures

Figure 3.1: Picture of the Mobile Lab classroom from the rear expandable door. The rear test cell is off camera to the left of workstation 4.....	4
Figure 3.2: Picture of workstation 1. The power panel can be seen mounted on the wall below the upper monitor.....	5
Figure 3.3: Image of the CHEV. This vehicle is used for several experiments in the Mobile Lab courses.....	6
Figure 3.4: Image of the Hybrid Electric HMMWV. This vehicle has been used for several research projects and course experiments.....	6
Figure 4.1: Picture of front test cell with component callouts.....	7
Figure 4.2: Picture of rear test cell with component callouts.....	8
Figure 4.3: Picture of the Mobile Lab fuel system in the trailer belly box.....	9
Figure 4.4: Low pressure fuel delivery system located next to tanks in trailer belly box. The rear tank system is pictured here.....	10
Figure 4.5: Flow diagram of the belly box fuel system.....	10
Figure 4.6: Fuel control panel in front test cell.....	12
Figure 4.7: Flow diagram of the fuel control panel in the front test cell.....	12
Figure 4.8: Picture of ventilation fan and ducting in front test cell.....	17
Figure 4.9: Picture of ventilation fan and ducting in rear test cell.....	17
Figure 4.10: Image of nameplate for Baldor Reliance motor.....	18
Figure 4.11: An inside view of the dyno drive cabinet.....	20
Figure 4.12: Picture of the pressure taps in the Kohler ECH730 cylinder head from combustion chamber. Pressure taps are circled in red.....	26
Figure 4.13: Picture of pressure taps in Kohler ECH730 cylinder head around the spark plug hole. There are plugs installed in these pressure taps.....	26
Figure 4.14: Picture of pressure tap in Kohler ECH730 cylinder head opposing the spark plug hole. There is a plug installed in this pressure tap.....	27
Figure 4.15: Picture of exhaust system in front test cell. Thermocouple bung locations are circled in green. O2 sensor bung locations are circled in red.....	28
Figure 4.16: Belt drive system to mate the engine to the dyno in the front test cell.....	30
Figure 4.17: Picture of the D&D Motor Systems ES-32B-50 electric machine.....	32
Figure 5.1: Picture of PXI hardware.....	36
Figure 5.2: Picture of the PXI breakout panel used to measure sensor signals.....	38
Figure 5.3: Screenshot of the Inertia launch window. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	41
Figure 5.4: Screenshot of the main window of the Inertia system explorer. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	42

Figure 5.5: Screenshot of Inertia system explorer in the alarm editing menu. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	43
Figure 5.6: Screenshot of the Inertia alarm editing window. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	44
Figure 5.7: Screenshot of the Inertia system explorer window under the procedure branch. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	45
Figure 5.8: Screenshot of the window to add or edit a task of a procedure. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	46
Figure 5.9: Screenshot of the Inertia 6.1 front panel. The static and dynamic window areas are highlighted. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	48
Figure 5.10: Labeled Inertia workspace toolbar buttons. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	48
Figure 5.11: Screenshot of the manual data logging control and editor windows. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	49
Figure 5.12: Image of the Inertia automated test editor utility window. The test shown is to run a speed - load sweep. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	50
Figure 5.13: Screenshot of the popup windows used to create a local variable. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	51
Figure 5.14: Screenshot of the Veristand launch window. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	53
Figure 5.15: Screenshot of Veristand workspace. From here, the test cell can be monitored and controlled. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	54
Figure 5.16: Screenshot of the Veristand Project Explorer window. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	55

Figure 5.17: Screenshot of the Veristand System Explorer window. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	56
Figure 5.18: Screenshot of XNET Database Editor. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	57
Figure 5.19: Screenshot of the XNET Database Editor with the signal properties window open. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	58
Figure 5.20: Screenshot of the System Explorer showing the XNET bus expanded in the system tree. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.	59
Figure 5.21: Screenshot of the Veristand automated test profile editor. This tool is a part of the Inertia plugin. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	60
Figure 5.22: Data log file settings window for the Veristand automated test function. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	61
Figure 5.23: Screenshot of the Inertia automated test execution tool in Veristand. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	62
Figure 5.24: Image of hybrid control unit mounted on breakout panel in front test cell .....	63
Figure 5.25: MotoTron ECU used to control the engine in the front test cell. ....	64
Figure 5.26: MotoTune display and calibration windows for the Kohler engine. ....	66
Figure 5.27: Screenshot of the MotoTune display used to control the Yanmar engine. ....	68
Figure 5.28: Combustion analysis cDAQ unit mounted in the 19" rack in the Mobile Lab test cell.....	69
Figure 5.29: PCB 443B102 charge amplifiers. ....	70
Figure 5.30: Front panel of the PCB 443B102 charge amplifier. ....	70
Figure 5.32: Screenshot of the $\mu$ DCAT workspace. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.....	71
Figure 5.33: Engine setup menu. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	72
Figure 5.34: Synchronous setup menu. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	73

Figure 5.35: Screenshot of the Batch Process Raw Files program. Instructions are in the text box. A prefix of <i>Sum_</i> should be added to help identify the output files as summary files. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	75
Figure 5.36: Screenshot of the Raw File Viewer program. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	76
Figure 6.1: Picture of the communication breakout box for the HE-HMMWV. The DB-9 connector used to communicate over the CAN bus is the top left connector on the box. ....	80
Figure 6.2: Virtual instrument front panel for logging data on the HE-HMMWV. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....	81
Figure 6.3: Inlet of HMMWV air filter with marks showing where to block off with tape. The tape should go on the side of the line with the arrow. There is another line and arrow on the hidden side of the inlet. ....	82
Figure 6.4: Picture of the HMMWV air cleaner indicating the port to connect the manometer to. The port is the one with a piece of tape above by the air cleaner outlet. ....	82
Figure 6.5: Picture of the NI cDAQ mounted in the trunk of the Malibu. The OBD II to serial cable is called out along with the Kvaser Leaflight HS cable. ....	84
Figure 6.6: Default windows for the basic CanKing template. This image contains content under copyright ownership of Kvaser. Use of this material is covered under their CANlib SDK license agreement found in Appendix C. ....	85
Figure 6.7: Message window for CanKing. This image contains content under copyright ownership of Kvaser. Use of this material is covered under their CANlib SDK license agreement found in Appendix C. ....	86
Figure 6.8: Picture of the front of the dyno servo control box. The power and control mode switches are shown. ....	90
Figure 6.9: Picture of the bottom of the servo control box with the servo control cable plugged in. ....	91
Figure 6.10: Screenshot of the new run window with the vehicle tab shown. This image contains material under copyright ownership of Land & Sea. Permission to use this material can be found in Appendix A. ....	92
Figure 6.11: Dyno-Max automated test setup wizard for the clone run test profile. This image contains material under copyright ownership of Land & Sea. Permission to use this material can be found in Appendix A. ....	93
Figure 6.12: Dyno Max chassis cell console window. The current vehicle speed, along with upper and lower speed limits, is shown on the speedo graph and dial. This image contains material under copyright ownership of Land & Sea. Permission to use this material can be found in Appendix A. ....	94

Figure 6.13: Battery fundamental activity VI front panel. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B. ....98

Figure 6.14: Screenshot of the CG Calculator spreadsheet. The results and grey boxes are in metric units of kilograms and meters while the green boxes are in imperial units of pounds and inches. Only the green boxes should be edited.....100

Figure 7.1: Network diagram for the Mobile Lab. The IP of each node is indicated on the diagram. ....103

## Table of Tables

Table 1.1: List of courses and content delivered by the Mobile Lab. The content of each course can be tailored to meet specific needs.....	1
Table 4.1: Bill of materials for the fuel system. ....	14
Table 4.2: Values of each critical parameter for correct operation after a drive reset. A complete list of parameter values can be found in the drive manual. ....	24
Table 4.3: Specifications of Kohler ECH-730 Command Pro EFI engine.....	25
Table 4.4: Specifications of Yanmar L100V engine.....	33
Table 5.1: PXI 1050 slot population specifications.....	37
Table 5.2: Description of the tasks inserted by each command tab of the Inertia Command Editor window. ....	47
Table 5.3: Engine controller pinout and DIN rail connections .....	65
Table 6.1: Message contents of PID request messages for CAN messaging experiment. ....	86
Table 6.2: Connection locations for Bosch ETB pigtail harness on the CHEV. Locations of the ECU pins can be found in the back of the lab manual. ....	87
Table 6.3: Values for mass and coast-down parameter in DynoMax run setup menu for the CHEV.....	92



# Acronyms

$\mu$ DCAT – Micro DRIVEN Combustion Analysis Tool

BOM – Bill of Materials

CAN – Controller Area Network

cDAQ – Compact Data Acquisition

CHEV – Configurable Hybrid Electric Vehicle

CG – Center of Gravity

CVT – Continuously Variable Transmission

ECU – Engine Control Unit

ETB – Electronic Throttle Body

EVC – Exhaust Valve Closing

EVO – Exhaust Valve Opening

HE-HMMWV – Hybrid Electric High Mobility Multi-Wheeled Vehicle

IVC – Intake Valve Closing

IVO – Intake Valve Opening

MAP – Manifold Absolute Pressure

PCM – Powertrain Control Module

RH – Relative Humidity

VI – Virtual Instrument

WTI – Winneman Technologies Incorporated

# Abstract

The automotive industry saw a large shift towards vehicle electrification after the turn of the century. It became necessary to ensure that new and existing engineers were qualified to design and calibrate these new systems. To ensure this training, Michigan Tech received a grant to develop a curriculum based around vehicle electrification. As part of this agenda, the Michigan Tech Mobile Laboratory was developed to provide hands-on training for professional engineers and technicians in hybrid electric vehicles and vehicle electrification. The Mobile Lab has since then increased the scope of the delivered curriculum to include other automotive areas and even customizable course content to meet specific needs.

This thesis outlines the development of the Mobile Laboratory and its powertrain test facilities. The focus of this thesis is to discuss the different hardware and software systems within the lab and test cells. Detailed instructions on the operation and maintenance of each of the systems are discussed. In addition, this thesis outlines the setup and operation of the necessary equipment for several of the experiments for the on and off campus courses and seminars.

# 1 Introduction

The Mobile Lab was developed as part of a DOE grant to train engineers on topics pertaining to hybrid electric vehicles. The goal was to develop a platform that could deliver hands-on training to professional engineers across the country [1,2]. The lab has developed beyond the scope of the grant and continues to offer hands on professional development courses on a wide variety of automotive topics and technologies. In addition to the courses listed in Table 1.1, a custom course can be developed, given enough time, to meet the specific needs of a client.

Table 1.1: List of courses and content delivered by the Mobile Lab. The content of each course can be tailored to meet specific needs.

<b>Short Course Title</b>	<b>Days</b>
Instrumentation & Experimental Methods	2-3
Advanced Powertrain Instrumentation	2-3
Digital Signal Processing and Dynamic Measurements	2-4
Experimental Studies in Hybrid Electric Vehicles	2-5
Modeling and Synthesis of Hybrid Electric Vehicles	2-3
Ground Vehicle Dynamics	2-3
Battery Engineering	2-3
Electric Machines	2-3
Power Electronics	2-3
SI Engine Fundamentals	2-5
SI Engine Management Systems	2-4
Diesel Engine Fundamentals	2-5
Diesel Engine Management Systems, Emissions, and Aftertreatment	2-4
Powertrain Calibration	2-3
Embedded Control System Design	2-3
Model Based Control System Design	2-3
Optimization of Control Systems using MATLAB Simulink	2-3
Nonlinear Control System Analysis and Design	2-3
Smart Microgrid Systems	2-3
Engineering Courses in Thermo, Heat Transfer, Fluids, Energy Systems	1-5

Several parties have been involved in the development of the Mobile Lab and new students and employees are always being added to the team. Due to the complexity of the lab, it was determined that a common user manual was needed for all of its systems. A single Mobile Lab course utilizes several components to deliver content

including the classroom area, test cells, vehicles, software control interfaces, software used to data analysis, and all of the supporting systems. This document will cover the design, operation, and maintenance of these systems so that the Mobile Lab is guaranteed to deliver a quality educational experience to its customers.

## 2 Literature Review

The importance of 'hands-on' engineering education has been realized for some time now. In the late 20<sup>th</sup> century, engineering education in the U.S. took a shift towards theoretical knowledge and training. Due to this shift, the strength of the U.S. industry took a hit because entry level engineers lacked the design experience to develop effective products [3]. As a result of this, a huge push to re-introduce practical and design based education was started.

Undergraduate laboratories and education was completely redesigned to include more hands-on and practical skills to ready graduates for a career in industry. Fundamental objectives for the undergraduate laboratory were developed to address this shift in focus. After completion of a degree, graduates should have an understanding of instrumentation, experimentation, design, and data analysis among other skills [4].

Advancements in technology soon allowed for the offering of off-campus courses through an online interface [5]. This once again raised the issue of how to offer a hands-on education to a student that is not on campus. A study at Purdue University showed that the grades of off-campus students improved significantly by introducing interactive discussions and activities [6].

A previous student, Timothy Okkema, implemented a cart system in the rear test cell [7]. The aim of this system is to standardize the two test cells and make swapping powertrains fast and simple. In addition, a high pressure fuel system was added to the rear test cell to allow for the testing of common rail diesel engines.

### 3 Mobile Lab Overview and Outline

The Mobile Lab is a unique education platform that is designed to deliver hands-on education to professionals. The lab is composed of many elements, the biggest of which being the semi-trailer that was converted into a classroom space with powertrain test cells in the front and back. The classroom area is created by an expandable section of the trailer. As seen in Figure 3.1, there are four workstations inside the classroom. Workstations 1 and 2 are intended to control the front test cell while the other two workstations control the rear test cell.

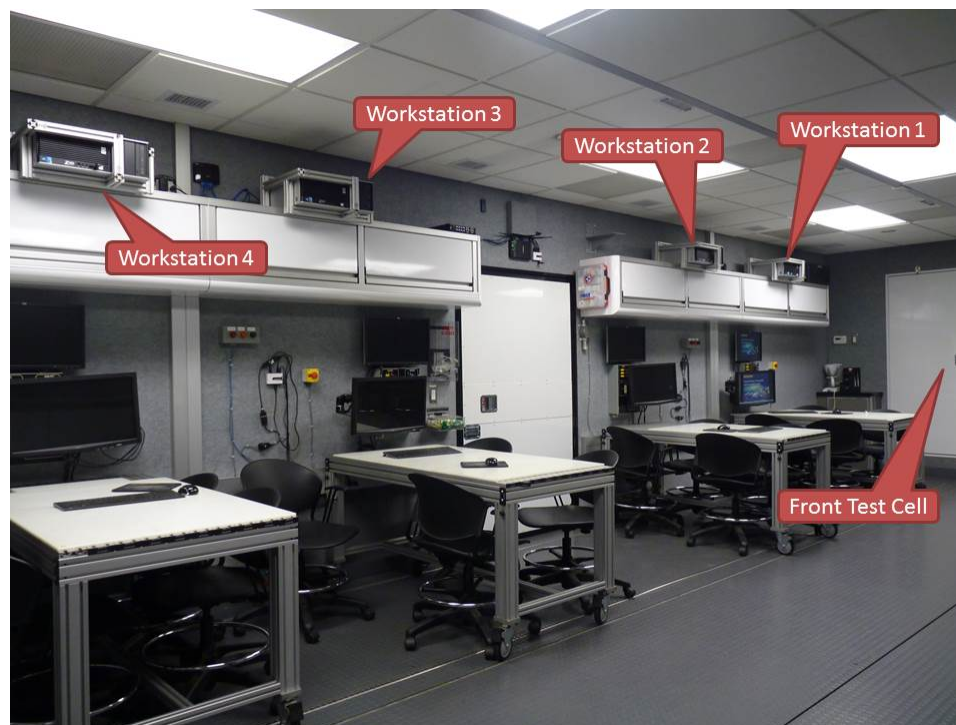


Figure 3.1: Picture of the Mobile Lab classroom from the rear expandable door. The rear test cell is off camera to the left of workstation 4.

Each workstation can comfortably sit four people. The first workstation is pictured in Figure 3.2. The test cells and data acquisition systems can be controlled from the computer. For the front test cell, this control is shared between the front two computers through software lockouts. In addition to the computers, each workstation also has power panel with AC and DC supplies for benchtop experiments. In addition, there is also a quick connect for compressed air. For safety,

an emergency stop switch is located by each workstation that will kill power to both test cells and the fuel system in the belly box.

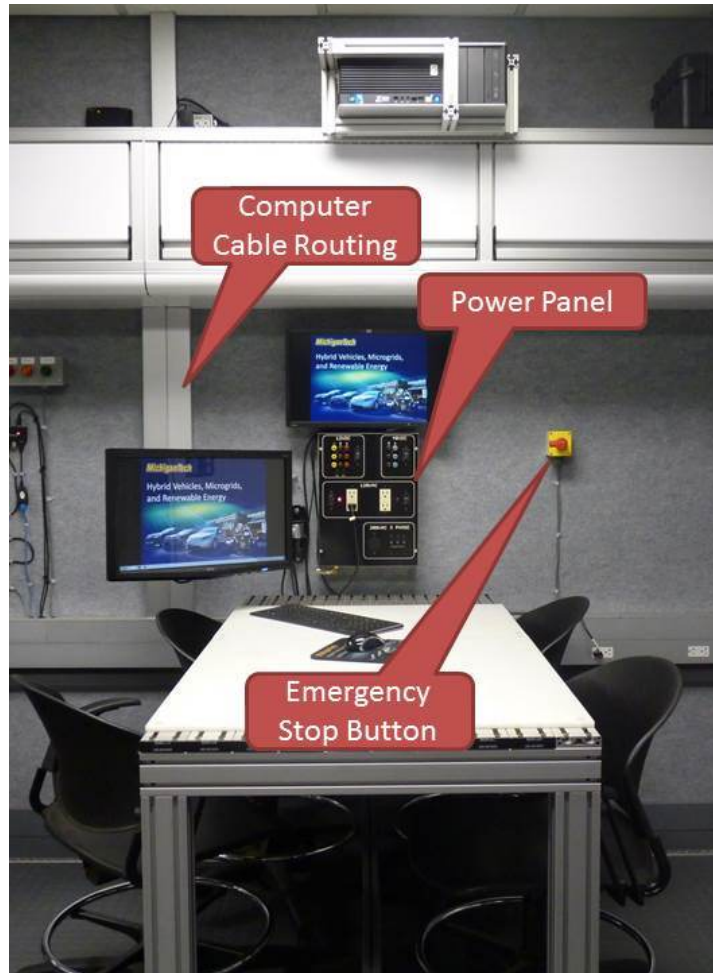


Figure 3.2: Picture of workstation 1. The power panel can be seen mounted on the wall below the upper monitor.

In addition to the mobile classroom, the lab also consists of several production and prototype vehicles. These vehicles range from conventional SI engine vehicles to full electric vehicles. These vehicles are used in some of the experiments for the on campus and short courses. Some of the more notable vehicles are the CHEV and the HE-HMMWV pictured in Figure 3.3 and Figure 3.4 respectively.





Figure 3.3: Image of the CHEV. This vehicle is used for several experiments in the Mobile Lab courses.



Figure 3.4: Image of the Hybrid Electric HMMWV. This vehicle has been used for several research projects and course experiments.



## 4 Test Facility Hardware

The Mobile Lab contains two powertrain test cells that despite being in a mobile platform are very capable. Since the courses being taught out of the lab deal with advanced concepts in the field of automotive engineering and other engineering fields, the test facilities had to be equipped with current equipment and analysis tools. In addition, the facilities had to be built around the idea of continuous improvement so that new technology and equipment could be easily integrated in the future.

Both test cells contain similar hardware. Each cell consists of a high-level control and DAQ system, a dyno drive system, and a combustion analysis system. The front test cell is setup to run low pressure fuel systems while the rear test cell has a high pressure system. Pictures of each test cell with the different systems highlighted can be seen in Figure 4.1 and Figure 4.2.

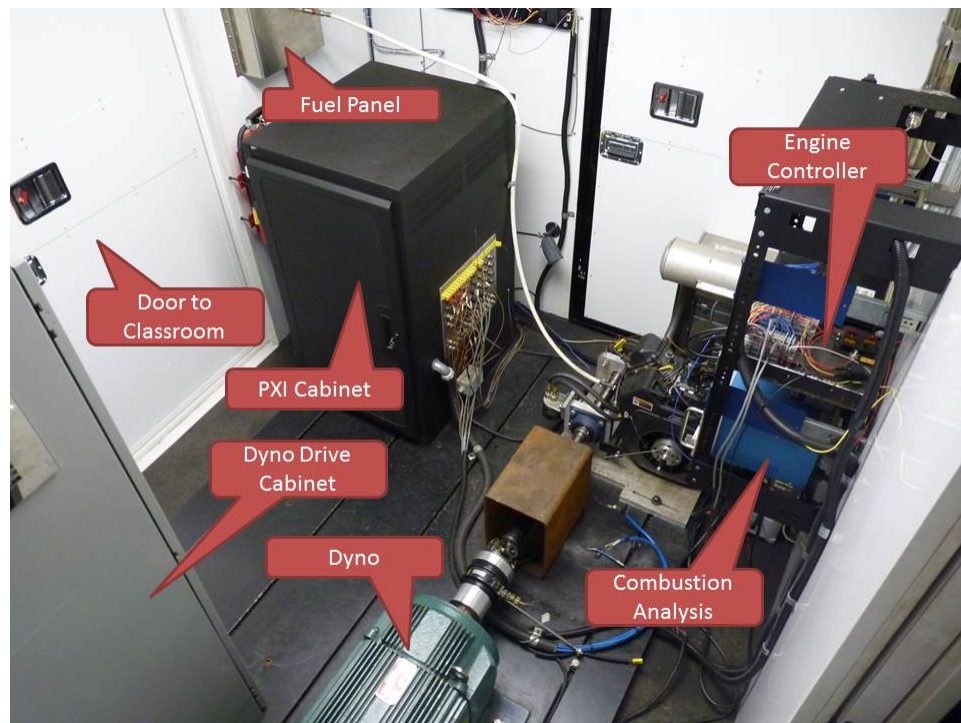


Figure 4.1: Picture of front test cell with component callouts.

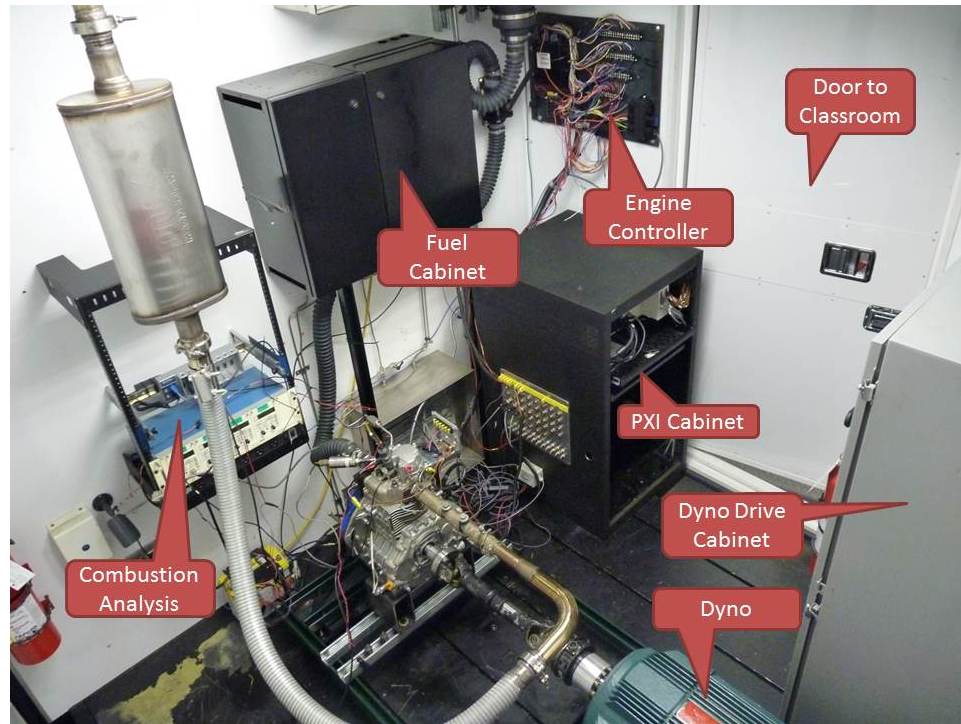


Figure 4.2: Picture of rear test cell with component callouts.

Each of the systems shown in Figure 4.1 and Figure 4.2 are discussed in this section. To supply fuel to the engine under test, a fuel system had to be designed to accommodate a wide range of engines to allow room for different engine platforms in the future. The design and specifications of this system are discussed in section 4.1. A way to ventilate and cool the test cell needed to be developed as well. The ventilation system is discussed in further detail in section 4.2. Mated to the powertrain is a dyno. The dyno and dyno drive hardware specifications are discussed in section 4.3. In section 4.4, the different powertrain options are discussed. Multiple options are needed since the content being delivered out of the lab needs to cover a wide range of topics and technology found in the automotive industry.

## 4.1 Fuel System

The Mobile Lab has a fuel system that is capable of supplying two different fuels to the test cells. There are two 12 gallon fuel tanks located in a rearward belly box of the trailer as seen in Figure 4.3. The tanks are strapped to a shelf on heavy duty

slides so that they can be easily exposed to be filled or removed. Each tank has a filter, pump, and regulator that supply low pressure fuel to the test cells. In addition, there is a solenoid controlled valve that cuts fuel delivery to the cell if it is not using it. These valves are located along the frame rail of the trailer. There are four valves in total, one for the delivery of fuel from each tank to each test cell. Figure 4.4 shows the low pressure fuel delivery system in the belly box. A flow diagram of the belly box fuel system can be seen in Figure 4.5. Every component of the fuel system needs to be capable of handling several different types of liquid fuels; most notably gasoline, ethanol blended fuels, and diesel fuels.



Figure 4.3: Picture of the Mobile Lab fuel system in the trailer belly box.

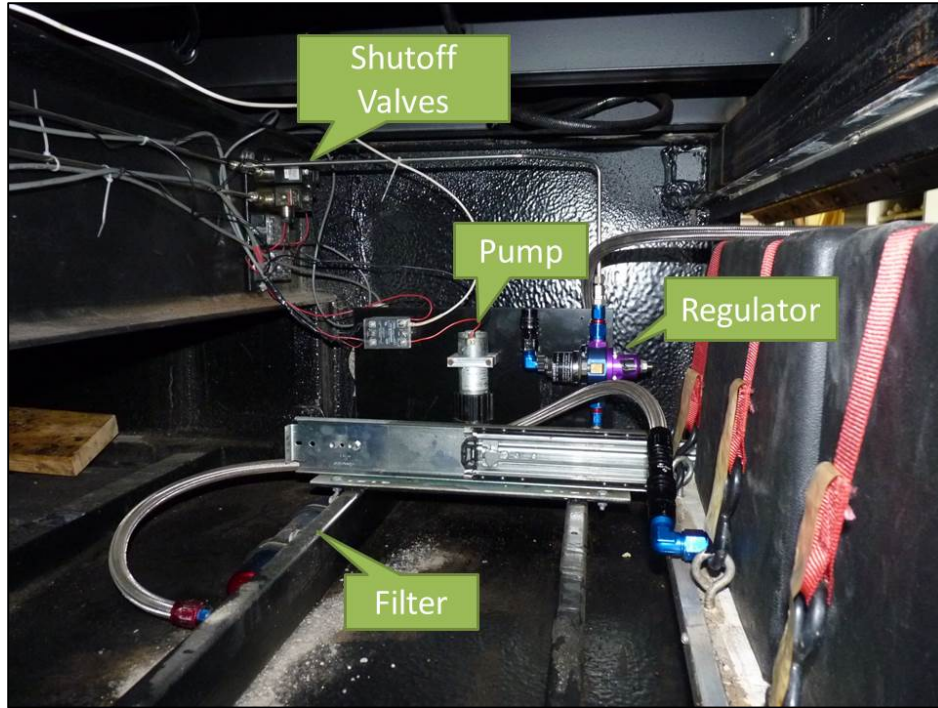


Figure 4.4: Low pressure fuel delivery system located next to tanks in trailer belly box. The rear tank system is pictured here.

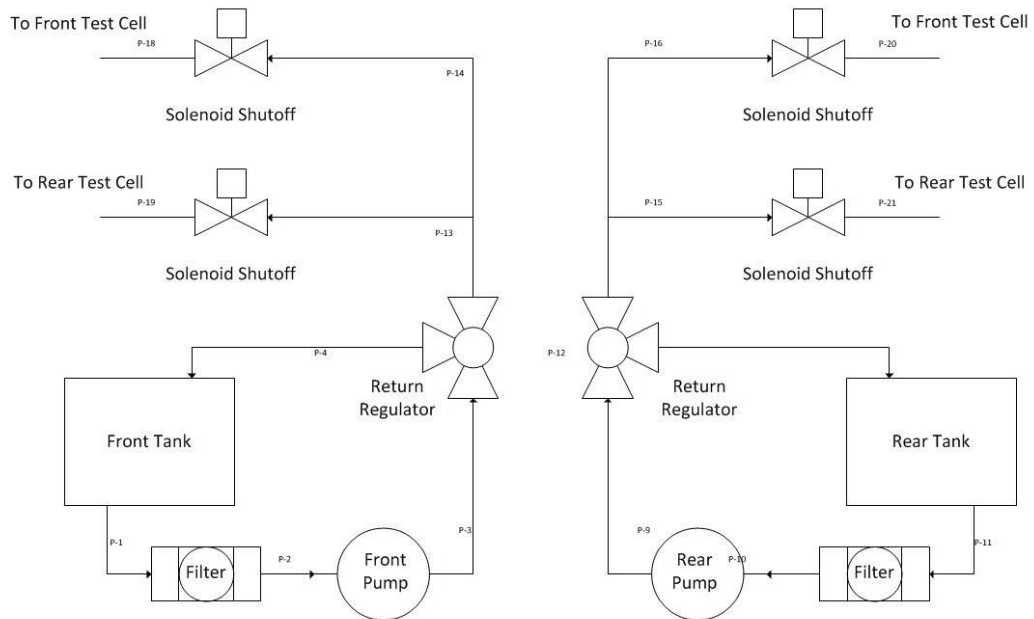


Figure 4.5: Flow diagram of the belly box fuel system.

The pump used to supply low pressure fuel to the test cells is a MG200 series pump from Clark Solutions. This is a stainless steel gear pump capable of a maximum flow rate of 55 L/h at a pressure of 3 bar. This pump provides plenty of room to run higher powered engines in the test cell in the future. Assuming an engine thermal efficiency of 25%, this pump could supply enough fuel to run a 130 kW engine at a 3 bar injection pressure. This is more than double the rated power of the dyno in the test cell. This will also give us plenty of additional flow for less efficient industrial engines.

The fuel control panel for the front test cell is pictured in Figure 4.6. On the panel, there is a manual selection valve to select between the front and rear tanks. The fuel flow can also be shutoff using this valve. This valve only switches what lines are feeding the engine, not what pump is on in the belly box. The pumps and solenoid shutoff valves are controlled through the PXI system. After the tank selection valve, there is a bleed valve. This is used to purge the system of air and to flush the lines between the tanks and the fuel panel if the fuel type is ever changed. Following the bleed valve, there is a return-less regulator with a dial indicator. This is how the pressure is controlled to the engine. An electronically controlled regulator was too expensive so a manual one was used. For the most part, the regulator can be left alone and doesn't need to be changed during a test. A thermocouple, pressure sensor, and flow meter are positioned between the regulator and the supply line to the engine. The pressure sensor is a 0-100 psig transducer from Omega. The flow meter is an FTO-1 turbine flow meter from Flow Technologies with a Linear Link linearizer to condition the signal to the PXI system. This flow meter is capable of measuring a flow rate from 0.2 to 18 L/h. This is the expected flow range of the Kohler engine and is good for engines below 18 kW. If a larger engine is to be run in the test cell, a larger flow meter will need to be installed. A flow diagram of the fuel control panel can be seen in Figure 4.7 below.





Figure 4.6: Fuel control panel in front test cell.

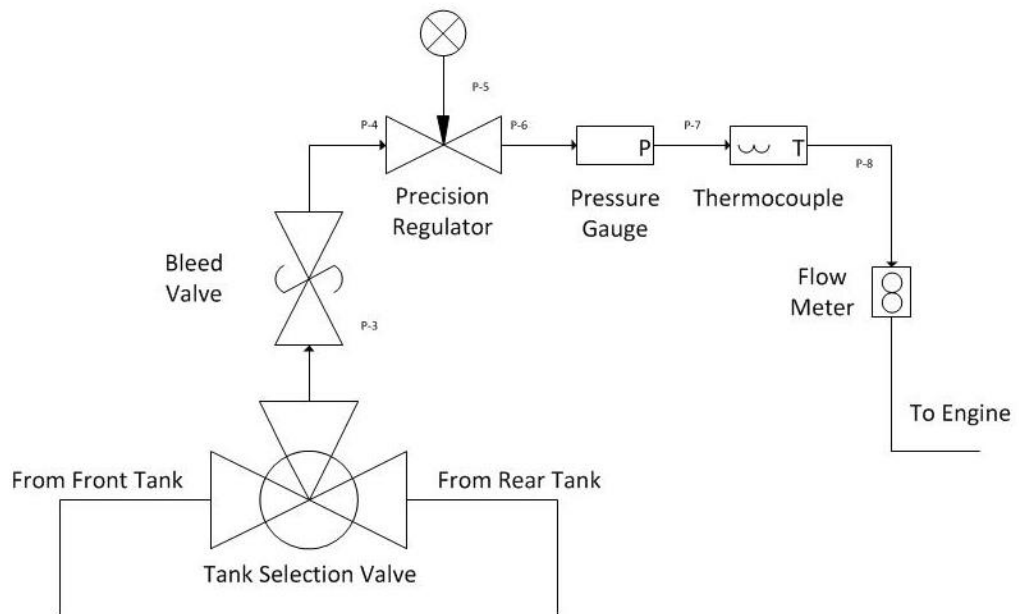


Figure 4.7: Flow diagram of the fuel control panel in the front test cell

The rear test cell contains all of the components of the front test cell. However, rather than delivering fuel to the engine directly, the low pressure system is hard-lined into a high pressure, common rail fuel system. For more details on this system, please refer to Timothy Okkema's thesis [5]. A complete BOM for the fuel system can be found in Table 4.1 on the next page.

Table 4.1: Bill of materials for the fuel system.

<b>BELLYBOX</b>						
<b>Item</b>	<b>Description</b>	<b>Manufacturer</b>	<b>Part Number</b>	<b>Quantity</b>	<b>Vendor</b>	
1	Gear pump	Fluid-O-Tech	MG204XPS17	2	Clark Solutions	
2	12VDC motor for gear pump		MGC4	2	Clark Solutions	
3	Fuel regulator (automotive)	MagnaFuel	MP9925-B	2	Summit Racing	
4	12in, 10 micron fuel filter	Russell	649000	2	Summit Racing	
5	Solenoid shutoff valves	Sitzo Tech Corp	2S025-1/4-1-D-V	4	Sitzo Tech Corp	
6	Fuel tank	Summit Racing	discontinued	2	Summit Racing	
<b>TEST CELL</b>						
<b>Item</b>	<b>Description</b>	<b>Manufacturer</b>	<b>Part Number</b>	<b>Quantity per cell</b>	<b>Vendor</b>	
1	Returnless fuel regulator	Swagelok	KPR1EJC412E600G0	1	Swagelok	
2	Turbine flow meter	Flow Technology	FTO-1NIXSBLHC-5	1	Flow Technologies	
3	Linear Link signal conditioner	Flow Technology	LN-5-C-V1B6	1	Flow Technologies	
4	SS 3-way valve	Swagelok	SS-43GXS4	1	Swagelok	
5	SS bleed valve	Swagelok	SS-BVM4-C3-SH	1	Swagelok	
6	SS hose with Silicone cover	Swagelok	SS-ST4SL4PM4-78	1	Swagelok	
7	Hose to tube adapter	Swagelok	SS-4-HC-A-401	1	Swagelok	
8	SS quick connect female body	Swagelok	SS-QC4-B1-400	1	Swagelok	
9	SS quick connect male body	Swagelok	SS-QC4-D-4PF	1	Swagelok	



## 4.2 Ventilation System

Another support system for the test cell is the ventilation system. The ventilation system serves two purposes. The first is to control the temperature of the test cell. While the engine is under test, heat is being released into the environment which causes the ambient temperature to rise. Even the equipment being on will cause the temperature to increase. A ventilation system was designed to move enough outside air into the test cell to manage the temperature. Calculations were made to determine the amount of airflow needed to maintain a test cell temperature of 50° C. This temperature was set as a target because it is a maximum safe operating temperature specified by the electrical equipment in the test cell. These calculations are outlined below.

If the test cell is considered to be the control boundary, then energy balance on the cell can be described as

$$\dot{Q}_{air} = \dot{m}_{air}(h_{a,out} - h_{a,in}) \quad (4.1)$$

where  $\dot{Q}_{air}$  is the heat-transfer rate to the test cell air and  $\dot{m}_{air}$  is the flow rate of air through the test cell. This equation does not include the energy transferred in and out of the system by the fuel and exhaust because they are lumped into the heat transfer term by

$$\dot{Q}_{air} = (1 - \eta_{gears}\eta_{dyno})P_b + \dot{Q}_{cool} + \dot{Q}_{exhaust} \quad (4.2)$$

where  $P_b$  is the brake power,  $\dot{Q}_{cool}$  is the heat transferred to the test cell air by the engine cooling, and  $\dot{Q}_{exhaust}$  is the heat transferred to the air from the exhaust system.

The values for the terms in Eq. 4.2 can be determined by making a few assumptions and considering the energy conservation equation of the engine:

$$P_b + \dot{Q}_{cool} + (\dot{H}_{e,ic} + \dot{m}h_{e,s}) = \dot{m}_f Q_{LHV} \quad (4.3)$$

where  $\dot{H}_{e,ic}$  is the reference state of the exhaust enthalpy, and  $h_{e,s}$  is the exhaust sensible enthalpy. These terms are roughly evenly split between the fuel heating value term with 25% going to the brake power, roughly 30% going to the cooling fluid, and the remaining 45% going to the exhaust system [8]. It can also be assumed that about 12% of the energy in the exhaust stream is radiated to the surroundings. For the case of calculations, this value was assumed to be 20%. This means that 20% of the exhaust energy is transferred to the test cell air before leaving the trailer which is represented by the variable  $\dot{Q}_{exhaust}$  in Eq. 4.2.

Using the above assumptions and Eq. 4.2 and 4.3, it was determined that the ventilation system needed to reject approximately 32.5 kW of heat while the engine is operating at a full load of 18.6 kW. The required mass flow of air to keep the test cell temperature at 50° C can then be calculated given different ambient air conditions. A couple different scenarios were considered for this. The first condition assumed that the lab was operating in the Midwest during a typical summer day. According to national weather data, the average high temperature seen in Detroit was 28° C with a RH of 70%. In order to maintain the test cell temperature at the desired target in this condition, an air flow rate of 0.247 kg/s or roughly 475 CFM is needed. The other condition considered was on the extreme side with an ambient temperature of 40° C and a RH of 40%. In this condition, the ventilation system would need to flow roughly 1400 CFM of air. To meet these and even more extreme conditions a fan capable of 2000 CFM of air flow was chosen.

In addition to rejecting heat, the ventilation system is used to maintain a negative barometric pressure in the test cell. This is for safety reasons and to keep fumes from entering the classroom area. In the event of a fuel leak, maintaining a pressure below atmospheric ensures that air is being pulled into the test cell through the seals on the door and walls rather than the opposite direction.

The ventilation system consists of an intake louver, fan, ducting and an exhaust louver. Pictures of the front and rear ventilation systems can be seen in Figure 4.8 and Figure 4.9 respectively. For both the front and rear test cells, the intake louver is located near the bottom of the test cell by the engine location. The ventilation fan

is located on the opposite side of the test cell to pull the air across the cell. The ducting then directs the air back across the top of the cell and out through the exhaust louver. The intake and exhaust vents have louvers that open when the ventilation system is powered on. These louvers prevent rain and road debris from enter the trailer while it is on the road.



Figure 4.8: Picture of ventilation fan and ducting in front test cell.



Figure 4.9: Picture of ventilation fan and ducting in rear test cell.

There were a few packaging restrictions that made it difficult to install the ventilation system that was necessary in both the front and rear test cells. The front was the largest challenge due to its reduced size in both square footage and ceiling height. The total height of the front test cell is 90 inches due to the HVAC ducts above and being located over the kingpin of the trailer. As seen in Figure 4.8, the reduced clearance between the door and the ceiling forced the area of the ducting to be reduced to avoid an overhead hazard. In addition there is only a 14 inch gap between the studs in the walls of the trailer. This restricted the size of the exhaust louvers since structural modifications to the trailer would have been too costly. This limited the louver size to 12 x 12 inches, the largest size available without making custom louvers. The rear test cell did not have the height restrictions but did have the same issue with the stud spacing.

### 4.3 Dyno and Drive

In both test cells, the powertrain is mated to a 3-phase 75hp Baldor Reliance RPMAC motor. This line of motors is commonly used as traction motors, test stands and other industrial applications. The rated speed of this motor is 3545 RPM at 460V and 120 Hz. An image of the motor nameplate can be seen in Figure 4.10.



Figure 4.10: Image of nameplate for Baldor Reliance motor

To control the dyno motor, the ABB ACS800 variable frequency drive pictured in Figure 4.11 is used. This drive is capable of controlling motors ranging from 7.5 to 125hp. The drive receives both analog and digital signals from the PXI system to determine operating modes and set points. A 0-10V analog signal is used to determine the set point of the drive. This can be either speed or torque depending on the mode of operation. This analog signal is a bipolar signal. This means that the full negative to positive command is represented by the 0-10V signal. In the case of operating in the speed control mode, 0 speed is commanded by a 5 volt signal, full reverse speed by 0 volts, and full forward speed by 10 volts. The other signals are digital. There are two digital signals, DI1 and DI2, from the PXI system. The first digital channel, DI1 is used to put the drive in run mode. The second signal, DI2 is used to select whether the drive is in speed or torque control mode. When DI2 is false, the drive will be in speed control mode. Specifics on how the PXI system uses these control signals can be found in section 5.1.2. One suggested future modification is to use a mod-bus rather than analog and digital signals to control the dyno. This modification would require significant modifications to the PXI and drive systems but would tighten control of the set points due to the reduction of error and noise in the analog set point signal. A mod-bus adapter has already been purchased and installed in the front cell dyno drive. It is currently not being used though.

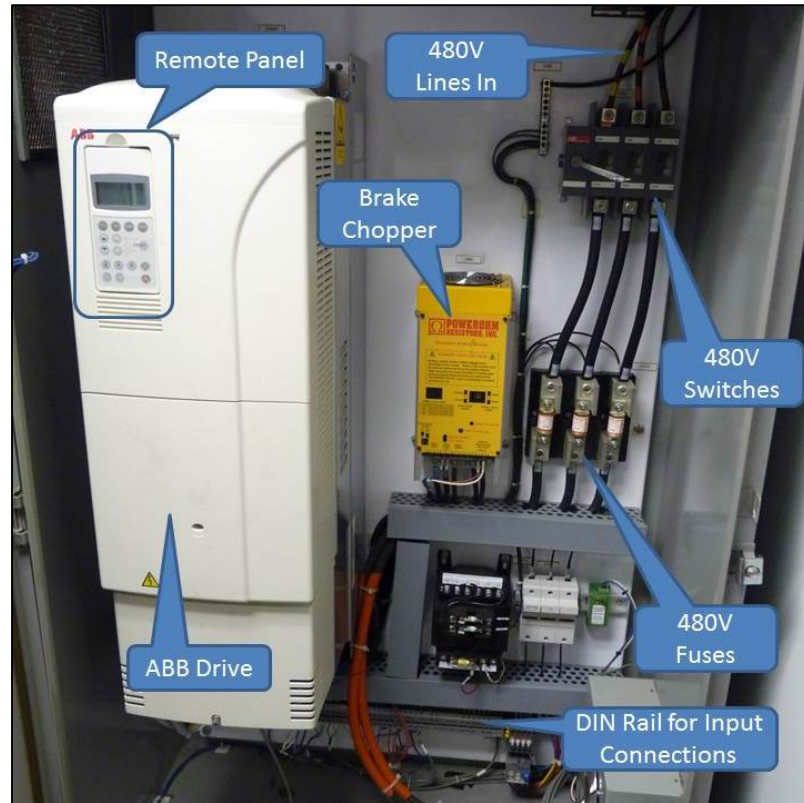


Figure 4.11: An inside view of the dyno drive cabinet.

There are additional safety interlocks used by the drive. The primary one is an emergency stop circuit. The e-stop circuit is tied to the 12V system of the trailer and looped through each test cell. The e-stop circuit is also tied into the drive main power. For the circuit to not be tripped, both front and rear dyno drives need to be on. Each workstation has an e-stop switch. In addition, there is also an e-stop switch located on the drive enclosure. If the e-stop circuit is broken, either by a switch or the drive not being powered on, a popup indicator will show up on the test cell control screen. This is only if the e-stop is sensed by the PXI system of that cell. This means that if the rear drive is not powered up, or a rear workstation switch is tripped, the front cell will shut down but will not provide an on-screen indicator to inform you. There is also a run enable digital input which is tied into the door interlock. If the door is not closed, the drive will not go into run mode.

The correct operation of the dyno is dependent on several firmware settings of the drive. These settings can be accessed using the remote panel on the drive. The drive

cabinet needs to be energized for any firmware settings to be modified. Since the remote panel is located inside the drive cabinet, extreme caution should be used when making changes to any drive parameters. There are extension cables that allow the remote panel to be used away from the drive. It is suggested that this be purchased so that changes to drive parameters can be made without opening the cabinet while it is energized. The cabinet can be opened while it is energized by depressing a small pin in the side of the on-off lever on the front of the enclosure. Drive parameters should not be modified unless the user fully understands the issue being addressed and the effects of the changes. Drive parameters should only need to be modified if hardware changes are made or basic functionality needs to be changed. For a description of each parameter as well as the current values for each parameter, please refer to the ABB firmware manual located in the Mobile Lab file repository. A hard copy is also located with the other user manuals in the Mobile Lab filing cabinet. The drive parameters can be modified by entering the parameter menu using the “PAR” button the remote panel. The parameter groups can then be navigated using the double arrow keys while the parameters within the group can be navigated using the single arrow keys. Pressing the enter button will allow the user to make changes to that parameter.

In the event of a drive reset, there are guided setup menus that can be used to set the correct values for the drive parameters. Details on the guided setup menus can be found in the ACS800 firmware manual starting on page 15. The display of the remote panel will indicate that language selection can be started by pressing the “FUNC” button. The single arrow keys can then be used to select the correct language. After hitting enter, the display will change indicating that the “FUNC” button will now begin guided motor setup. Follow the on screen instructions to setup the motor. All of the necessary information can be found on the motor nameplate.

In addition, the drive can be setup manual by navigating the parameter menu. A few of the more critical parameter settings for correct operation will be discussed here in the event of a drive reset. These critical drive parameters and their values can be seen in





Table 4.22. The first digits in the parameter number are the parameter group that it belongs to. The following digits are the parameter number within that group. Entering the following values into the parameter menu should allow correct operation of the dyno. For a complete list of the correct parameter values, refer to the parameter printout found in the drive manual.

Table 4.2: Values of each critical parameter for correct operation after a drive reset.  
A complete list of parameter values can be found in the drive manual.

Parameter Number	Parameter Name	Parameter Value
10.01	EXT1 STRT/STP/DIR	DI1
10.02	EXT2 STRT/STP/DIR	DI1
10.03	REF DIRECTION	REQUEST
11.02	EXT1/EXT2 SELECT	DI2
11.03	EXT REF1 SELECT	AI1 BIPOLAR
11.04	EXT REF1 MINIMUM [RPM]	0
11.05	EXT REF1 MAXIMUM [RPM]	3454
11.06	EXT REF2 SELECT	AI1 BIPOLAR
11.07	EXT REF2 MINIMUM [%]	0
11.08	EXT REF2 MAXIMUM [%]	100
13.01	MINIMUM AI1	0
13.02	MAXIMUM AI1	10
13.03	SCALE AI1 [%]	100
15.01	ANALOG OUTPUT1	PARAM 15.11
15.06	ANALOG OUTPUT2	PARAM 15.12
15.11	AO1 PTR	0x540E
15.12	A02 PTR	0x5418
20.01	MINIMUM SPEED [RPM]	-3454
20.02	MAXIMUM SPEED [RPM]	3454
21.01	START FUNCTION	CNST DC MAGN
21.03	STOP FUNTION	RAMP
21.07	RUN ENABLE FUNC	RAMP
22.02	ACCEL TIME 1 [S]	5
22.03	DECEL TIME 1 [S]	5
23.01	GAIN	10
23.02	INTEGRATION TIME [S]	2.5
50.01	PULSE NR	1024
50.02	SPEED MEAS MODE	A _ _ _ B _ _ _
50.06	SPEED FB SEL	ENCODER
98.01	ENCODER MODULE	RTAC-SLOT1
99.01	LANGUAGE	ENGLISH AM
99.02	APPLICATION MACRO	T-CTRL
99.04	MOTOR CNTRL MODE	DTC
99.05	MOTOR NOM VOLTAGE [V]	460
99.06	MOTOR NOM CURRENT [A]	95
99.07	MOTOR NOM FREQ [Hz]	60
99.08	MOTOR NOM SPEED [RPM]	1775
99.09	MOTOR NOM POWER [hp]	75

## 4.4 Powertrain Options

There are several powertrain configurations that can be achieved in the test facility. At the moment, there are three propulsion units that are readily available for test within the test cells. The three options available cover the most common propulsion systems found in the automotive market today and are a gasoline engine, a Diesel engine, and an electric motor. These three systems are outlined in sections 4.4.1 through 4.4.3.

### 4.4.1 Kohler Engine System

For an SI engine platform, a Kohler ECH-730 is used. This is a 23 HP, V-Twin industrial engine with port fuel injection. Detailed specifications for the engine can be found below in Table 4.3.

Table 4.3: Specifications of Kohler ECH-730 Command Pro EFI engine

J1995 Power <sup>1</sup>	22.5 HP @ 3600 RPM
Peak Torque <sup>1</sup>	51.7 Nm
Displacement	747 cc
Bore	83 mm
Stroke	69 mm
Conn-rod length	114 mm
Wrist pin offset	0.4 mm
Compression Ratio	9.1:1
I/O	-370 CAD
I/V	-105 CAD
E/O	120 CAD
E/V	400 CAD

<sup>1</sup> These values are gathered from the Kohler engines website [9]. All other values were measured.

Slight modifications were made to the engine for use in the test cell. Pictures of the cylinder head pressure taps can be seen in Figure 4.12 through Figure 4.14. Each head has four pressure taps for a M5 cylinder pressure transducer like the AVL GH15D. Three of the pressure taps are located around the spark plug while the remaining pressure tap is located on the opposite side of the combustion chamber. The reason for having four pressure taps is so that the engine could be used to perform competitive benchmarks on different in cylinder pressure transducers. The

machining for the pressure taps was performed by Calumet Machine in Calumet, Michigan.

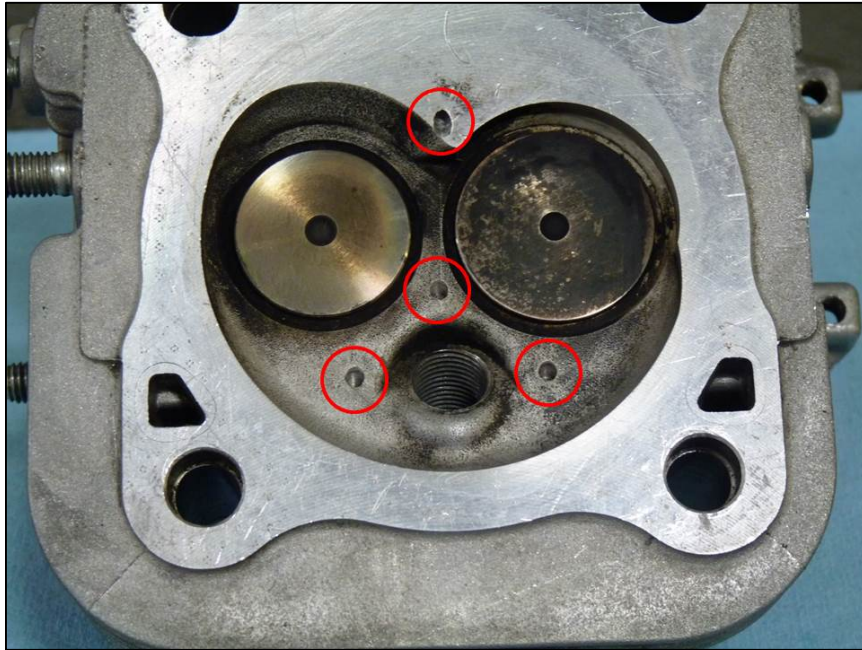


Figure 4.12: Picture of the pressure taps in the Kohler ECH730 cylinder head from combustion chamber. Pressure taps are circled in red.

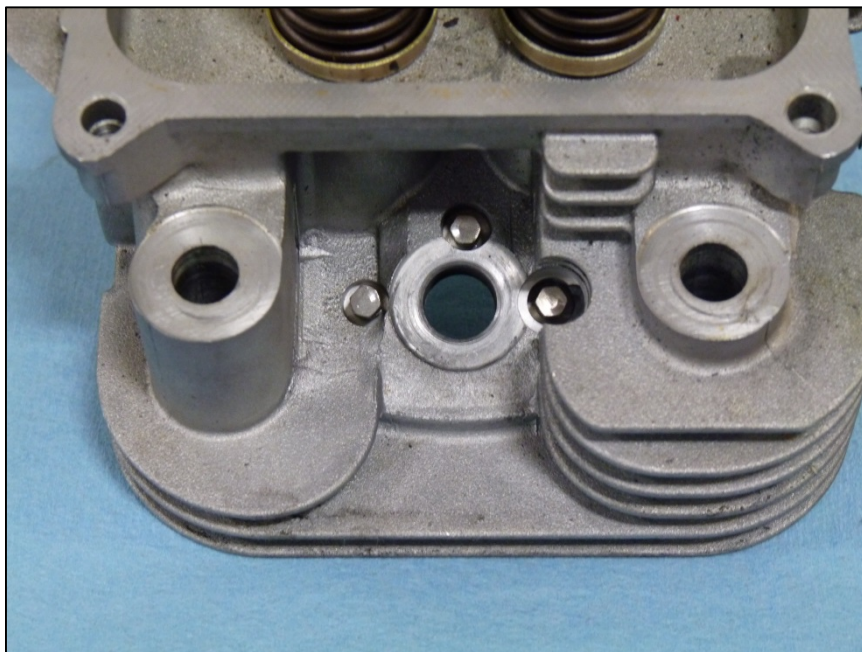


Figure 4.13: Picture of pressure taps in Kohler ECH730 cylinder head around the spark plug hole. There are plugs installed in these pressure taps.

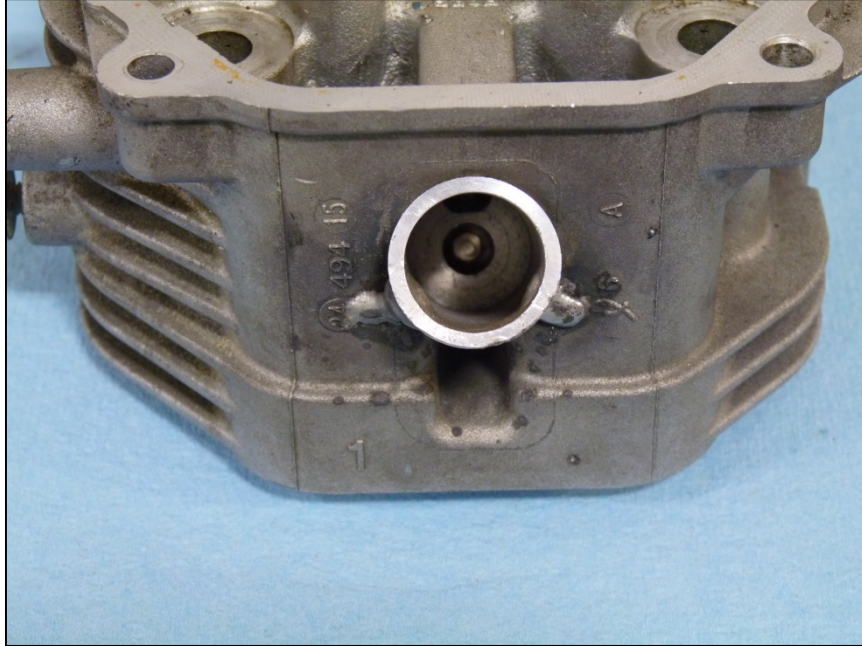


Figure 4.14: Picture of pressure tap in Kohler ECH730 cylinder head opposing the spark plug hole. There is a plug installed in this pressure tap.

The other modifications to the engine mentioned earlier were done to make controlling the engine with a rapid prototyping control solution possible. The control system used in the test cell is a MotoTron<sup>®</sup> 112-pin electronic control module. This system is discussed in further detail in section 5.3. The mechanically actuated throttle body was replaced with a Bosch 32mm electronic throttle body (PN: 0 280 750 148) so that the MotoTron controller system could have control over the engine load. The Bosch 32mm electronic throttle body is oversized for this engine application but was chosen because of its use with other engine programs at MTU and it was readily available. With this throttle body, the engine achieves full load at around 40-50% throttle plate opening. In addition to the air induction system, the ignition system was also modified so that it could be used with the MotoTron<sup>®</sup> controller. The stock coil packs were replaced with GM LSx smart, coil-near-plug packs. These packs were chosen because the MotoTron controllers require external ignition modules and control the ignition through TTL signals.

The exhaust system was also modified so that temperature and gas species data could be collected. First, the exhaust manifold was extended so that there was room

to include thermocouple ports next to the exhaust port. In addition, this positioned the muffler above the engine to allow room for the test cell ventilation system and easy access to the belt drive system for maintenance. Each exhaust runner includes two 1/8" NPT bungs so that thermocouples can be installed to measure the exhaust gas temperature of each port. The stock muffler is used but modified with a Marmon flange on the tailpipe so that it can be mated to the exhaust system in the test cell and routed outside the lab. The section of exhaust above the Marmon flange is a fixture of the test cell. The exhaust systems of additional engines will mate up to this point. There are three M18x1.5 bungs installed in the test cell exhaust system that can be used for wide-band O<sub>2</sub> sensors or as a port for emissions analysis. Although the lab does not have an emissions bench permanently installed in the facility due to space constraints, a portable five-gas emissions bench is available to the lab and has been used on several occasions. A picture of the exhaust system can be seen in Figure 4.15 below with the bung locations circled.



Figure 4.15: Picture of exhaust system in front test cell. Thermocouple bung locations are circled in green. O<sub>2</sub> sensor bung locations are circled in red.

Temperature of the oil, intake air, and the cooling air around the engine is also measured using K-type thermocouples. The oil temperature is measured at a tee fitting in the line feeding the oil filter which was relocated to allow for easy access since it was positioned close to a bearing tower. There are also a couple of pressure measurements for both safety and data analysis. The oil pressure is measured using an Omega 100psig pressure transducer. The pressure pickup is from the stock location for the oil pressure switch in the breather cover. The intake manifold pressure is measured using an Omega 15psia pressure transducer remotely installed in the intake runner for cylinder 2 close to the production MAP sensor.

The engine is currently mated to the dyno via a belt system as seen in Figure 4.16. The pulleys on the input and output can be changed to give different gear ratios between the engine and dyno. Typically, the gear ratio is left at 1:1 for most of the testing that is conducted in the lab. An idler pulley tensions the belt and allows for changes in the size of the drive or driven pulleys if the ratio was ever changed. A synchronous GT-Carbon belt from Gates is used. The belt is 960 mm long and 36 mm wide. It is planned to replace the belt drive system when the front test cell implements a cart system like in the rear. In this new system, the engine will be directly coupled to the dyno.



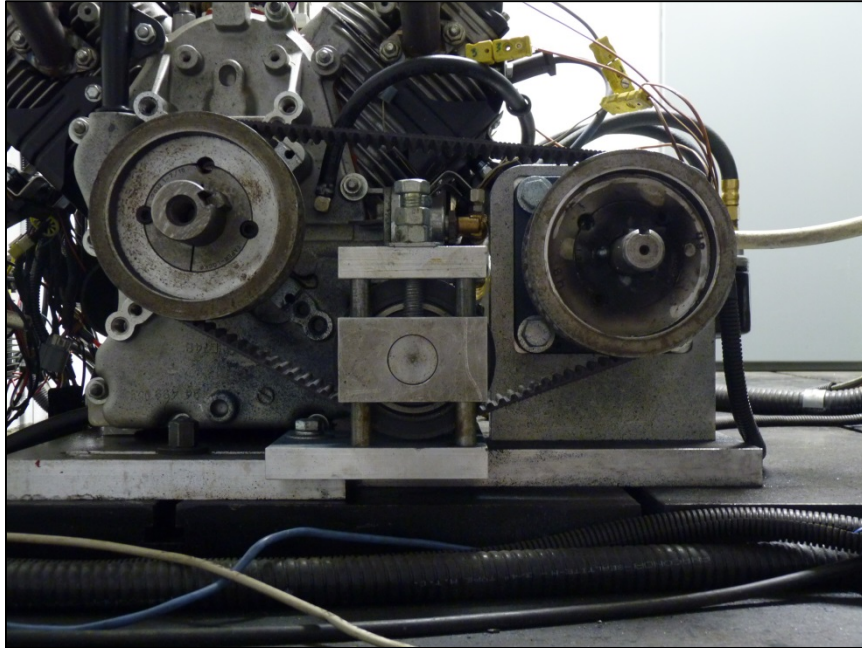


Figure 4.16: Belt drive system to mate the engine to the dyno in the front test cell.

In the event that the belt has to be removed for servicing or engine maintenance, the following procedure should be used remove and then reinstall the belt. The tools needed for this are a 1/4" allen key, 7/32" allen key, 3/4" socket and ratchet, 3/4" wrench, torque wrench capable of reading up to 23 ft-lbs, a steel ruler, and a pencil type force gauge. Follow the procedure below to remove the belt for servicing or replacement.

1. Before servicing the belt drive system, the drive system should be disabled by triggering an e-stop. In addition, the shaft needs to be locked in place. This can be done by inserting a bar through the universal joint of the drive shaft and the T-slot in the bed plate.
2. To release the tension on the belt, loosen the lock nuts and unscrew the tensioning bolts. The idler pulley then can be slid up to remove the tension on the belt.
3. To remove the belt, the pulley on the secondary shaft will need to be removed from the bushing. This is to reduce the risk of damaging the belt when sliding it off of the pulleys. The pulley can be released from the bushing by removing the set screws with a 7/32" allen key.



4. Using one of the removed setscrews, insert it in the hole that is between the two holes that the setscrews were just removed. Tighten the setscrew until the pulley pops off of the bushing.
5. The belt can now be removed completely by using a 1/4" allen key to remove the top braces of the tensioning system and sliding the tensioner up and off of the bracket posts.

To reinstall the belt, follow the above procedure in reverse order. When reinstalling the pulley onto the bushing, the setscrews should be tightened to 23 ft-lbs. Once the pulley is installed on the bushing and the idler bracket is put back together, the belt should be tensioned using the two tensioning bolts. The belt should be tensioned so that there is 3/8" belt deflection in the center of the span when 12 lbs of force is applied to it. This should be done by using the pencil type force gauge and a ruler. It is also a good idea to ensure that both sides of the tensioner are tightened equally. This can be done by measuring the distance between the idler shaft supports and the top cross members of the bracket.

#### **4.4.2 D&D Motors System**

The electric machine platform, shown in Figure 4.17, is a 13 hp, separately excited DC electric machine from D&D Motor Systems (PN ES-32B-50). D&D motors are commonly used in golf carts and electric vehicle conversions. In the test cell, the motor is controlled by a Curtis 1244-5562 motor controller capable of a continuous 500A at 48V. A Curtis 1314 PC programming station is used to communicate with the controller. This device is currently connected to the USB to Ethernet device plugged into the USB switch mounted on the wall between workstations 1 and 2. The location of the software for the controller is unknown at the moment but more information can be found on the 1244 manual located on the portable drive at *~\Manuals\Curtis 1244.pdf*. This manual includes all of the programmable parameters and their functions.

In the CHEV, the Curtis controller was swapped out for a Kelly Controls LLC KDZ separately excited controller with regen capabilities. This controller is capable of supplying 550 amps continuously to the motor. This was done because the Curtis

controller did not allow the regen current to be controlled unlike the Kelly controller. The Kelly controller has an analog brake input that is used to determine the amount of regenerative braking to apply. The Kelly controller can be communicated with via a RS232 port located under the J2 connector on the controller. Any USB to RS232 cable can be used for communication. The manual for the controller can be found on the portable drive at *~\Manuals\KDZ-GDZUserManual.pdf*. The software for communicating with the controller can be found on the portable drive as well at *~\Software Files\Kelly KDC KDZ User Program Release Setup v4.1.zip*.

When the Kelly controller is installed in the test cell, an analog input could be used to simulate a brake pressure sensor. This would require the scaling of the current pedal output from the PXI system since all of the analog outputs are current in use. The best solution would be to make the current accelerator pedal output a bi-polar signal. That would make 10V correspond to 100% positive torque, 0V correspond to 100% negative torque, and 5V correspond to no torque.

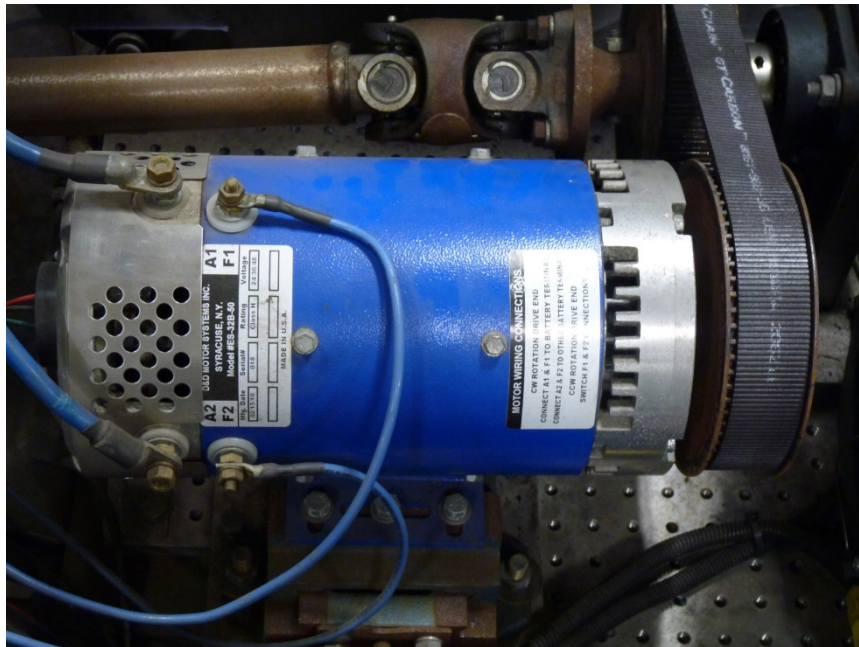


Figure 4.17: Picture of the D&D Motor Systems ES-32B-50 electric machine.

#### 4.4.3 Yanmar Engine System

The CI engine platform is a Yanmar L100V engine. This engine was taken from a Yanmar YDG5500 genset. Specifications for the engine can be seen in

Table 4.4. The engine genset was originally a test platform used by Nostrum Energy. All of the instrumentation for combustion analysis and testing was done by Nostrum.

Table 4.4: Specifications of Yanmar L100V engine

Net Continuous Power <sup>1</sup>	8.3 HP @ 3600 RPM
Displacement	435 cc
Bore	86 mm
Stroke	75 mm
Conn Rod Length	120.7 mm
Wrist Pin Offset	0.7 mm
Compression Ratio	18.3:1
IVO	-355 CAD
IVC	-140 CAD
EVO	155 CAD
EVC	355 CAD

<sup>1</sup> These values are gathered from the Yanmar engines website [10]. All other values were measured.

## 5 Test Facility Data Acquisition and Control

The Mobile Lab test cells include several control and DAQ systems that work together to execute tests and collect the necessary data. Test cell control and primary data acquisition is handled by a NI PXI system. This system and its function are discussed further in section 5.1. In the front test cell, there are two rapid prototyping controllers. The first is a powertrain controller and is discussed further in section 5.2. The other rapid prototyping controller is used to control the engine and further details can be found in section 5.3. In the rear test cell, only one rapid prototyping controller is used. This controller's purpose is to control the CI engine and it is discussed in section 5.4. Both test cells have a built in combustion analysis system. The systems are the same between the front and rear and a discussion on the hardware and software can be found in section 5.5.

### 5.1 Test Cell Control and DAQ System

High level test cell control and the primary data acquisition are handled by an NI PXI system running WTI Inertia in the front and NI Veristand with the Inertia plugin in the rear. This system is the main interface for operating the test cell. The PXI controllers have many functions which include safety, test execution, and data acquisition.

The most important function of the PXI controller is to handle all safety alarm systems and execute appropriate shutdown procedures in the event of an alarm.

The same hardware is used in both test cells and is discussed in further detail in section 5.1.1. At the moment, the front and rear test cells have different software packages but have the same procedures and capabilities. The Inertia software used in the front test cell is discussed in section 5.1.2 while the Veristand software package with the Inertia plugin used in the rear is discussed in section 5.1.3.

#### 5.1.1 PXI Hardware and Supporting Systems

The PXI hardware used to control the test cell can be seen in Figure 5.1. A PXI-1050 chassis is used in both the front and rear test cells. This chassis has 8 3U PXI slots with 4 integrated SCXI module slots. The SCXI modules are used for signal

conditioning and are internally cabled to the controller. The module configuration is detailed in

Table 5.1 with further discussion of each module below the table. Wiring diagrams for the PXI system can be found on the portable drive in the *~\Test Cell\Electrical Drawings* directory. There is no difference in the PXI hardware or the wiring between the front and rear test cells.

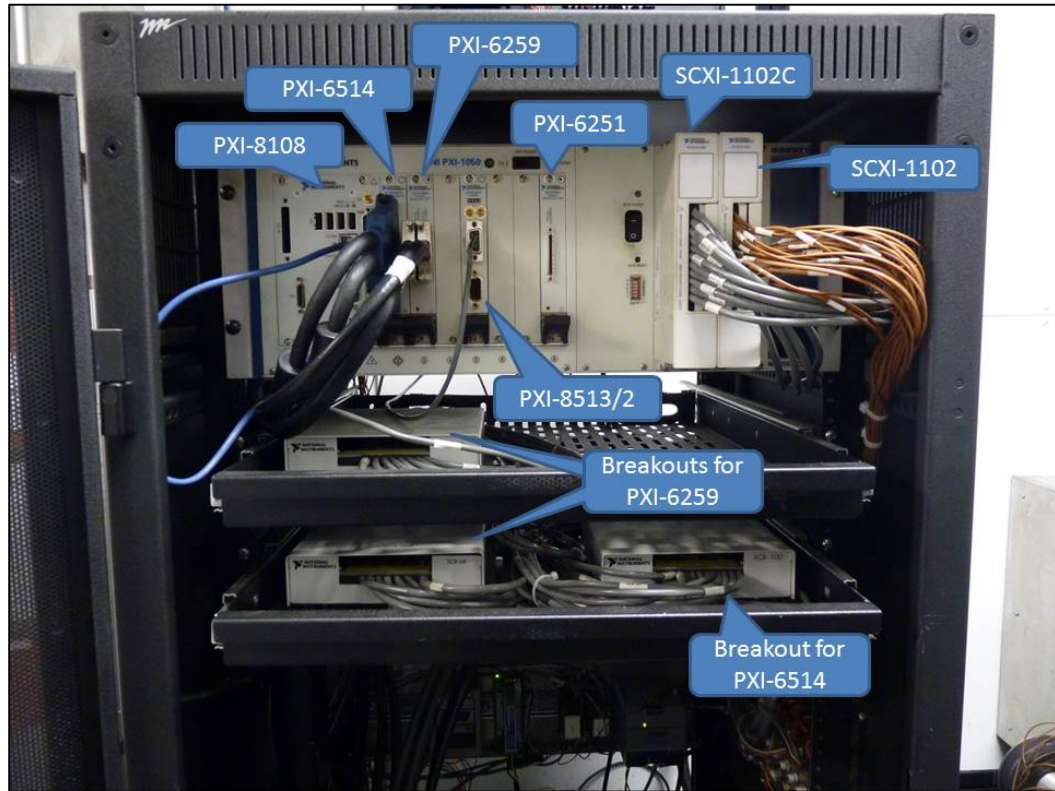


Figure 5.1: Picture of PXI hardware

Table 5.1: PXI 1050 slot population specifications

Slot	Module Number	Description
PXI – 1	PXI – 8108	2.53 GHz Dual-Core PXI Embedded Controller
PXI – 2	PXI – 6514	Industrial Digital I/O (32 inputs, 32 outputs) Module
PXI – 3	PXI – 6259	16-bit, 1 MS/s, 32-Channel Analog Input Module
PXI – 4	Empty	
PXI – 5	PXI – 8513/2	2-Port NI-XNET CAN Interface
PXI – 6	Empty	
PXI – 7	Empty	
PXI – 8	PXI – 6251	16-bit, 1 MS/s, 16-Channel Analog Input Module
SCXI – 1	SCXI – 1102C	32-Channel Amplifier Module (10 kHz Bandwidth)
SCXI – 2	SCXI – 1102	32-Channel Thermocouple Input Module
SCXI – 3	Empty	
SCXI – 4	Empty	

#### 5.1.1.1 PXI 6514 Module

The PXI-6514 digital I/O module is used for background test cell control. This module has 8 digital ports, each with 8 signals for a total of 64 digital signals. Each port is capable of sourcing 75 mA over each channel or 350 mA if only one channel is connected to the port. The module is also configured so that 24 V is a logic high condition on the inputs and outputs. Ports 0 through 3 are inputs while ports 4 through 7 are outputs. Port 0 is used to sense that the emergency stop circuit has been tripped. Port 3 is connected to the door switch relay and senses if the door has been opened. Port 4 is used for the digital control signals to the drive, DI0 and DI1. Port 5 is used to turn on the ventilation system, the fuel pump relays, the load bank cooling relay, and the indicator lamps for the test cell status. Port 6 acts like the key switch for the MotoTron controllers while Port 7 is used to set the powertrain mode and direction of operation.

#### 5.1.1.2 PXI 6259 Module

The PXI-6259 module is a multi-functional module with 16 differential analog inputs, 4 analog outputs, 24 differential digital I/O lines, and 2 counter/timers. This

module is used for the majority of the sensor acquisition and is broken out to the panel on the side of the PXI enclosure seen in Figure 5.2. The 2 counter/timer channels are not used even though they appear on the breakout panel.

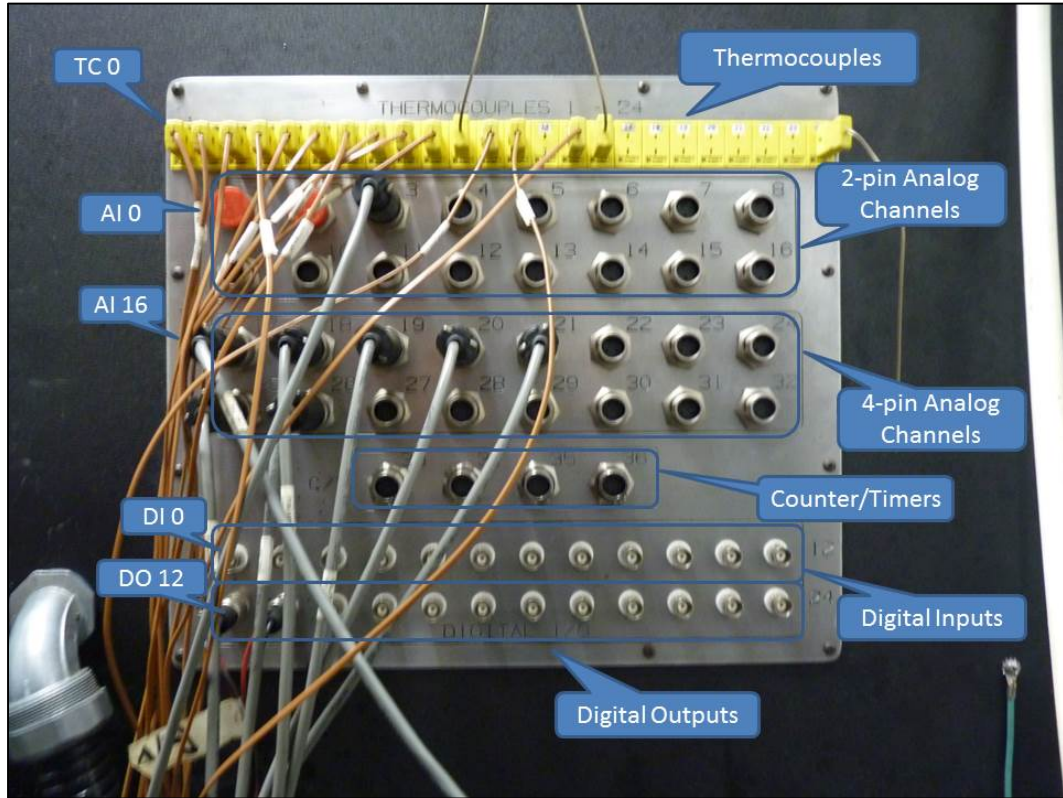


Figure 5.2: Picture of the PXI breakout panel used to measure sensor signals.

The 16 analog inputs are split into two different connector types on the breakout panel. Analog channels AI 0 through AI 7 are wired to 2-pin connectors for sensor that do not require external excitation. These correspond to the connectors labeled AI 1 through AI 8 in the top cluster of connectors. Analog channels AI 16 through AI 23 are wired to 4-pin connectors and are meant for sensors that require external excitation. These correspond to the connectors labeled AI 17 through AI 24 in the middle cluster of connectors. The excitation voltage is 24V and should be only used with sensors that can handle that voltage. If a sensor is connected that needs a lower excitation voltage, a two pin analog channel should be used with the excitation voltage coming from another source. At the moment, there is no standard for connecting sensors the breakout panel for the front test cell. When the system is



switch over to the cart system, the standard mentioned in Timothy Okkema's thesis should be adopted [5].

Even though there are connectors for all of the analog input channels on the PXI-6259, not all of them are wired to the module. The exceptions are AI 0, AI 1, AI 7 and AI 23. These channels are used for measuring the 48 V battery pack voltage (AI 0) and current (AI 1) as well as the speed (AI 7) and torque (AI 23) feedback from the ABB dyno drive. It is suggested that channels AI 0 and AI 1 be re-wired to the breakout panel rather than be directly wired to measure the 48 V battery pack. This is because the pack is not always present in the cell, freeing up two analog channels for other purposes. This can simply be done by replacing the wire connected to the top DIN rail in the back of the PXI cabinet on the terminal indicated in the wiring diagrams. Each pin is already pre-wired, they just need to be swapped.

The digital I/O connections on the breakout panel are connected to the digital port of the PXI-6259 module. Digital channels DI0 through DI11 are configured as inputs and correspond to connectors 1 through 12. The remaining 12 digital channels, DO12 through DO23, are configured as outputs at correspond to connectors labeled 13 through 24. In both the front and rear test cells, channels DO12 and DO13 are used to control the front and rear tank solenoid valves respectively. These channels should never be disconnected or the fuel system will not function properly.

The analog output channels of the PXI-6259 are wired to the lower terminal strip of the PXI enclosure. AO0 is used as the throttle or load command to the powertrain controller. AO1 is used as the regen command to the powertrain controller. AO2 is used to set the ventilation fan speed. AO3 is used as the command reference to the dyno drive.

#### *5.1.1.3 PXI 8513/2 Module*

The PXI-8513/2 is a two port CAN module used to communicate over the CAN bus to the powertrain controller. Currently there are issues with sending CAN messages using Inertia 6.1. Sending CAN messages using Veristand has not been tested yet. On this module, channel 0 is used to communicate with the powertrain controller.

Data is received based on a protocol developed in house. The format of the messages is very similar to J1939. This channel is connected to the CAN2 bus of the powertrain controllers. The second CAN channel can be used to communicate to the dyno drive using MOD bus. Currently this channel remains unused.

### **5.1.2 WTI Inerita Software Package**

The front test cell PXI system is currently running WTI Inertia 6.1. It is planned to update this package to NI Veristand with the Inertia plugin currently being used in the rear test cell. Both of the systems are very similar except for a few differences.

Once Inertia is launched, a launch window will pop up as shown in Figure 5.3. From here, the configuration menu can be accessed, the system explorer can be launched, and the front panel can be synchronized with the real-time target to operate the test cell. Clicking the *Setup button* will open the configuration menu where the IP address and the configuration file being executed on the target can be changed. Unless a major crash has occurred, this menu should be left alone. The system explorer is where the configuration file can be edited. This should be done when procedures, alarms, or DAQ hardware configurations need to be modified. More details on how to use the system explorer can be found below. The run button will open up the front panel and synchronize with the real-time target to allow for operation of the test cell. The components of the front panel and how to operate the test cell will be discussed later in this section.

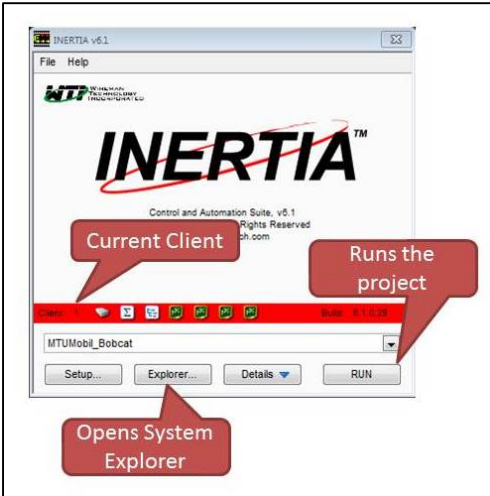


Figure 5.3: Screenshot of the Inertia launch window. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

#### 5.1.2.1 System Explorer and Configuration File

The real-time configuration file can be edited through the system explorer by clicking the 'Explorer' button on the launch window. The explorer will display a prompt for login credentials. The username is 'a' and there is no password. The system explorer will then be opened as shown in Figure 5.4. From here, the PXI system can be configured. Changes to the configuration of the DAQ cards can be made by expanding the list in the navigation tree for that card. This will allow the user to change the names of signals, configure whether they are inputs or outputs, and other features depending on the type of card. It is not good practice to name the channels in the configuration file if they are wired through the breakout panel. The only channels that should be given a detailed name here are the channels that are hardwired. Calculated variables can be edited by expanding the 'Calculated Variables' item in the system tree. Calculated variables can be used to apply filters, running averages, and math operations on channels and between channels. The 'CAN Messages' item in the tree is where the CAN messages being received and transmitted by the system are configured. The 'User Variables' item in the tree is where user defined variables can be created for use in procedures and alarms.

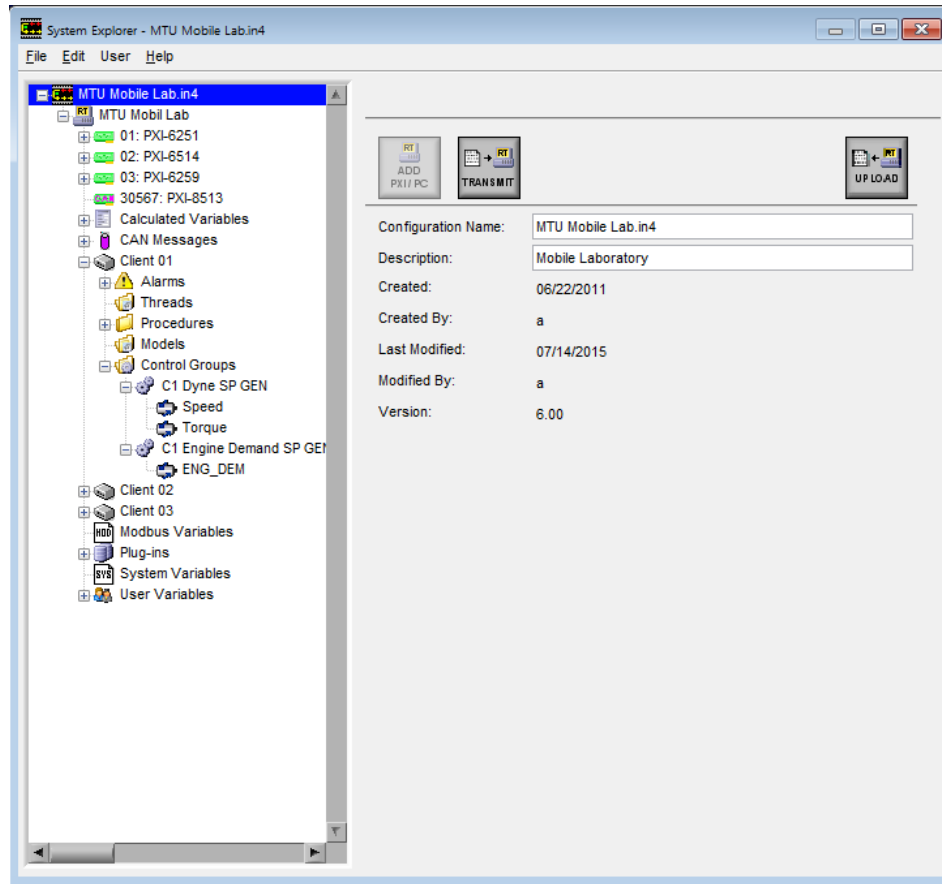


Figure 5.4: Screenshot of the main window of the Inertia system explorer. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

Alarms can be added and edited by expanding the alarm item on the system explorer tree under the desired client. Doing so will display all of the current alarms for that client. To add an alarm, highlight the 'Alarms' item on the system tree and click the button with the plus sign on the alarm symbol. To view details of an existing alarm, click on an alarm in the system tree. The system explorer window will then look like the Screenshot in Figure 5.5. Clicking on the edit button will pull up a new window, shown in Figure 5.6, where settings of the alarm can be modified. This is the same window that will pop up when creating a new alarm. The upper and lower limits are typically going to be user defined variables. These user defined variables will have values assigned to them in a procedure. This will allow for the alarms to be adaptive to different operating modes and conditions. The procedure for handling a tripped

alarm is defined by the alarm action drop down box. This procedure will need to be defined for that client in the Procedure branch of the explorer tree.

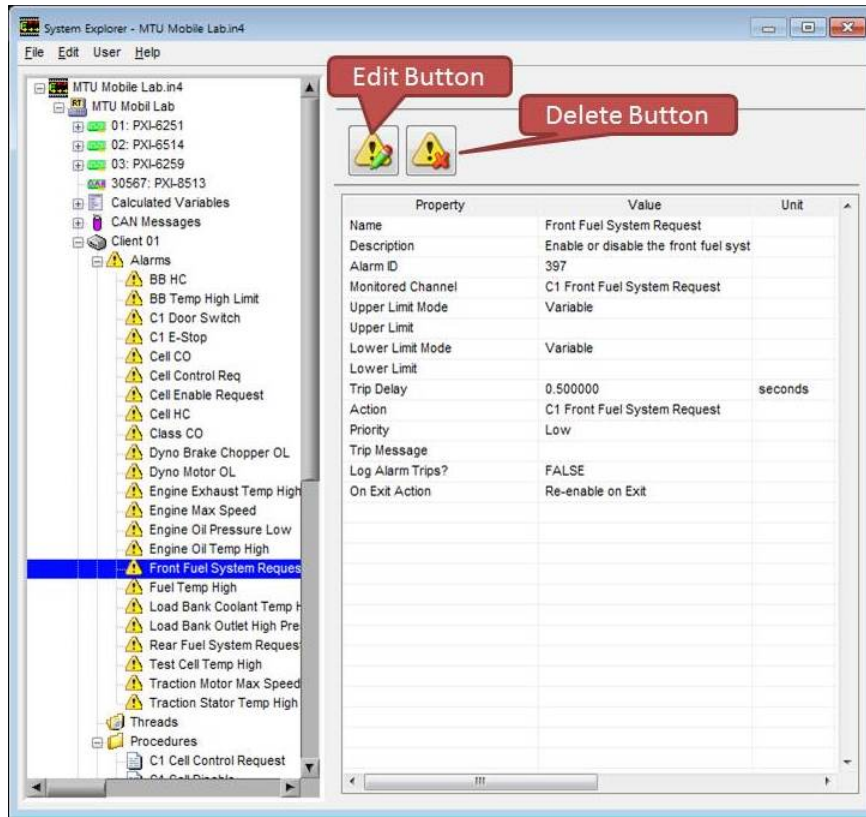


Figure 5.5: Screenshot of Inertia system explorer in the alarm editing menu. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

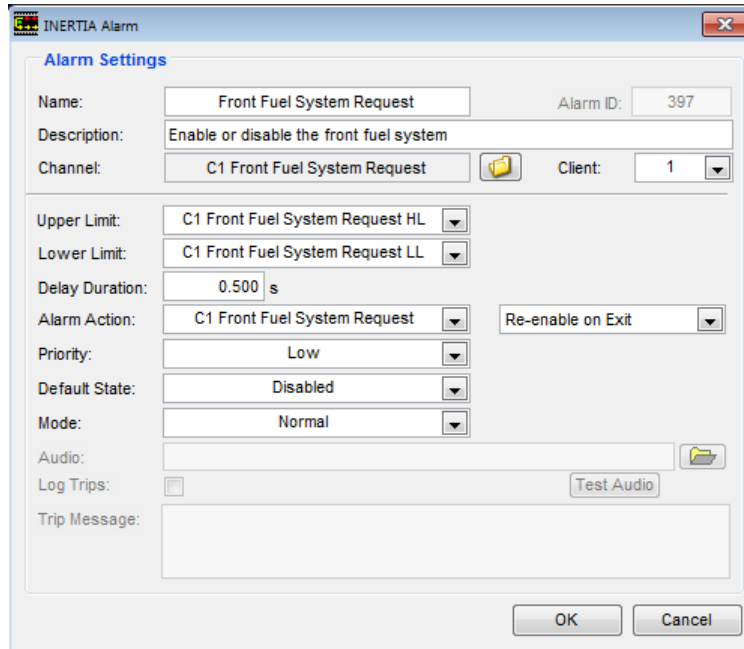


Figure 5.6: Screenshot of the Inertia alarm editing window. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

Operating procedures are defined in the procedures branch of the system explorer tree. Expanding the list will display all of the current procedures defined for that client. The procedure labeled 'Main' is the default procedure that is run on startup of the client. From this procedure, other procedures are called to initialize the user defined variables for the alarms. The other procedures on the list not called by the main loop are alarm actions and act as software interrupts when the alarm is triggered. For example, the 'C1 Cell Control Request' procedure shown in Figure 5.7 is an alarm action procedure that is triggered when the 'C1 Cell Control Request' Boolean is outside of the limits defined in the initialization procedure. Depending on whether the request is above or below the limit window determines the actions taken as shown in the task list of the procedure.

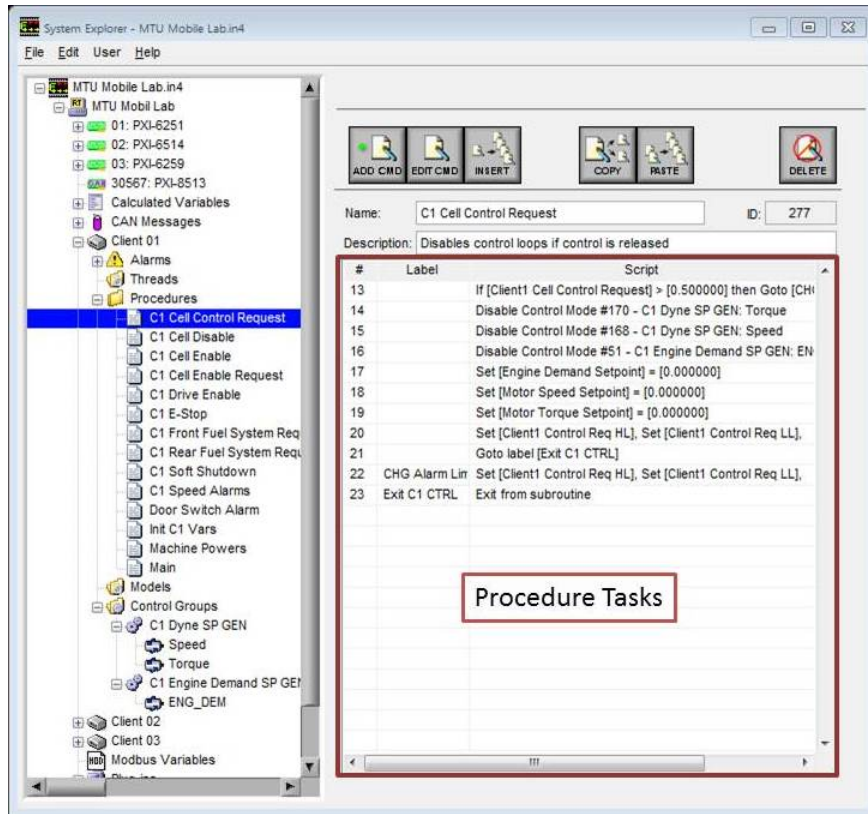


Figure 5.7: Screenshot of the Inertia system explorer window under the procedure branch. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

Procedures can be modified using the toolbar buttons to add or duplicate commands. Clicking the 'ADD CMD' button will pull up the window shown in Figure 5.8. This is also the same window that will pop up when a task is edited by double clicking on the task in the list. There are several tabs shown at the top of the window that will insert different commands. For example, the 'Set Variable' tab shown in the figure will insert a command line that sets a selected variable to a value. This value can be either another variable or a constant by using the drop down box to the right of the 'Value #1' form box. In addition, math can be performed on the variable using the 'Function' drop down box. A brief description of each command tab can be found in

This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

Table 5.2. Additional details can be found in the help manual for Inertia in the ‘Help’ menu on the System Explorer window.

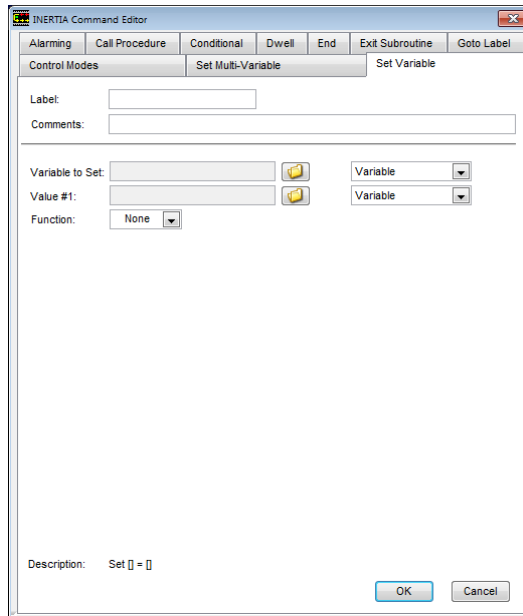


Figure 5.8: Screenshot of the window to add or edit a task of a procedure. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.



Table 5.2: Description of the tasks inserted by each command tab of the Inertia Command Editor window.

Command Tab	Description
Set Variable	Task to set the value of variable to another variable or constant. Math can also be applied to the variable using the 'Function' drop down box.
Set Multi-Variable	Task to set a single or multiple variables to other variables or constants. Math cannot be applied using this task.
Control Modes	Task used to switch between control modes of a control group.
Goto Label	Task used to skip other tasks in the procedure. This task will jump to the task with the label specified.
Exit Subroutine	Task used to exit the current procedure and return to the parent procedure.
End	???
Dwell	Task used to wait a specified amount of time.
Conditional	Task used to enter an if statement. Goto labels are used internally here to jump to the correct section of code.
Call Procedure	Task used to call a subroutine.
Alarming	Task used to handle alarms. This should be used in alarm action procedures to procedurally reset the alarm after it has been addressed.

#### 5.1.2.2 Inertia Workspace

The Inertia workspace can be seen in Figure 5.9. There are multiple screens that can be cycled through using the arrows on the toolbar as shown in Figure 5.10. The current screen being displayed is indicated in the display next to the screen selection arrows. There is a portion of the workspace window that remains static between each screen. This is indicated by the box labeled static content in the figure. Information on this portion of the screen contains all of the control items and general feedback information such as dyno speed and torque. The portion of the window that is labeled dynamic content is unique to each screen. In the figure, the engine screen is selected and the dynamic portion of the window is displaying information relevant to testing an engine in the cell.



Figure 5.9: Screenshot of the Inertia 6.1 front panel. The static and dynamic window areas are highlighted. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

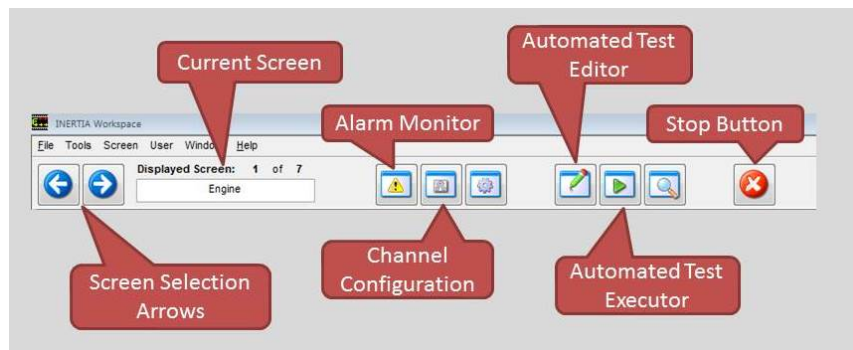


Figure 5.10: Labeled Inertia workspace toolbar buttons. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

### 5.1.2.3 Data Logging

Data can be recorded outside of an automated test by using the 'Manual Data Logger' found in *Tools > Manual Data Logging Example*. A data logging control window will be opened. From here, a data logging file set can be opened using the open button or created using the edit button. Logging files sets have been created for

running experiments on the engine, electric machine, and battery pack. These file sets are located in 'C:\Data\Inertia\' on the workstation computer. Clicking on the edit button will open a window to edit the logging file set. This window will configure the logging file and what channels are to be recorded. Triggers can be setup as well to automatically start logging data in the event of an alarm or when a selected variable has exceeded a specified limit. Both the logging control window and editor window can be seen in Figure 5.11 below. To manually log data, even with a trigger, the file has to be run using the green arrow on the control window. Data logging can be paused by hitting the pause button that will replace the run arrow, or stopped by hitting the stop button on the control window toolbar.

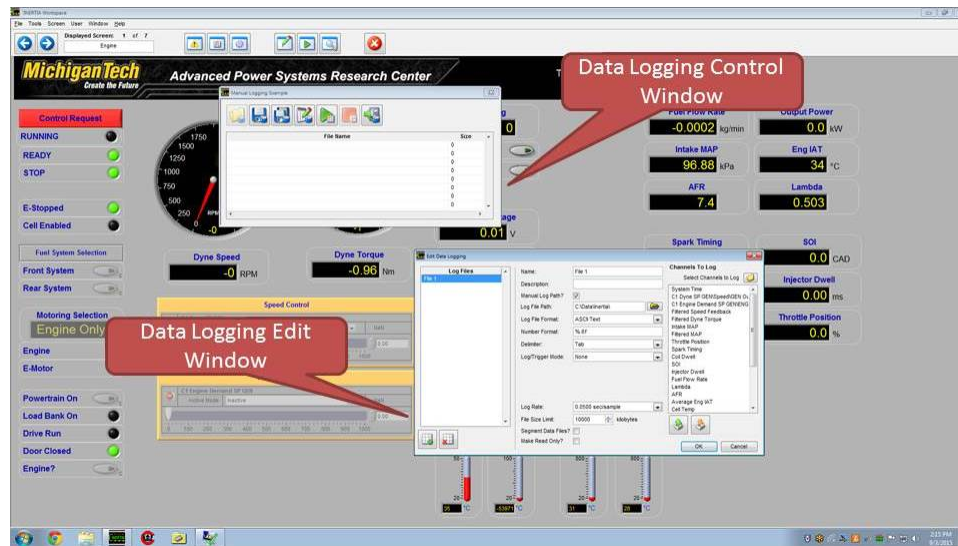


Figure 5.11: Screenshot of the manual data logging control and editor windows. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

#### 5.1.2.4 Automated Testing

There are a series of buttons on the toolbar for automated testing. The automated test editor button will open up a window to create or edit automated test scripts. General test scripts have already been developed for mapping the engine as well as some quick 4-point demonstration tests for the engine and electric motor. Test scripts are located in 'C:\Program Files (x86)\WTI\Inertia 6.1\Libraries and Support Files\Test Setup\''. The automated test editor window can be seen in Figure 5.12 below. The tools used to write a script for an automated test is very similar to

the ones used to write a procedure in the system explorer. In the left most sub-window are the test commands. The two sub-windows to the right are the script windows for each control group. Commands can be dragged and dropped from the test command window and dropped in the control group windows.

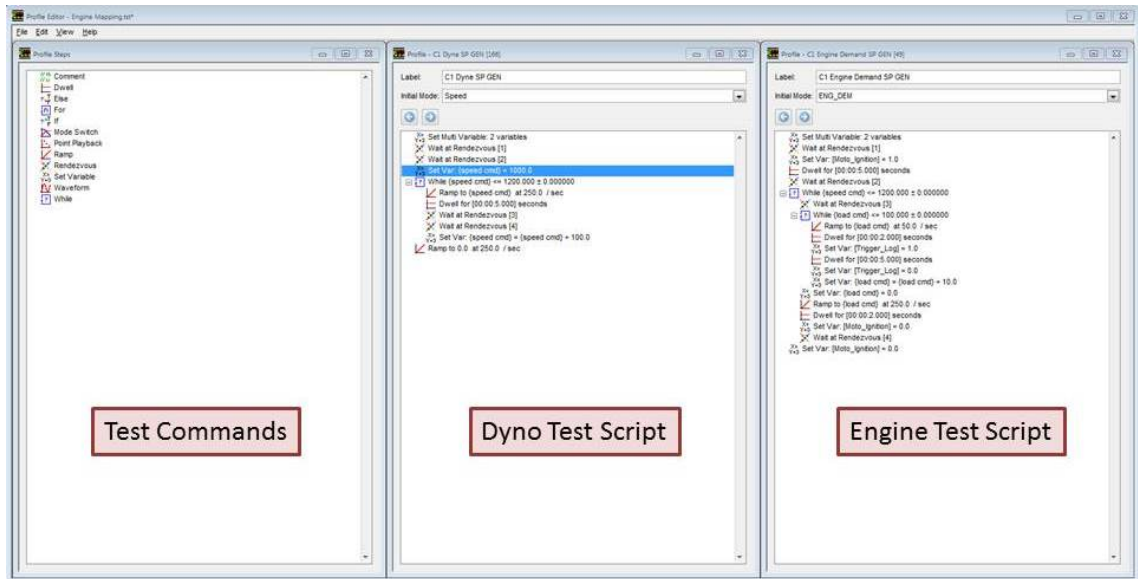


Figure 5.12: Image of the Inertia automated test editor utility window. The test shown is to run a speed - load sweep. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

There are several program flow commands that can be used to cycle through points and synchronize the two control groups. In addition, local test variables can be created for the use in for and while loops. For example, in Figure 5.12, the variable *load cmd* in the engine control group script is a local variable that is used to cycle through the different load set points. Local variables can be created by using the set variable command and setting the type to local. Browsing for a variable will then open a window where local variables can be selected or created. This is shown in Figure 5.13. In addition, rendezvous can be used to synchronize the two scripts. A script will pause execution at a rendezvous until both scripts have reached the rendezvous. The automated test script shown in Figure 5.12 will perform as 10-point load sweep from 10% commanded throttle to 100% commanded throttle at 3 speeds

starting at 1000 rpm and stepping by 250 rpm until 1200 rpm. These points can be modified by changing the step sizes and the limits on the while loops.

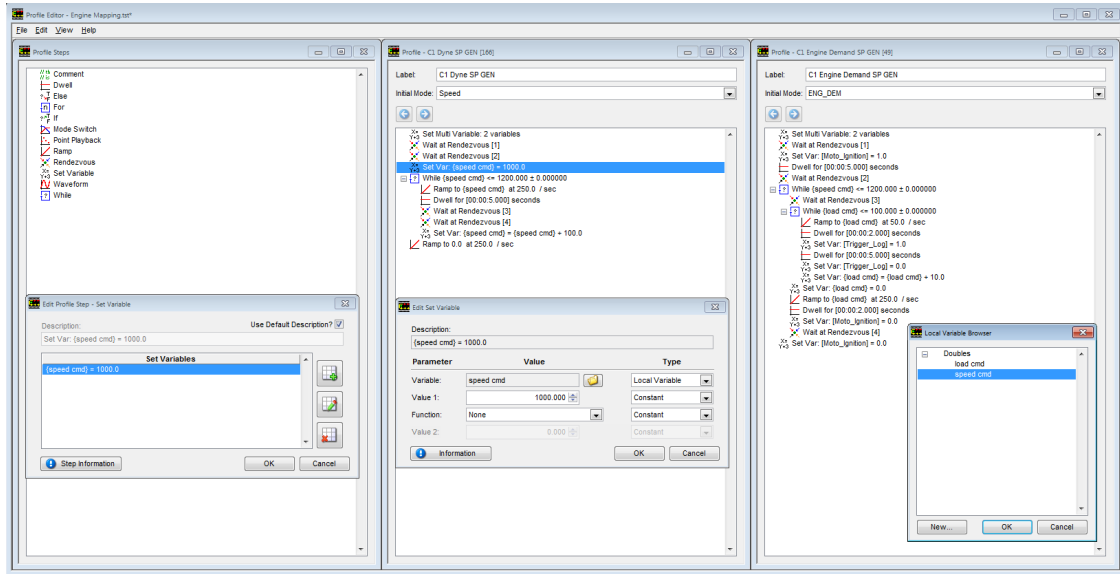


Figure 5.13: Screenshot of the popup windows used to create a local variable. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

Data can be recorded during an automated test by using the data logging setup menu in *Edit > Data Logging*. The setup is the same as the one used in the manual data logging example above except a trigger is used to start and stop the log file. A user variable has been created called 'logging trigger' that can be toggled on and off during the test scripts by using the *set variable* command. It is good practice to only trigger the logging from one control group. This will prevent data from being missed or repeated. The logging should be triggered from the nested loop. In the case of the scripts shown in Figure 5.12, the engine load loop is nested within the dyno speed loop due to the way the loops are executed. The engine load sweep is repeated for every speed rather than the speed sweep being repeated for every load.

Without the ability to send CAN messages to the powertrain controllers, automated testing is limited to load and speed sweeps. In order to conduct automated spark and fuel tests, CAN messaging needs to be used to send the ignition advance and fueling set points. In addition, engine startup is messy in automated testing since there are no programmatic startup enrichments in the engine control software. The best

practice is to have the engine started and warmed up before executing an automated test on the engine.

### **5.1.3 NI Veristand with Inertia Plugin Software Package**

The rear test cell uses NI Veristand 2011 to interface with the PXI system in the rear test cell. It should be noted that there are few issues with the 2011 version of Veristand. One major issue is that the data logging VI for Veristand 2011 is built in a version of LabVIEW that is not compatible with the 2011 version of LabVIEW. Attempts to fix the issue have failed and an update to a newer version of Veristand is needed to fix this issue. The following paragraphs will describe the different features of Veristand 2011 and how to configure the project to operate the rear test cell.

Currently, Veristand with the Inertia plugin is only on the third workstation computer. After Veristand is launched, the window shown in Figure 5.14 will pop up. From here, the project can be executed or configured using the different buttons on the bottom of the window. The active project is shown in the appropriately labeled window. The correct project is 'MTU Mobile Lab.nivproj' and can be found at '*.\Documents\Original Config Files for Veristand\MTU Mobile lab\*'.



Figure 5.14: Screenshot of the Veristand launch window. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

### 5.1.3.1 Veristand Workspace

When the project is run, Veristand will synchronize with the real-time controller and open the workspace shown in Figure 5.15. The layout of the Veristand workspace is very similar to that of the Inertia workspace discussed in section 5.1.2. Items on the left side of the workspace are for test cell control and high level feedback. On the right side of the workspace are the items specific to the screen being displayed. Currently there is only one screen because it is all that is needed to display information about the Yanmar. Additional screens can be added like in Inertia. The only difference is that the items on the left side of the screen cannot be made into default items and need to be added manually.



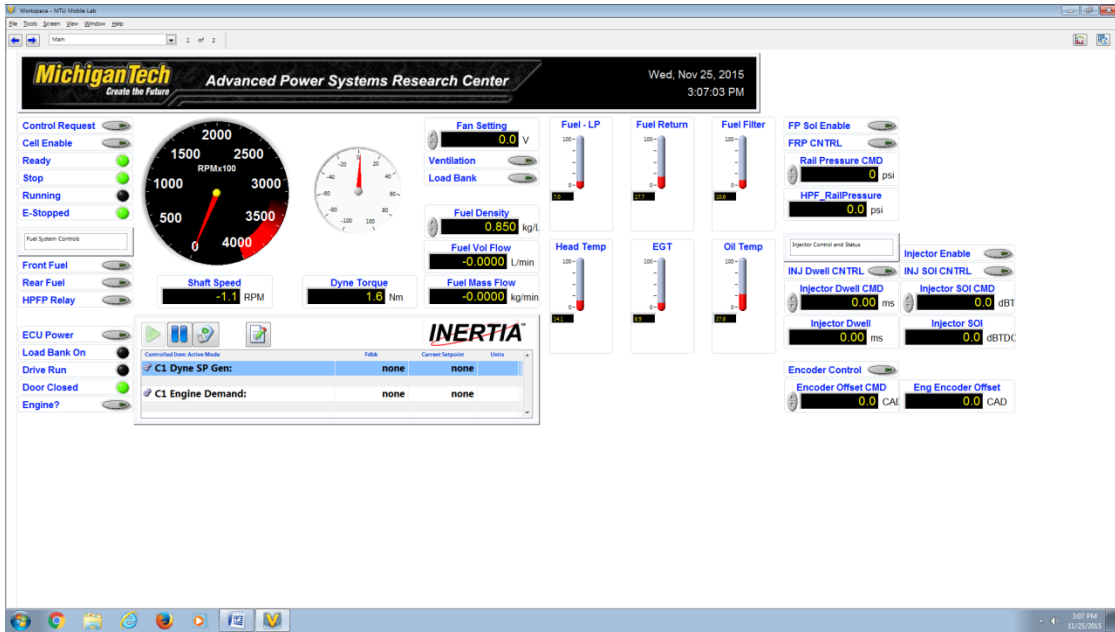


Figure 5.15: Screenshot of Veristand workspace. From here, the test cell can be monitored and controlled. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

Unlike in Inertia, the Veristand workspace needs to be put into an editing mode for screen items to be added or modified. This can be done from the menu bar by selecting Screen > Edit Mode or by using the keyboard shortcut *Ctrl + m*. Once in edit mode, a toolbar tab on the left side of the screen will appear. Clicking the tab will expand a menu with additional screen items that can be added to the workspace.

### 5.1.3.2 Project Explorer

Unlike Inertia, Veristand uses a project to bundle all of the files together. The project can be edited by opening the project explorer shown in Figure 5.16. The two main files of the project are the system definition file and the workspace or screen file. The project can be deployed and executed by using the run arrow on the toolbar or from the Operate menu. Additionally, the project can be deployed to the real-time target without running it on the local machine by using the deploy button on the toolbar or from the Operate menu.



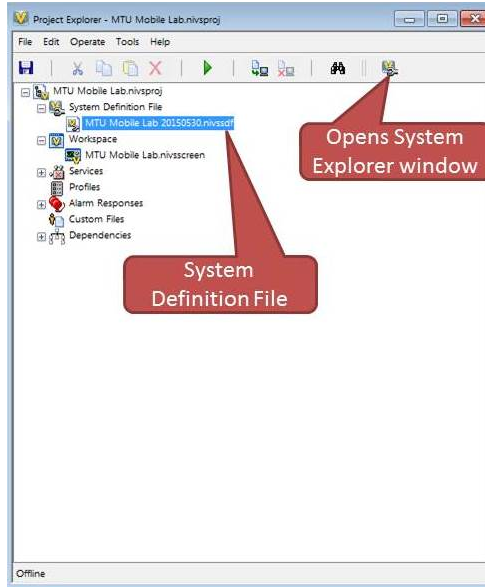


Figure 5.16: Screenshot of the Veristand Project Explorer window. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

### 5.1.3.3 System Explorer

If the system definition file is selected as shown in Figure 5.16, a button will appear on the toolbar. This button will open the system explorer which will allow the user to edit the configuration file. Figure 5.17 shows the main screen of the Veristand System Explorer window. This system explorer is very similar to the one in Inertia and will enable hardware and software configurations to be modified. There are a few differences in between the two System Explorers. One nice change is that the Veristand System Explorer has a window with detailed information and help for the selected item in the system tree. The hierarchy of the system tree is slightly different as well. There is still a branch for the target hardware and modules as well as clients. Unlike Inertia though, the clients only have control groups that are unique to them. The alarms and procedures are global items that are independent of the clients. Only one client is currently licensed for Veristand. Therefore, only one workstation computer will be able to control and view data from the rear PXI system at a time.

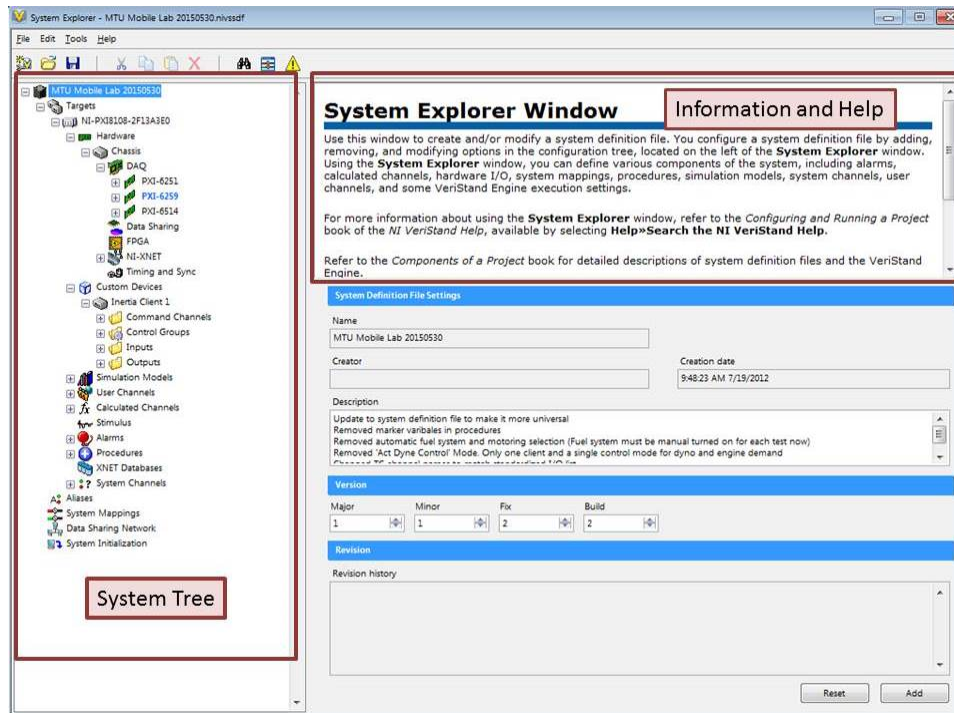


Figure 5.17: Screenshot of the Veristand System Explorer window. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

#### 5.1.3.4 CAN Control and Messaging

Currently, Veristand is setup to send and receive CAN messages containing engine commands and control parameters. This allows for complete control of the engine from Veristand without the need to use the Mototune interface. These control items and feedback parameters are located on the right side of the workspace as seen in Figure 5.15. To prevent erroneous changes to the operating conditions of the engine, control of each parameter has to be enabled before Veristand can take control over the CAN bus. These enable bits are located at the top of each control item cluster on the workspace.

An XNET CAN database file is used to define the messages being transmitted and received on the bus. This can be done in the System Explorer under the *XNET Databases* menu item in the system tree. The currently imported database can be seen and modified using the XNET Database Editor by clicking the toolbar button

with an image of a car and can after selecting the database. This will open the window shown in Figure 5.18.

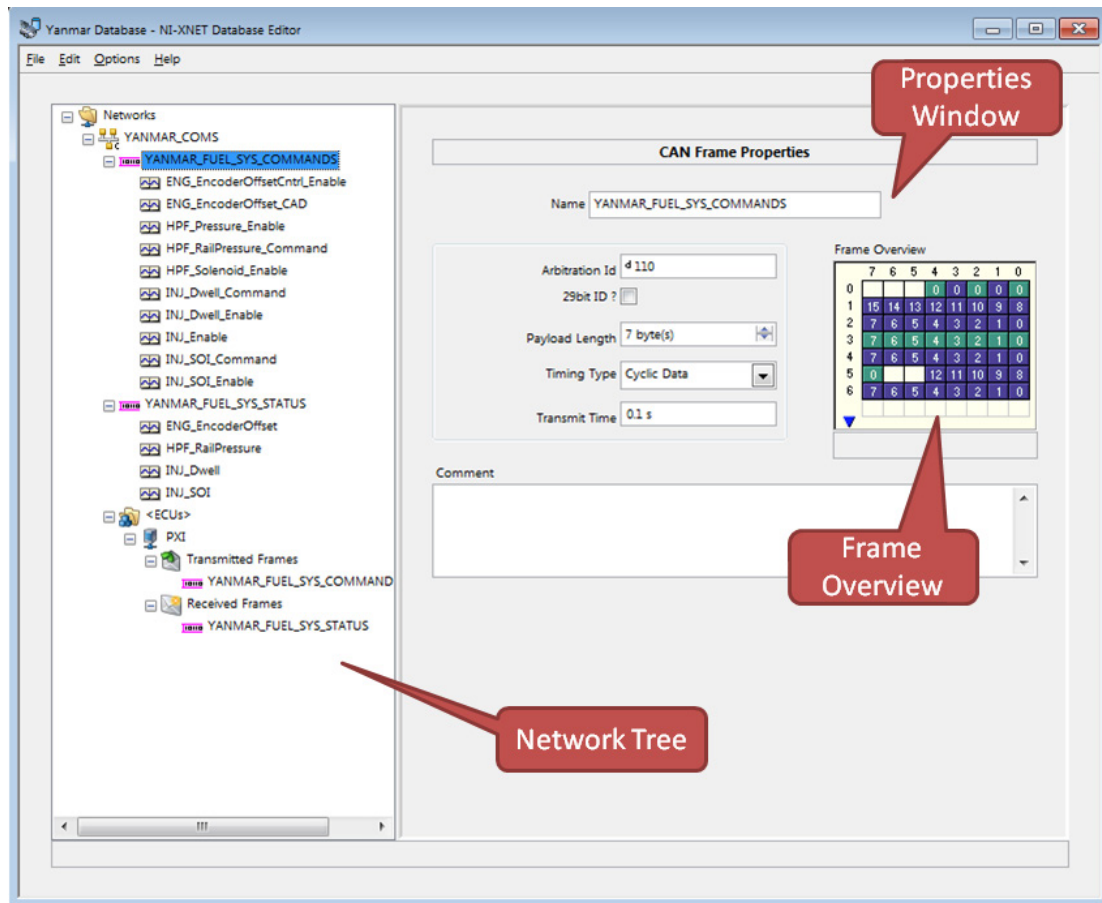


Figure 5.18: Screenshot of XNET Database Editor. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

The XNET Database editor can be used to add additional devices, clusters, messages, and signals. Clicking on each item in the database will open a specific window to modify the parameters of the item. For example, the window after highlighting a signal is shown in Figure 5.19. From here, the start bit, data length and byte order, as well as the scaling can be modified. All of this information should match the device that the message is intended for. Any item can be added by right clicking on the item it would belong to in the network tree on the left side of the window.

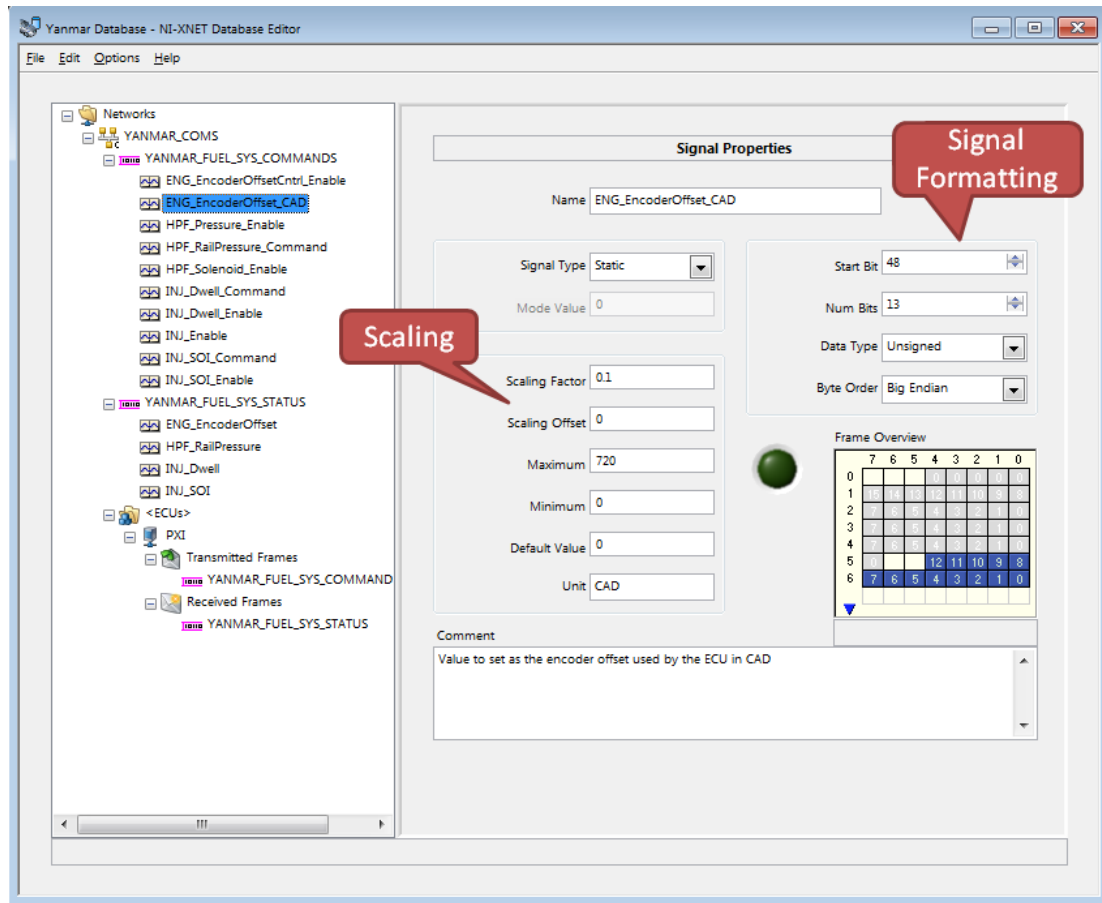


Figure 5.19: Screenshot of the XNET Database Editor with the signal properties window open. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

It is important to make sure that each message is associated with an ECU as either a transmitted or received message in the database as this will allow that message to be accessed as a device in the System Explorer. CAN messages can be added to the XNET bus from the database under the appropriate branch of the CAN port as shown in Figure 5.20. Additional ports can be added as the first port becomes over utilized. Additional messages can be added as either incoming or outgoing by highlighting the type of message under that branch and clicking the *Import Frames* button on the toolbar. This will need to be done every time that the database is edited. Each type of message contains some information on how to do this as seen in the screenshot. After adding the messages to the XNET hardware bus, the signals can then be accessed and displayed on the workspace.

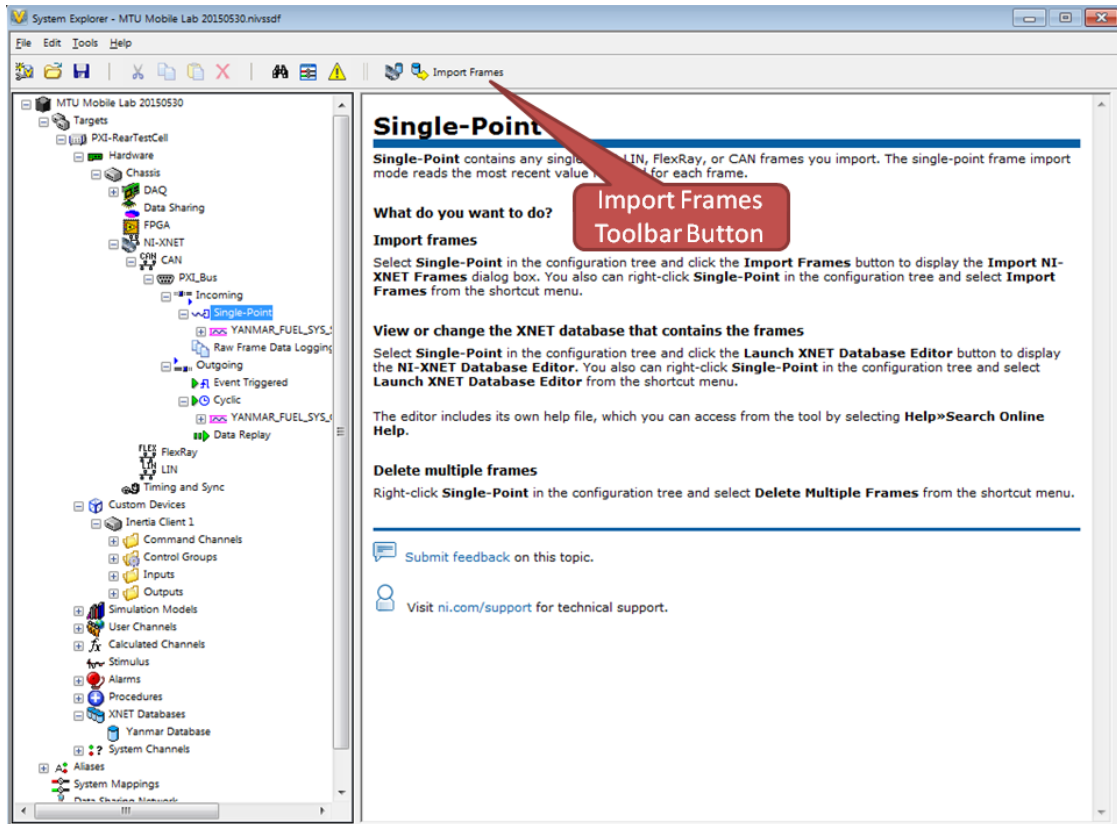


Figure 5.20: Screenshot of the System Explorer showing the XNET bus expanded in the system tree. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

### 5.1.3.5 Data Logging

Data logging does not work in Veristand 2011. The issue is that the VI for the embedded data logger was built in LabVIEW 2009 and all attempts to build the VI in LabVIEW 2011 have failed. After discussions with NI, upgrading to a newer version of Veristand is the fix to this issue.

### 5.1.3.6 Automated Testing

The Inertia plugin for Veristand 2011 comes with an automated testing tool similar to the tool in Inertia 6.1 outlined in section 5.1.2.4. The test editor can be found in the tools menu under *Tools > Test Editor*. This will open the test profile editor window shown in Figure 5.21 below. This editor will allow you to create new tests and modify existing ones using the commands from the list on the left. Some of the

commands from the list are used for the control object while other commands allow you to set other variables and outputs. For more information on creating a test, refer to section 5.1.2.4. When creating a new test, you will be prompted to select which control objects you want to use in the automated test. The test shown in the figure only uses the dyno control object so there is only one window.

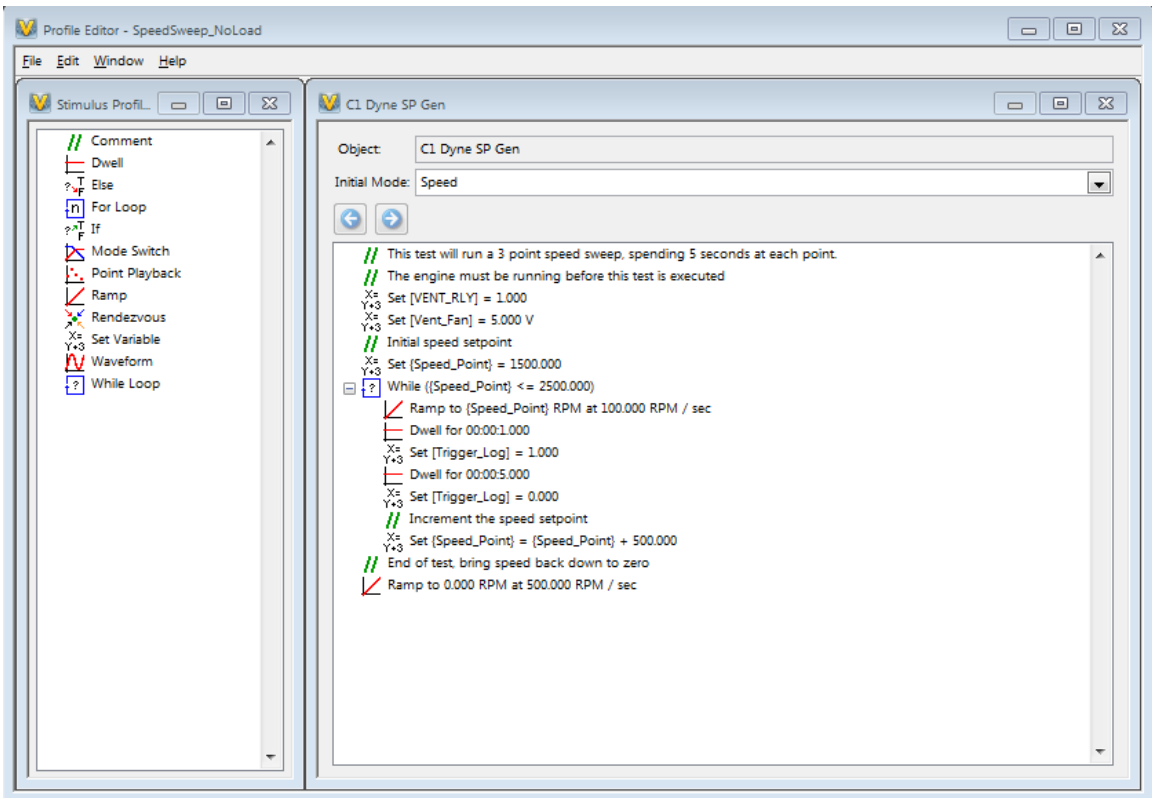


Figure 5.21: Screenshot of the Veristand automated test profile editor. This tool is a part of the Inertia plugin. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

Data logging can be setup in an automated test by using the tool found in the *Edit > Data Logging* menu. This tool works even though the embedded data logger does not. A screenshot of the setup window for data logging during an automated test is shown below in Figure 5.22.

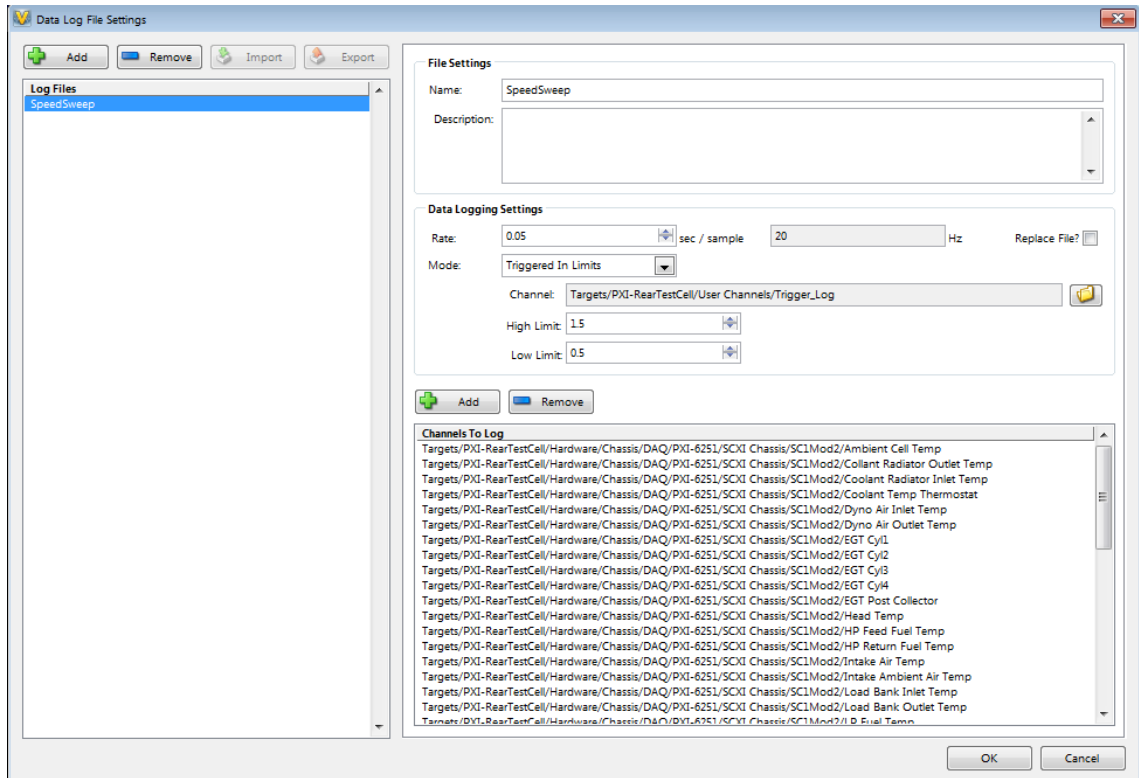


Figure 5.22: Data log file settings window for the Veristand automated test function. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

The test scripts can be executed using the test header found under the *Tools > Test Header Example* menu. A screenshot of this tool is shown in Figure 5.23.

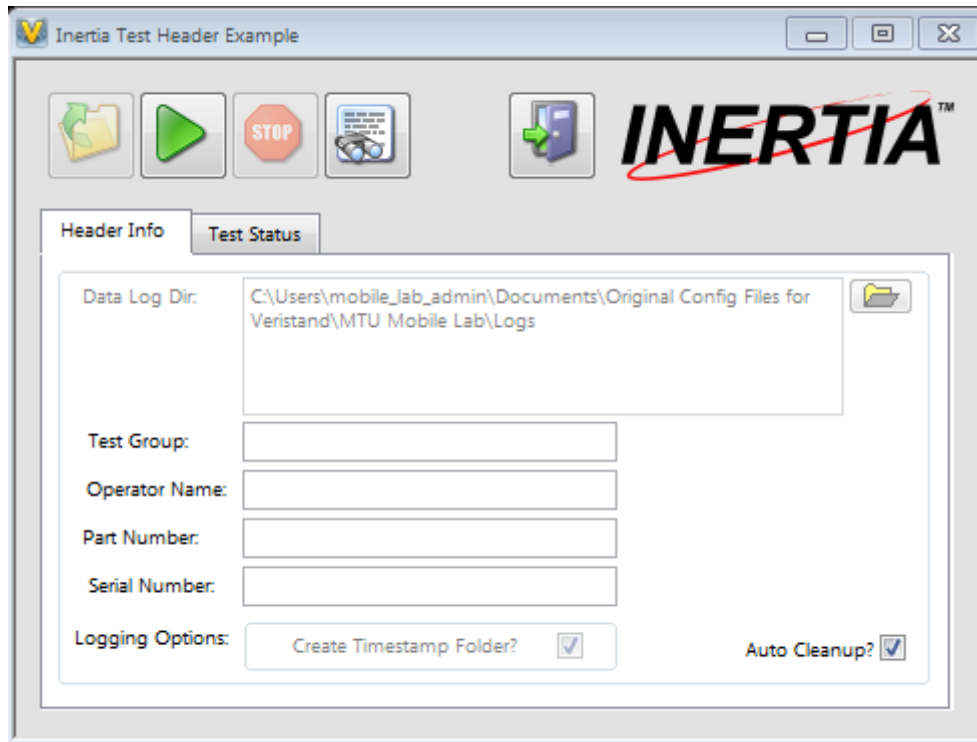


Figure 5.23: Screenshot of the Inertia automated test execution tool in Veristand. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

## 5.2 Hybrid Control System

The hybrid control system is only used in the front test cell. This system acts like the powertrain control module and has the same functionality as the controller on the CHEV with minor modifications for use in the test cell. The main functionality of this control system is to determine the torque split between the engine and electric motor depending on the motoring selection. The motoring selection can be either engine only, motor only, or hybrid where both the engine and motor split the torque delivery depending on the torque split calibration table. This system is mounted on the wall above the PXI cabinet and can be seen in Figure 5.24 below.



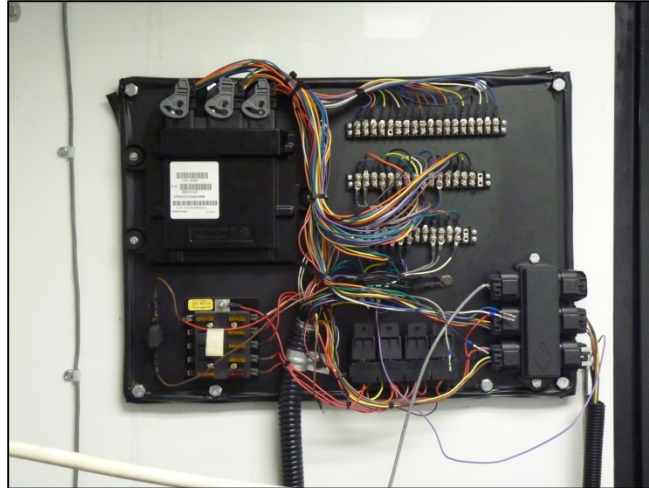


Figure 5.24: Image of hybrid control unit mounted on breakout panel in front test cell

The modifications made to the CHEV code for use in the test cell are very minor and mostly deal with input and output handling. The user request is received as a 0 – 10V analog signal from the PXI controller rather than the analog output from the accelerator pedal used on the CHEV. The other change is in how the engine throttle is controlled. The CHEV is still currently using the stock ECU with a stepper motor to control the throttle plate position. The throttle plate position on the CHEV is directly controlled via the hybrid controller. In the test cell, an open source controller is used for the ECU. This ECU controls the throttle position based on a request from the hybrid controller over the CAN bus.

### 5.3 SI Engine Control System

Like the hybrid control system, the SI engine control system uses a MotoTron® 112-pin controller. This system is mounted in the 19" rack as shown in Figure 5.25. Only the necessary pins for controlling the Kohler engine are broken out to DIN rail terminals. This is due to the limited space on the rack. There is room to add more terminals if needed in the future. The ECU pins that are broken out on each DIN rail are outlined in Table 5.3. The channel pin refers to the pin location on the ECU in the format of  $X-CR$  where  $X$  is the connector pocket,  $C$  is the pin column in the pocket, and  $R$  is the pin row. The orange boxes in the jumpers column indicates what

channels and pins are connected together on the DIN rail by the orange jumpers seen in Figure 5.25.

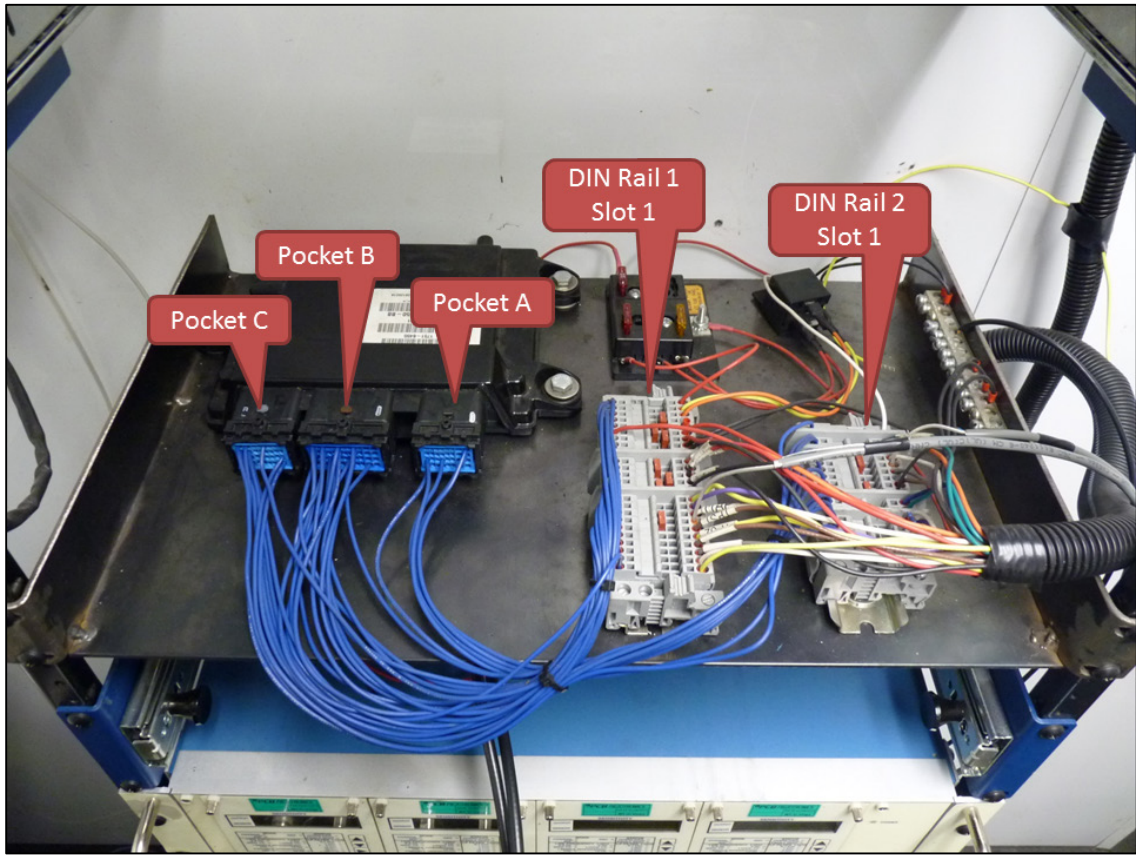


Figure 5.25: MotoTron ECU used to control the engine in the front test cell.

Table 5.3: Engine controller pinout and DIN rail connections

DIN Rail 1		
Channel Pin	Jumper	Channel Name
B-A3		Battery (12V)
A-D3		MPRD
C-G3		Drive Power 1
C-H3		Drive Power 2
B-G4		Key Switch
		12V power
C-G1		Battery ground
C-G2		Battery ground
C-D4		Sensor Power (5V)
		TPS motor (5V)
B-D3		Sensor ground
		TPS motor ground
		DCAT MAP ground
B-J2		Crank sensor (VR +)
B-J1		Crank sensor (VR -)
B-E3		MAP
		MAP to DCAT
B-F1		TPS 1
B-F2		TPS 2
C-A1		IAT
C-A2		Oil Temp
		E-Stop from H-ECU
B-E4		Lambda

DIN Rail 2		
Channel Pin	Jumper	Channel Name
B-H3		E-Stop
C-H4		Throttle control -
C-G4		Throttle control +
A-A3		Spark CMD Cyl 2
		Spark CMD to DCAT
A-A4		Spark CMD Cyl 1
B-L4		Spark CMD GND
		Spark GND to DCAT
A-H1		Injector Cyl 2
A-H2		Injector Cyl 1
B-A1		CAN1 High
B-A2		CAN1 Low
B-C1		CAN2 High
B-C2		CAN2 Low

The ECU can be monitored and calibrated through MotoTune. The calibration and display screens can be seen in Figure 5.26. The display screen can be used to turn on spark and fuel as well as override control and sensor signals. The display screen also allows the user to monitor the sensor and control values of the ECU. Additionally,

the calibration maps for injector dwell and ignition advance can be modified through MotoTune.

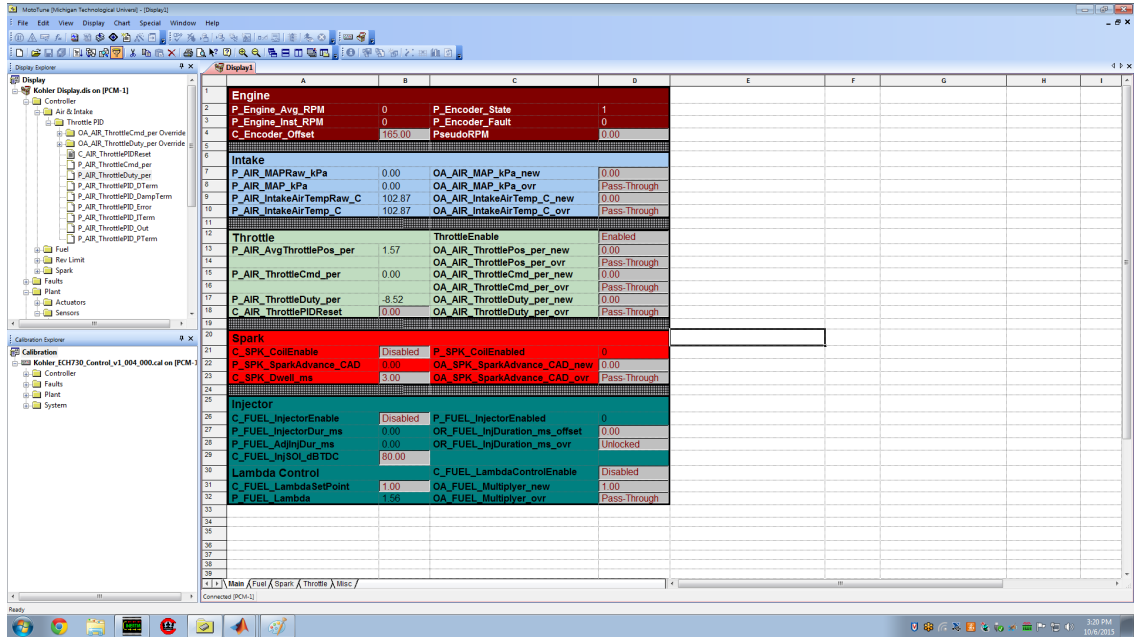


Figure 5.26: MotoTune display and calibration windows for the Kohler engine.

The control system is pretty basic. The following paragraphs will discuss in brief how the throttle, fuel, and ignition systems are controlled. The throttle is controlled using a PI loop. The commanded position comes over the second CAN bus and compared to the average position measure by the two throttle position sensors. The throttle command from the CAN bus can also be overwritten so that it can be controlled through MotoTune.

The ignition system has the ability to operate in set point or open loop control modes. The default mode is open loop. The ignition coil dwell is controlled through a set point in either mode and operates the same independent of the mode of operation. In open loop mode, the ignition advance is determined using a 2-D lookup table with speed and manifold pressure as the inputs. The ignition system can also be controlled by a set point using an override.

The fuel system is controlled in a very similar fashion except that there is also a closed loop mode of operation. The open loop control uses a 2-D lookup table with

speed and manifold pressure as the inputs to determine the amount of fuel to inject. There is also an option to offset the amount of fuel being injected using an override. This can either be an adaptive offset where the value from the calibration table is being constantly offset or fixed where the initial value is locked and doesn't update as speed and load changes. Additionally, closed loop fuel control can be enabled where the wide band O2 sensor is used to determine a multiplier that acts on the amount of fuel injected. This multiplier acts on both the calibration table and the offset value. The injection timing is set using a control and doesn't change in the different control modes.

## 5.4 CI Engine Control System

To control the CI engine in the rear test cell, the hybrid control system was re-wired and re-flashed for engine control. The CI engine control system is responsible for controlling the engine as well as the high pressure fuel system.

The high pressure fuel system is partially controlled by the engine controller. The pump motor is turned on by the PXI but the pressure is controlled by the MotoTron controller. This is done by controlling the duty cycle of a bypass valve on the high pressure pump. A pressure transducer in the high pressure rail is used as the PI loop feedback.

Set point control is used for the engine. The code was originally written to operate the engine in a genset and contains a closed loop speed control algorithm. This function still exists but is not necessary when the dyno is operated in speed control mode. Both the start of injection and duration can be specified for the injector.

The MotoTune display is shown in Figure 5.27. This display is formatted so that the control and probe items can easily be distinguished from one another. The control items are recessed into the screen while the probe items are raised. The display is also split up into different sections depending on the system that the variables belong to. Engine feedback and encoder settings are on the top, followed by the high pressure fuel variables in the middle, and the injector and speed control variables on the bottom.

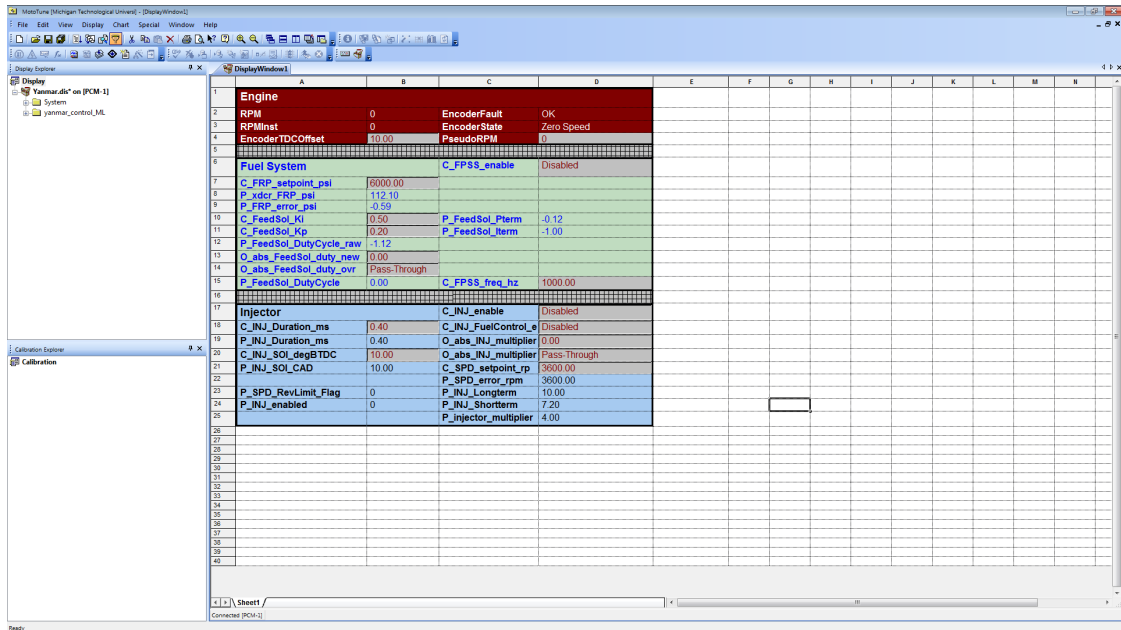


Figure 5.27: Screenshot of the MotoTune display used to control the Yanmar engine.

## 5.5 Combustion Analysis

### 5.5.1 NI Combustion Analysis Hardware

Combustion data is collected using an NI cDAQ running DRIVVEN  $\mu$ DCAT. As seen in Figure 5.28, there are three modules used to collect and process combustion data. An NI 9411 encoder module is mated with a 360 ppr optical encoder to measure the absolute position of the crankshaft. The other two modules are both NI 9215 4-channel synchronous AI cards used to collect cylinder pressure data and other supporting measurements. With the modules that are currently installed, the system is capable of collecting cylinder pressure data from four different cylinders or sensors. This number can be later expanded by adding extra 9215 modules in the vacant slots.

The second 9215 module seen in the figure is used to collect additional data needed for combustion measurements. The first channel is used to measure the manifold pressure for cylinder pressure pegging. The second and third channels are for the injector and spark command signals. The spark signal measurement is used to determine combustion delay. The last channel is left open.



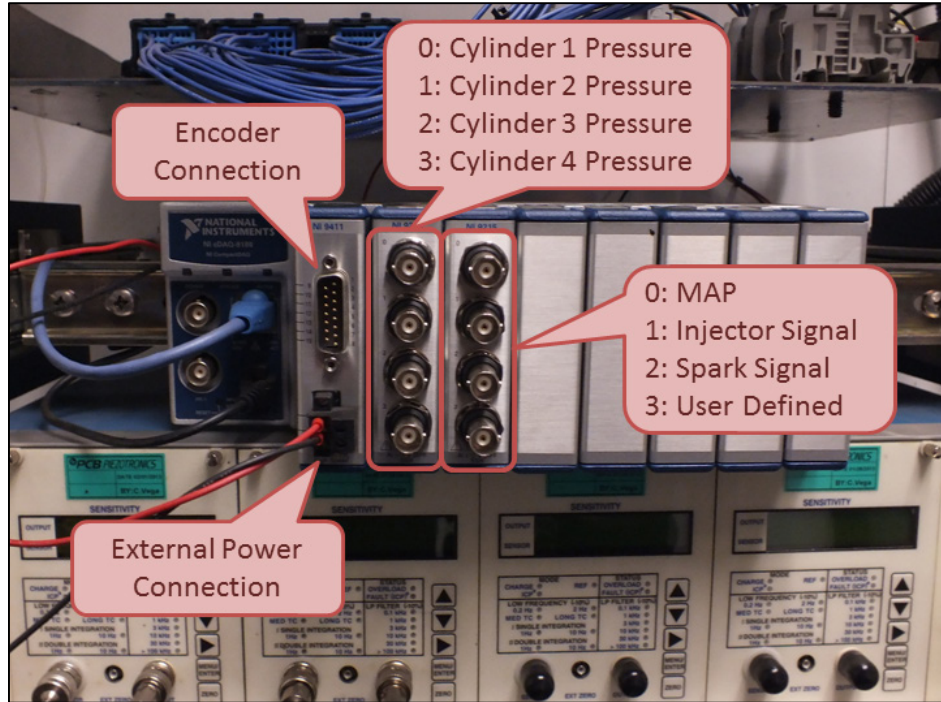


Figure 5.28: Combustion analysis cDAQ unit mounted in the 19" rack in the Mobile Lab test cell

### 5.5.2 PCB Charge Amplifiers

PCB charge amplifiers are used to condition the signals from the in cylinder pressure transducers before being read by the cDAQ unit. There are four PCB 443B102 charge amplifiers located in the 19" rack below the combustion analysis cDAQ. There is a display on the front panel of each charge amp that indicates the current settings using a combination of LED's and an LCD display. The display will indicate the settings for sensitivity and output scaling. The LED's indicate the time constant and filter settings as seen in Figure 5.30. The charge amp settings can be modified by using the menu and navigation buttons on the front panel shown in the picture.

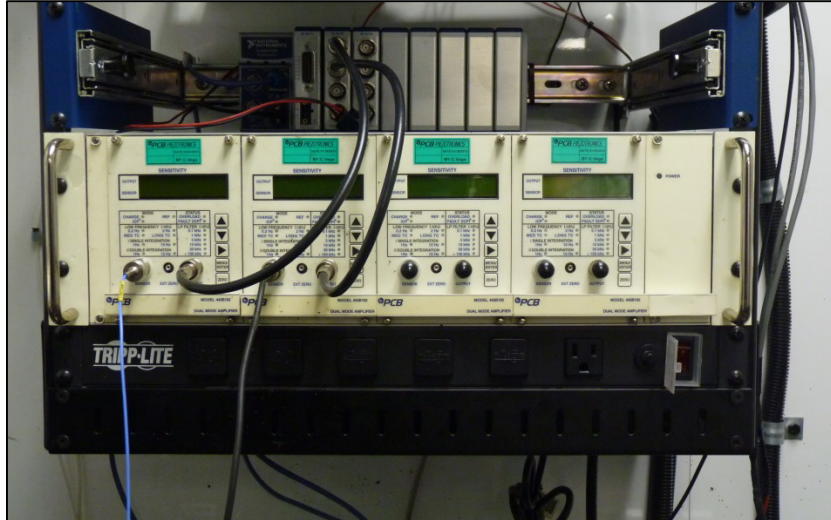


Figure 5.29: PCB 443B102 charge amplifiers.

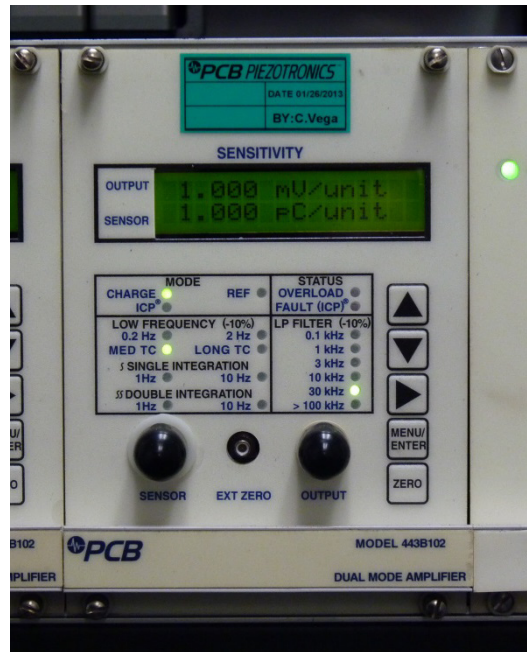


Figure 5.30: Front panel of the PCB 443B102 charge amplifier.

Additionally, the charge amps can be remotely configured using a LabVIEW executable called 443 Software. A shortcut is located on the desktop of each workstation computer. There is a serial port on the back of the charge amp chassis for communication to the computer. Since the workstation computers do not have a serial port, a USB to serial converter is needed. Using the software will allow the charge amp settings to be modified while a test is in operation. Otherwise, the test



would need to be stopped and the cell powered down to access the charge amp front panel. After selecting the slot number of a charge amp, the desired setting can be changed and transmitted to the device. As noted in Figure 5.29, each charge amp occupies two slots on the chassis. The left most module is in slot 0, the next module is in slot 2 and so on.

### 5.5.3 $\mu$ DCAT Software Package

The software package used for combustion analysis is  $\mu$ DCAT. After the program has been launched, the workspace will be opened in the last state that it was saved. The  $\mu$ DCAT workspace can be seen in Figure 5.31. Additional tables and plots can be added to the workspace by double clicking on any item under the *Tables* or *Plots* tabs on the main window. The windows that are already opened are displayed in bold under these tabs. The program can be in two states, setup and run. In the setup state, the menus under the setup tab can be used. In the run mode, these menus are grayed out. The engine setup menu is where details on the engine and encoder can be entered. The engine setup menu is shown in Figure 5.32.

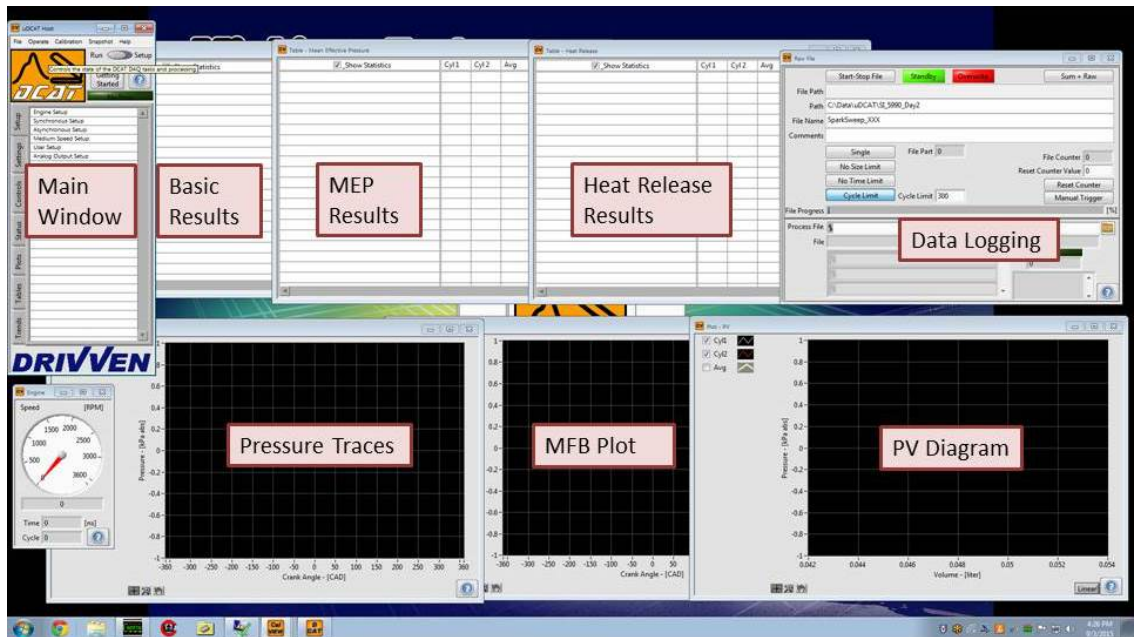


Figure 5.31: Screenshot of the  $\mu$ DCAT workspace. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

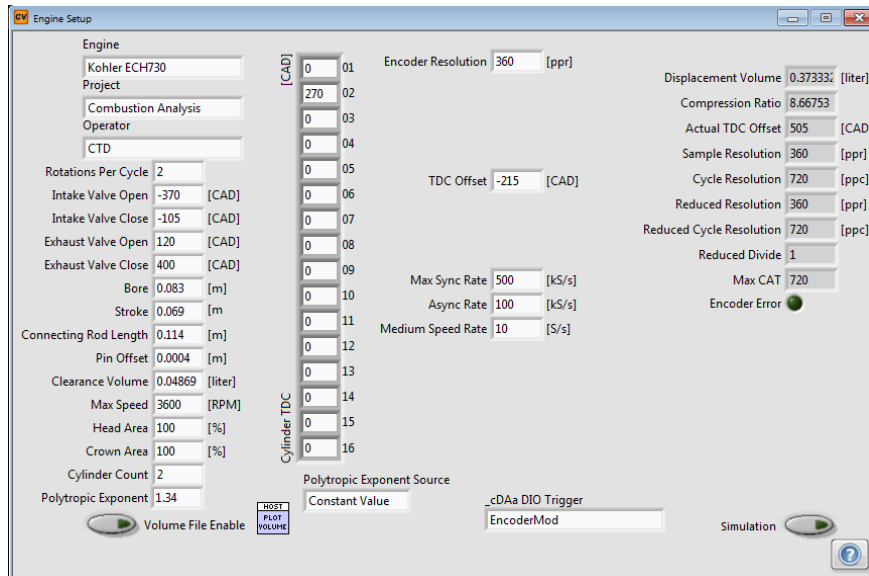


Figure 5.32: Engine setup menu. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

The synchronous setup menu is shown in Figure 5.33, and is where the measurements taken at the synchronous rate can be configured. Signal names and scaling can be entered as well as disabling certain signals. Being able to disable certain signals is useful for the user defined channel on the second 9215 module. Multiple channel setups can be defined and disabled until they are used. It is important to define a signal type and cylinder for each signal. The signal type is used to determine what signals to use for the different calculations. The other menus under the setup tab are for channel setup at the different measurement rates.

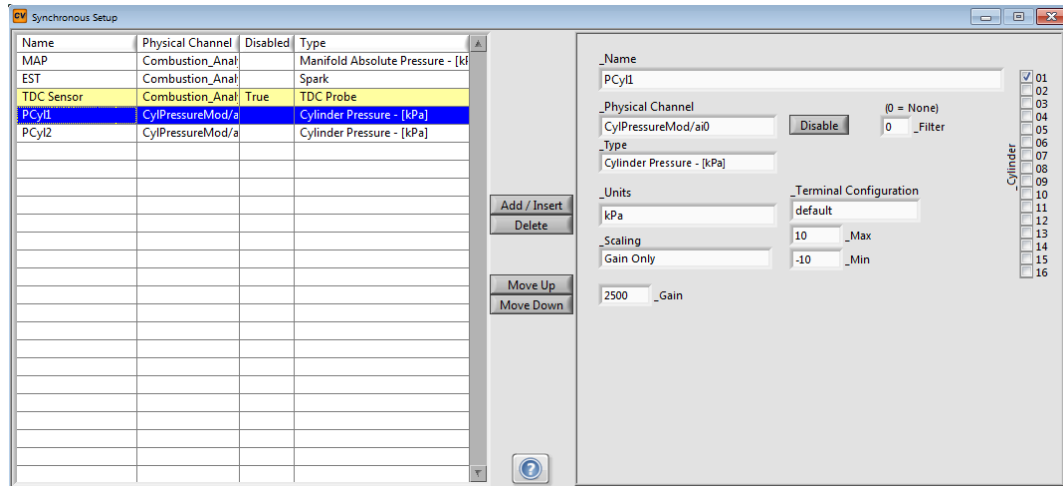


Figure 5.33: Synchronous setup menu. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

When  $\mu$ DCAT is in run mode, the other menus become active and data starts to be collected. The synchronous channels will not update until a pulse is seen from the encoder. The settings tab is where changes to the combustion calculations can be made. This tab is where menus for changing pegging, heat release, misfire, knock, and other calculations can be found. Each menu has a help button for more details on the options for each calculation. The controls tab has a menu for selecting which calculations are performed. Unnecessary calculations can be turned off to reduce the stress on the processor. The remaining tabs can be used to add new windows to the workspace to display different signals and calculations.

### 5.5.3.1 Standard Operating Procedure

The procedure for starting the combustion analysis system is laid out below. It is always a good idea to open up the engine settings window and check the name displayed for the DIO Trigger. It should list the name for the 9411 module.

1. Power on the charge amps and cDAQ using the switch on the power strip mounted in the rack.
2. Use NI MAX to perform a self-test on the cDAQ. The cDAQ is located under *Devices and Interfaces > Network Devices*.

3. Start  $\mu$ DCAT from the start menu under *Calview 3.11 > uDCAT Host*. A window will pop up asking which calibration to run. Select the appropriate calibration file when the window pops up on the screen. The calibration file for the Kohler engine is *Combustion Analysis*. The calibration file for the Yanmar is *Yanmar*. Note that this step has a timeout of about 1 minute after which the program needs to be relaunched.

It is best practice to put  $\mu$ DCAT in run mode after the engine is already spinning. It has been observed that the encoder phasing gets messed up when leaving zero speed. This has been especially true when spinning down and then back up without cycling in and out of run mode.

#### 5.5.3.2 Data Logging and Post Processing

Data is recorded using the logging window on the workspace. Limits can be applied to the length of the data log depending on time, size and number of cycles. Typically data is recorded for 300 cycles. There are two types of data files that can be recorded, a raw file and a summary file. A raw file contains the raw signals from the sensors and the settings for the calculations. A raw file does not include any calculations such as mean effective pressures or heat release. A summary file on the other hand contains the processed data and the resulting calculations. Both files are in a *.tdms* file format so there is no way to distinguish between the two without opening the file and inspecting the file type property. It is recommended to record the raw data and use the  $\mu$ DCAT batch processor to process the raw files. The batch processor, shown in Figure 5.34, can be found in the start menu under *DRIVVN > DCAT > Batch Process Raw File*.

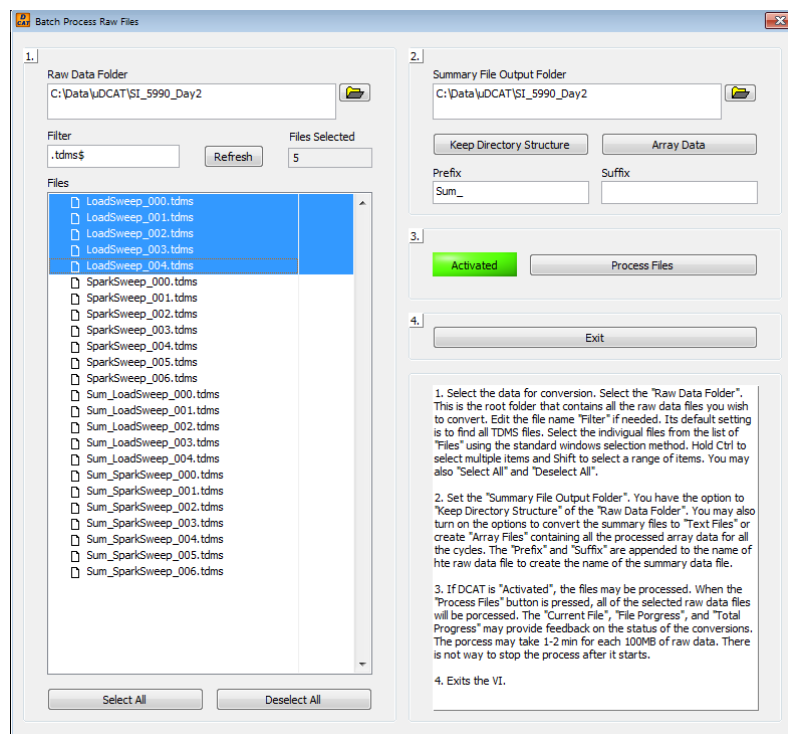


Figure 5.34: Screenshot of the Batch Process Raw Files program. Instructions are in the text box. A prefix of *Sum\_* should be added to help identify the output files as summary files. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

Additionally, calculation settings can be modified using the Raw File Viewer V2 program found on the desktop of each computer. This program can be used to view the data and calculation results of raw files and make changes to the calculation settings. A screenshot of the program can be seen in Figure 5.35. The raw file viewer is also capable of saving summary files but only a single file at a time. The batch processor should be used to process data sets that don't need corrections in post.

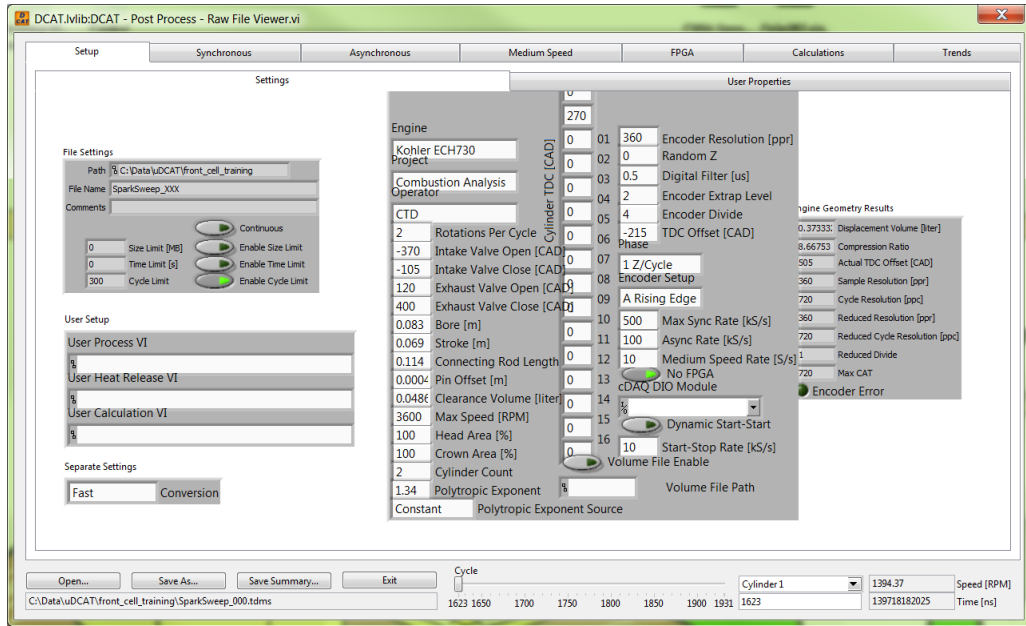


Figure 5.35: Screenshot of the Raw File Viewer program. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

## 6 Course and Experiment Development

Several experiments have been developed and used during the on campus and professional development courses. These courses include experiments with engines, batteries, electric machines and the vehicles. The setup and execution of these experiments are outlined in the following sections.

### 6.1 Electric Machines Experiment

The electric machine experiment is used in the ME4296 lab. The objective of this lab is to reinforce electric machine and power electronics concepts introduced in the lectures. Students perform a speed/load test matrix on a DC electric machine. The data from this test is then used to determine the performance metrics of the motor such as the torque/speed constant.

The electric machine experiment is conducted in the front test cell of the Mobile Lab. Both the 48V battery pack and D&D electric machine need to be installed in the test cell. It is best practice to decouple the engine from the dyno by removing the belt. This will reduce the wear on the engine as well as reduce the load on the electric machine. Since the torque meter is installed on the stub shaft of the dyno, the frictional losses due to the engine cannot be quantified during the test. Follow the standard operating procedures for inspecting the test cell and powering up the dyno drive and PXI. There is no need to setup combustion analysis for this experiment. If the engine is decoupled, then the engine oil pressure alarm needs to be disabled. This can be done by clicking the *Engine?* Boolean control at the very bottom of the control column on the left sides of the Veristand and Inertia front panels.

For this experiment, the Inertia workspace should be switched to the electric machine screen. This is the second screen in the list. The motoring selection should be changed to E-machine. This cell needs to be enabled with the motoring selection switch to E-Machine. The motoring selection cannot be changed while the cell is enabled.

The experiment can be run in two ways. The first is to run the test manually. The speed can be set using the Speed Control screen item and the load can be controlled using the Accelerator Pedal screen item. Data is logged using the manual data logger. There is already a logging file set for the electric machine called *E-Machine.lfs* found in *C:\Data\Inertia*. The only thing that needs to be modified in the file set is where the data is being saved.

The second method to run the experiment is to use the automated test tool. There is already a test script created called *Emotor Mapping.tst* in the test file directory. The test takes about 10 to 20 minutes to complete depending on how many points are run. The only changes that need to be made to the test file are where the data is being saved as well as the speed and load set points. This can be done by the students to help them become familiar with the test scripts.

## 6.2 Batteries Experiment

The battery experiment is used in ME4296. The objective of this experiment is to understand the operational theory of batteries and their mathematical model. During this experiment, students deplete the battery under constant load to determine the internal resistance as well as the capacity of the battery. An additional part of this experiment has the students build a vinegar battery. This part of the experiment is not discussed here however.

This experiment is identical in hardware setup as the electric machine setup in section 6.1. The only differences are in the software setup. The Inertia workspace should be switched to the battery screen which is the third in the list. There currently is no automated test scripts for this experiment so the test has to be run manually. There is nothing preventing a script from being written to perform this test automatically; however it would be hard to keep the students engaged for the test. The test takes approximately an hour to complete depending on the type and health of the 48V battery pack. Data should be logged continuously using the *Battery.lfs* logging file set found in *C:\Data\Inertia*. The only thing that needs to be modified on the file set is where data is being saved.



Currently, there is only one good set of batteries for the Mobile Lab. This makes running the experiment a bit of a hassle. Since there are two parts to this experiment it is best to run both groups in the test cell at the beginning of the first lab section, charge the batteries and have both groups in the second lab section to run the test cell at the end of their time slot. The test cell portion takes about 45 minutes to run so this allows for about 2 to 2.5 hours for the batteries to charge between tests.

### **6.3 Thermal Modeling Experiment**

The thermal modeling experiment is part of the ME5296 lab. It is part of a larger project where the students develop a thermal model of the HMMWV exhaust system in RadTherm. The experiment is used to collect real world data of the HMMWV system so that the students can calibrate and validate the model.

The thermal modeling experiment currently utilizes the HE-HMMWV to take thermal measurements of the exhaust system. The measurements taken on the vehicle are then used to calibrate a model in RadTherm. The experiment needs to be run with the vehicle in a stationary position. This poses as problem since the engine is cycled on and off based on the battery state of charge. Before the lab session begins, the HE-HMMWV should be driven around in electric mode until the batter SOC drops below 70%. This will ensure that the engine is on when in hybrid mode while being stationary.

The thermal measurements are taken using a cDAQ in the cab of the vehicle. This cDAQ receives data from the vehicle CAN bus using an NI 9862 CAN module. The vehicle has a messaging breakout box that is used to connect to the vehicle CAN bus. The correct serial connection is shown below in Figure 6.1. Thermocouple measurements are taken using an NI 9213 thermocouple module.

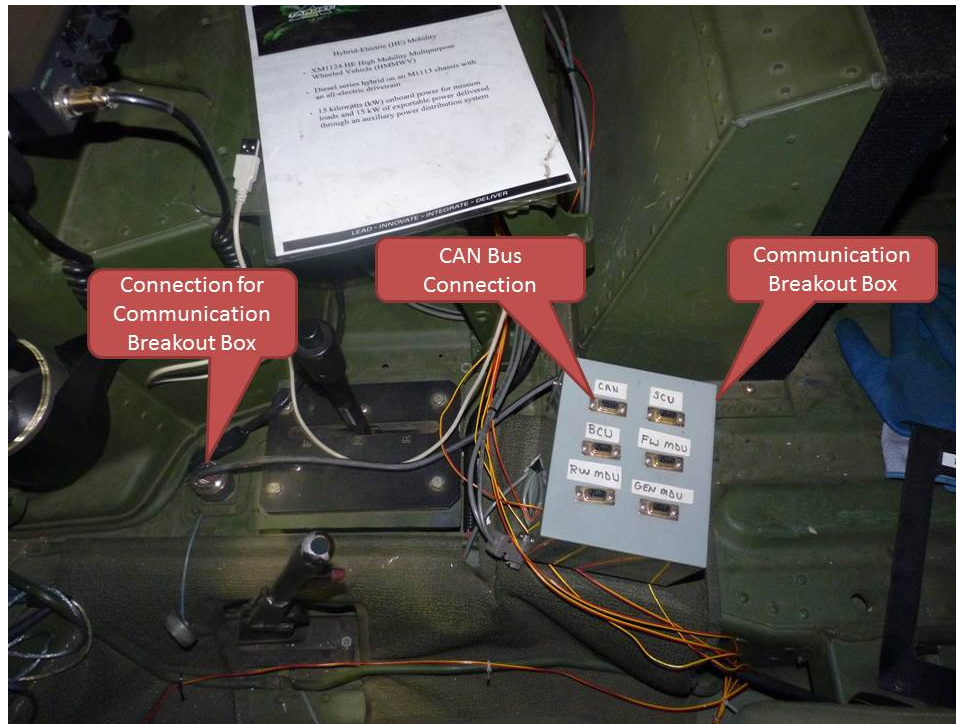


Figure 6.1: Picture of the communication breakout box for the HE-HMMWV. The DB-9 connector used to communicate over the CAN bus is the top left connector on the box.

The LabVIEW front panel for collecting the data is shown in Figure 6.2 below. The program is located on every vehicle laptop in *C:\Vehicle Logging\HMMWV*. The most up to date version is *HMMWV\_Logging\_v2*. There are additional versions located in the previous directory that only include CAN data or thermocouple data.



Figure 6.2: Virtual instrument front panel for logging data on the HE-HMMWV. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

In addition to the CAN data, manifold pressure also needs to be measured and recorded. Engine data is not being transmitted over the CAN bus so a manometer needs to be used to calculate the manifold pressure and then correlated to flow using the relationship found in the lab manual. Additionally, approximately 2/3 of the inlet to the air filter needs to be blocked off with tape. There are marks on the inlet to show where to tape as seen in Figure 6.3. A manometer can then be attached to the nipple marked with tape as seen in Figure 6.4. Simply replace the hose connected to it with the manometer hose. There should be a pressure differential of approximately 5"-7" of water.



Figure 6.3: Inlet of HMMWV air filter with marks showing where to block off with tape. The tape should go on the side of the line with the arrow. There is another line and arrow on the hidden side of the inlet.

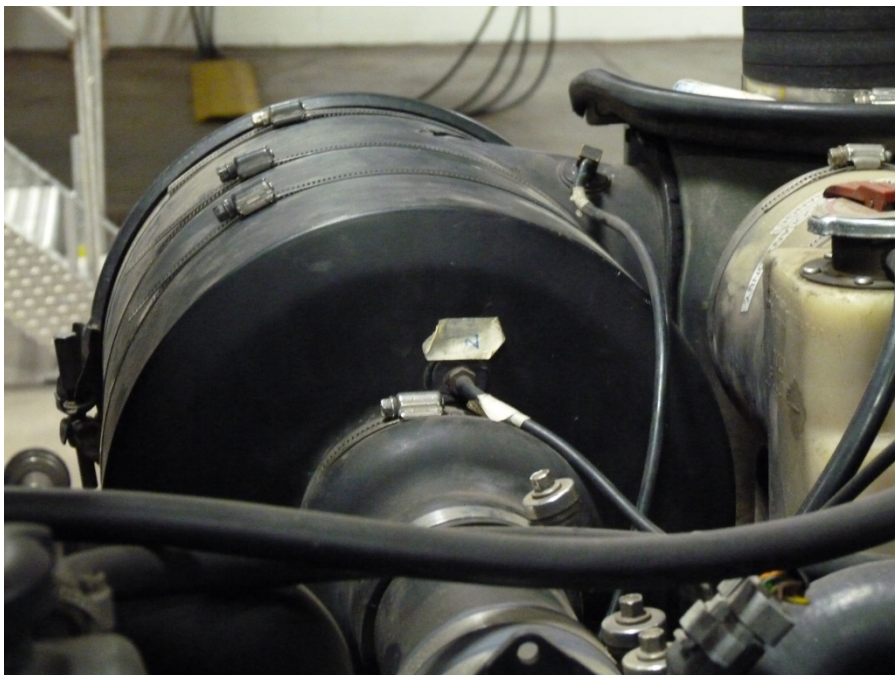


Figure 6.4: Picture of the HMMWV air cleaner indicating the port to connect the manometer to. The port is the one with a piece of tape above by the air cleaner outlet.

## 6.4 CAN Communication Experiment

The objective of the CAN communication experiment is to introduce the students to the CAN protocols on a modern vehicle and allow them to practice using binary. The goal is for the students to leave the lab with an understanding of how the SAE J1979 protocol can be used to collect data from the vehicle and how some of the data might already be transmitted over the OBD CAN bus. This is done by showing them the raw messages on the CAN bus and then making them request some information from the ECU using the methods outlined in the SAE J1979 protocol.

The CAN communication experiment is a simple experiment that utilizes the Chevy Malibu. There is no particular reason that the Malibu needs to be used. Any other vehicle with an OBD II port with a CAN bus will work. An OBD II to serial cable is needed for the communication with the vehicle. Currently the Volt, Vue, and Malibu have these cables installed in the vehicles and routed to the trunk space where the NI cDAQ is mounted as shown in Figure 6.5. The engine will need to be run briefly during this experiment so it should be setup by a ventilation drop point or outside. The only tools needed for this are a Kvaser Leaflight and a laptop with CanKing installed. All Mobile Lab laptops have CanKing installed. An OBD II to serial cable will also be needed if one of the Mobile Lab hybrid vehicles is not used for the experiment. The Leaflight cable should be plugged into the computer before CanKing is launched.



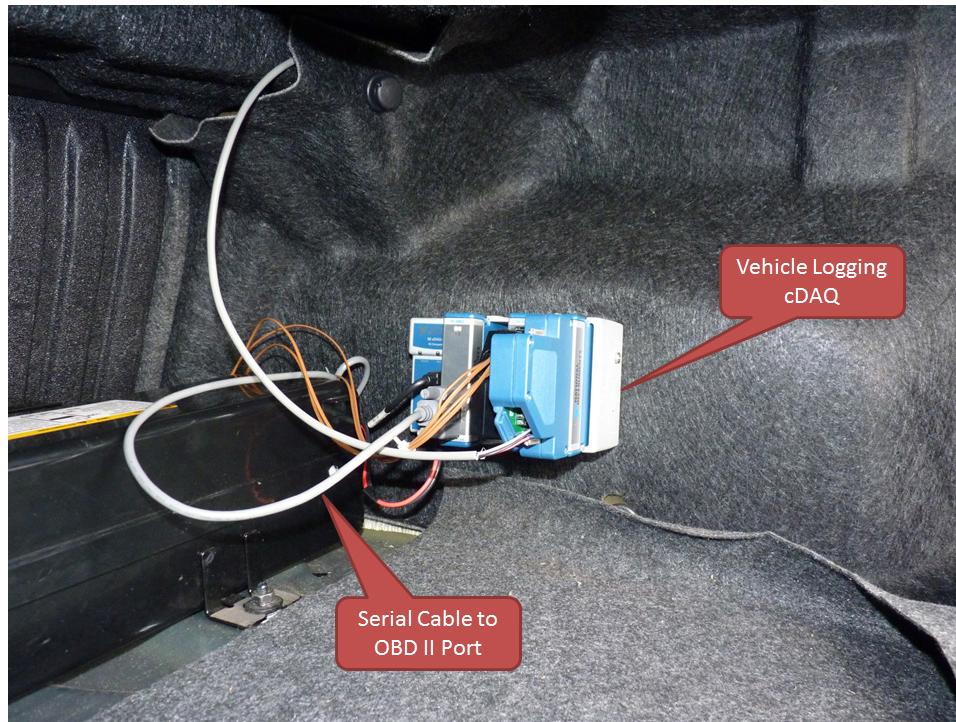


Figure 6.5: Picture of the NI cDAQ mounted in the trunk of the Malibu. The OBD II to serial cable is called out along with the Kvaser Leaflight HS cable.

There is a bit of setup to use CanKing for this experiment. Upon launching CanKing, a window will popup asking to create or open a project. Simply create a new project using a template. This should be the default option. The template that needs to be opened is *CAN Kingdom Basic*. The basic template is composed of four windows and is shown in Figure 6.6. The CAN controller window is used to setup the communication over the bus. The channel should be the Leaflight and the speed should be either 250 or 500 kbaud depending on the vehicle. These parameters can be changed in the *Bus Parameters* tab. Once the vehicle ECU is powered up, communication with the controller can be started by clicking the *Go On Bus* button on the *Bus Statistics* tab on the CAN Controller window. Lines should start populating the Output Window. If no lines or error lines are displayed, then the bus speed needs to be changed to the other speed listed above. Reconnect to the bus using the new settings.

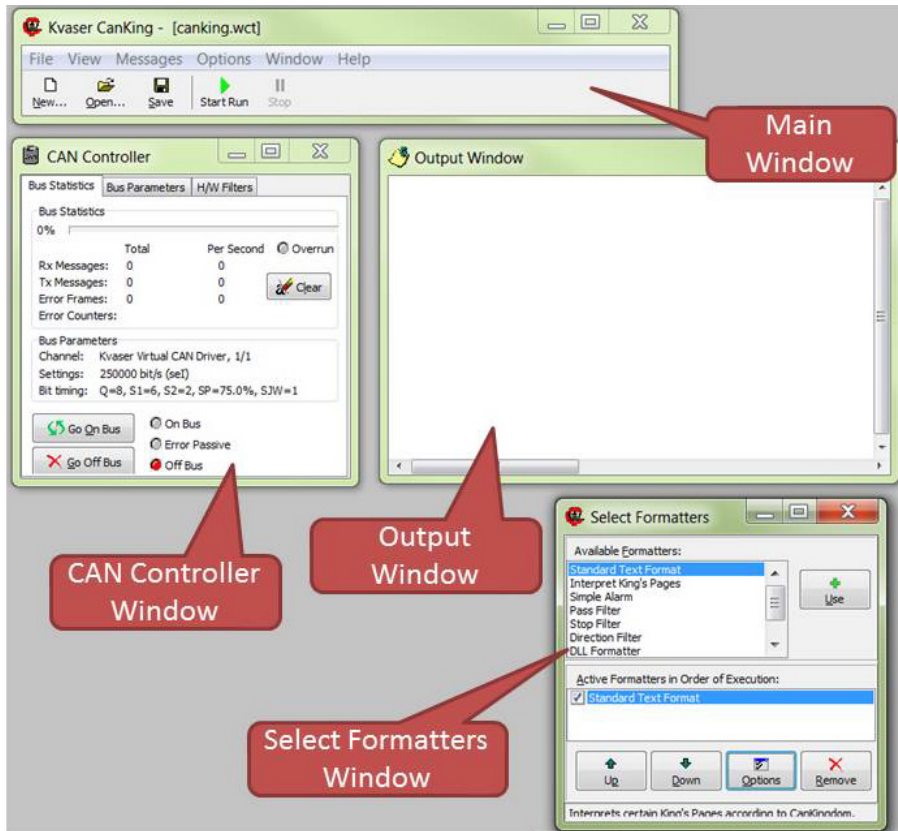


Figure 6.6: Default windows for the basic CanKing template. This image contains content under copyright ownership of Kvaser. Use of this material is covered under their CANlib SDK license agreement found in Appendix C.

It is easier to view the data on the bus in hexadecimal. The default output format is decimal. To change this, left click on the *Standard Text Format* option in the Select Formatters window. Clicking on the *Options* button will then pull up the options window for the standard text format. Change the numeric base to hex to start displaying all of the data in hexadecimal format. In addition, it is also easier to modify the Output Window so that data is displayed in fixed positions instead of a newline every time. This can be done by right-clicking on the Output Window and selecting *Fixed Positions* from the popup menu.

Another window needs to be opened in order to transmit data over the bus. This window can be opened from the *Messages > Universal > Universal* menu of the Main window or the keyboard shortcut of 'Ctrl + U'. The CAN Message window is shown in Figure 6.7 and can be used to transmit the messages required for the experiment.

The DLC must be populated with at least the number of critical bits being sent out otherwise data will be left out of the message.

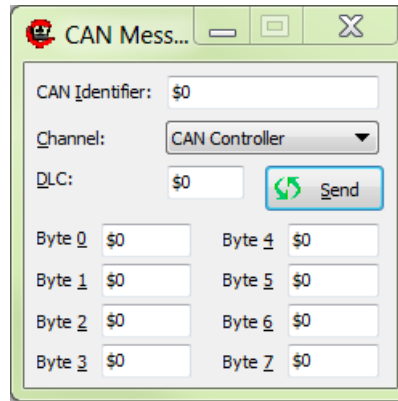


Figure 6.7: Message window for CanKing. This image contains content under copyright ownership of Kvaser. Use of this material is covered under their CANlib SDK license agreement found in Appendix C.

During the experiment, several parameters are requested from the ECU using the CanKing message window with the engine off and on. A list of PIDs and formulas for scaling the data can be found at [https://en.wikipedia.org/wiki/OBD-II\\_PIDs#Mode\\_01](https://en.wikipedia.org/wiki/OBD-II_PIDs#Mode_01). For a quick reference, Table 6.1 below includes the values that should be entered into the message window for each parameter request. The formula to scale the returned data is also included in the table. The data bytes not included in the table should be left blank or zero.

Table 6.1: Message contents of PID request messages for CAN messaging experiment.

Parameter	CAN ID	DLC	B0	B1	B2	Formula	Units
Manifold Pressure	\$7DF	\$03	\$02	\$01	\$0B	A	kPa (absolute)
Ambient Air Temp	\$7DF	\$03	\$02	\$01	\$46	A-40	°C
Mass Air Flow	\$7DF	\$03	\$02	\$01	\$10	$((A*256)+B)/100$	g/s
Engine Speed	\$7DF	\$03	\$02	\$01	%0C	$((A*256)+B)/4$	RPM



## 6.5 Throttle Body Experiment

The throttle body experiment is conducted in MEEM 5296 with the goal of introducing the students to embedded control and the basic control theory of PID controllers. This lab consist of a pre-lab assignment that requires the students to build a MATLAB Simulink model of the controller. A good portion of the lab should be dedicated to discussing this model since it is a critical part of the lab and it is important that the students understand the model before implementing the control on the CHEV. The students then calibrate a PID controller on the CHEV to practice tuning the gains experimentally.

Setup for this experiment should take less than 30 minutes to complete. The only items needed are the CHEV laptop with a Mototron license dongle, a Kvaser cable, and the Bosch ETB with the pigtail harness. The Bosch throttle body and harness can be found in a box labeled *ETB* located with the CHEV components in the Mobile Lab shop cabinets. The pigtail harness should be connected to the DIN rail breakouts on the CHEV according to Table 6.2 below and the pinout table included in the back of the lab manual for this experiment.

Table 6.2: Connection locations for Bosch ETB pigtail harness on the CHEV. Locations of the ECU pins can be found in the back of the lab manual.

Wire Color	Wire Name	ECU Pin Name
Purple	Motor Negative	H2-
White	Ground	XDRG
Blue	5V Power	XDRP
Orange	Motor Positive	H2+
Green	TPS 1 Signal	AN1
Yellow	TPS 2 Signal	AN2

The Simulink model for the control of the throttle body is 'MEEM5296\_ETB\_Assignment.mdl' and is located in the  $\sim$ \MATLAB Files\ML Experiments\MEEM5296\_ETB\_Assignment directory on the portable drive. This model can either be built in Simulink to generate a new program file or the most recent program file located in the directory C:\ECU Files\Programs on the CHEV computer can be used. Mototune should be used to flash the program file onto the

ECU. The Mototune display is also located in the *ECU Files* directory under the *Displays* folder with the same name as the model. This display can be used to calibrate the sensors as well as the PID gains to conduct the experiment.

## **6.6 MEEM 5296 Final Project**

The final project for MEEM 5296 spans several weeks and consists of many parts. The objective of this project is to enable the students to exercise the knowledge they have gained from the 4296 and 5296 past lectures and labs. The students are split into teams and tasked with the problem of improving upon a baseline vehicle. They do this through a combination of computer simulations and chassis dynamometer testing. This project is done using the CHEV.

### **6.6.1 AVL Cruise Simulation**

The computer simulations are done using an AVL Cruise model of the CHEV. Cruise is installed on the four Mobile Lab workstations as well as the computers in the MEEM. The model and Cruise tutorial are located on the portable drive under the *~\Mobile Lab\Courses\Campus Labs\MEEM 5296\Final Project 2015* directory. The tutorial will give an introduction to Cruise and how to use the model for the project. It is recommended to become familiar with the model and Cruise prior to starting the project.

### **6.6.2 Chassis Dyno Setup**

The dyno testing is done using the portable chassis dyno and the CHEV. The CHEV should be loaded and strapped down to the dyno before their first sessions on the dyno. If the circumstances allow, it is best to leave the CHEV on the dyno for the remainder of the project. The following project will describe the process for getting the CHEV positioned on the dyno rolls and strapped down. Everybody should be trained on the dyno prior to helping and all safety guidelines should be followed.

The CHEV needs to be backed onto the dyno using the aluminum ramps. The ramps should be positioned so that the outer edges of the top ramp lines up with the markings labeled with "CHEV" and secured using cotter pins. After applying the roll brake, the CHEV can be backed up onto the platform and onto the roll. You will need

three people for this task, one person driving, one person guiding, and the last person pushing. The electric motor doesn't have enough power to get the CHEV up the ramps so a little extra help is needed. For safety reasons, do not get a running start and 'gun' the CHEV up the ramps.

To strap the CHEV to the dyno, follow the guidelines of having four straps pulling the CHEV back and two pulling it forward. The first set of straps pulling the CHEV back should be crossed and connected to the drop downs of the rear swing arm using the chassis hooks back to the D-rings at the rear of the platform. The other set of straps pulling the CHEV back should be wrapped around the frame rails by the front tires and connected to the slide rails on the platform. The set of straps pulling the CHEV forward should be wrapped around the axle and connected to the forward D-rings on the platform. Leave the straps a little loose initially and run the CHEV in the forward direction with the roll brake released and the dyno power off. This will center the CHEV on the roll. Be sure to tighten the straps before conducting any tests.

### **6.6.3 DynoMax Setup**

To control the load on the dyno, DynoMax 2000 should be run on the dyno computer. A shortcut for the program can be found on the desktop. Ensure that the servo control cable is connected to the black servo box and that the operation switch is in the computer controlled position. Pictures of the servo control box can be found in Figure 6.8 and Figure 6.9. Additionally, the magnetic pickup should be clamped around a spark plug wire. The rest of the setup is done in software. If additional setting need to be changed beyond the ones outlined below, the DynoMax manual should be consulted in the help menu.



Figure 6.8: Picture of the front of the dyno servo control box. The power and control mode switches are shown.

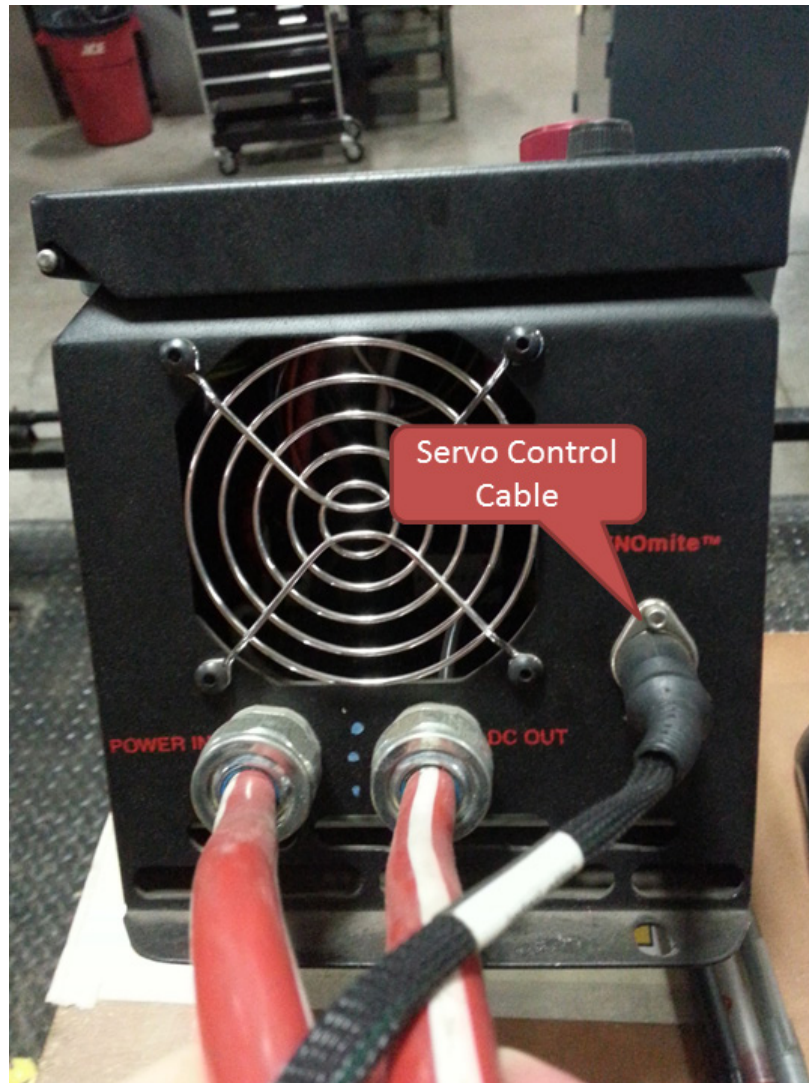


Figure 6.9: Picture of the bottom of the servo control box with the servo control cable plugged in.

The mass and coast down deceleration values in DynoMax have been calibrated to match real-world data collected on the CHEV. These values should be used as the baseline numbers and are shown in Table 6.3 below. These parameters can be edited in the new run window under the vehicle tab. A template has been created for the baseline CHEV that can be imported using the button highlighted in the screenshot shown in Figure 6.10. The baseline template can be found on the computer at *C:\Program Files\DYNO-MAX 2000\CHEV\CHEV\_BASELINE.ifo*.

Table 6.3: Values for mass and coast-down parameter in DynoMax run setup menu for the CHEV.

Parameter	Value
Mass	499 kg
Coast-Down	1.19288 m/s <sup>2</sup> @ 55 mph

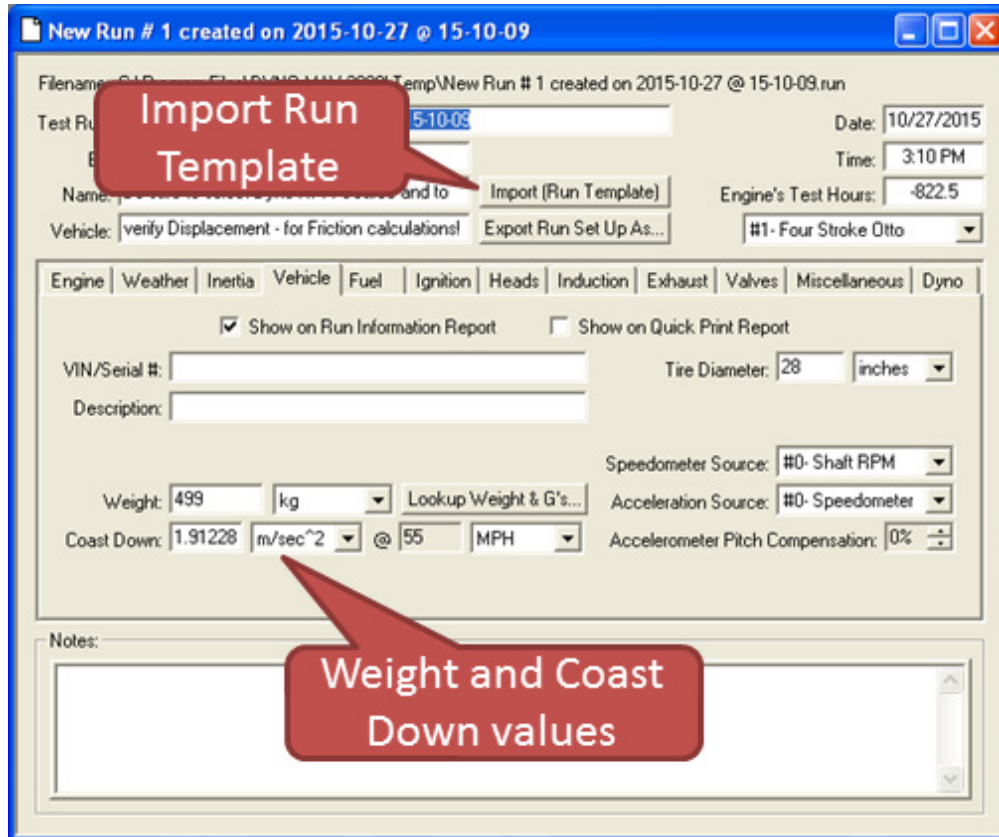


Figure 6.10: Screenshot of the new run window with the vehicle tab shown. This image contains material under copyright ownership of Land & Sea. Permission to use this material can be found in Appendix A.

Once a new run is setup, the drive cycle can be imported using the automated test setup window found under *Edit > Automated Test Setup...* menu or by hitting *F7*. This will bring up the window shown in Figure 6.11. There are several computer controlled tests that can be run, but for the CHEV city and highway cycles the *Clone Run Test* tab should be selected. From here, the city or highway cycle can be cloned using the *Select File to Clone:* button. The currently selected test is shown in the file path next to the button. To change the test profile, simple select the test when the



dialog pops up and hit *OK*. The drive cycles for the CHEV are located at *C:\Program Files\DYNO-MAX 2000\Drive Cycles\* and are named '*CHEV\_city\_cycle.csv*' and '*CHEV\_highway\_cycle.csv*' for the city and highway drive cycles respectively.

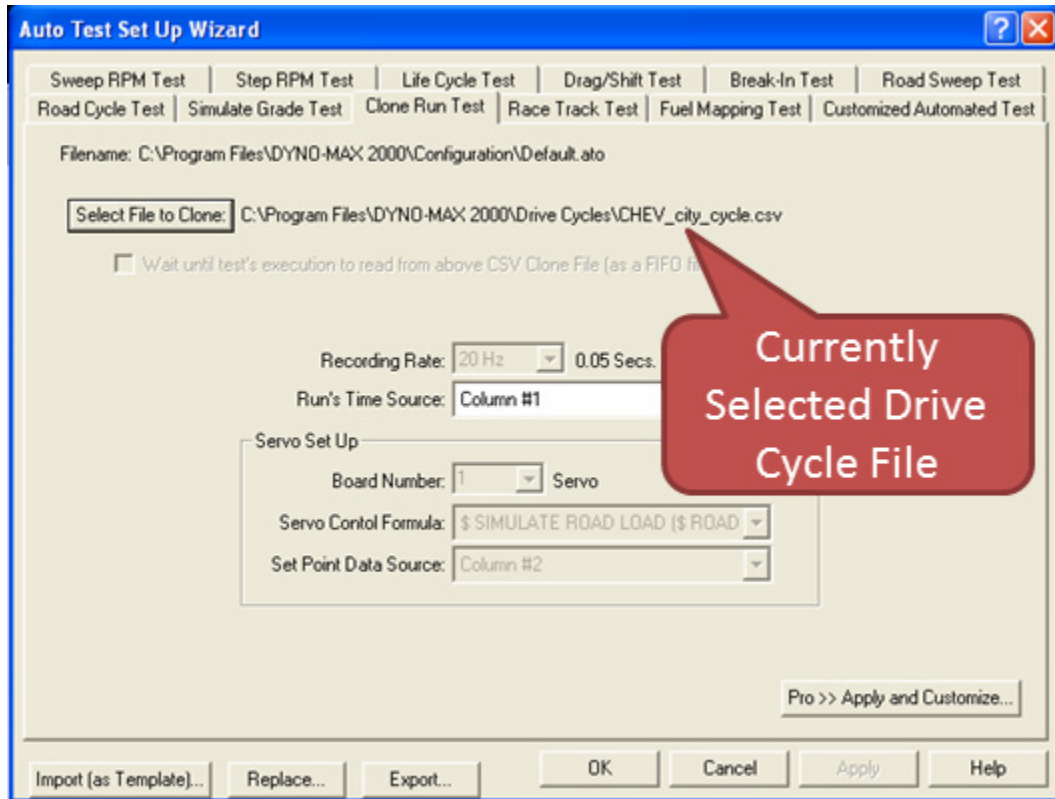


Figure 6.11: Dyno-Max automated test setup wizard for the clone run test profile. This image contains material under copyright ownership of Land & Sea. Permission to use this material can be found in Appendix A.

To run a test, the chassis cell console needs to be opened. This can be done through the *Console > Chassis Cell...* menu or by hitting *F11*. Doing so should open up the window shown in Figure 6.12. A speedometer dial and graph can be found on the console window. These elements display not only the current vehicle speed, but also the upper and lower target speeds for the drive cycle. The driver for the test should, to the best of their ability, try and keep the vehicle speed within these targets. To run an automated test, the console first needs to be put into auto mode by hitting the *Auto* button to turn in green. To start the test, simply hit the *Run Auto Test* button. The upper and lower targets will follow the drive cycle cloned by the

automated test setup wizard. The drive cycles have been designed with a 10-15 second idle period before the speed begins to ramp up.

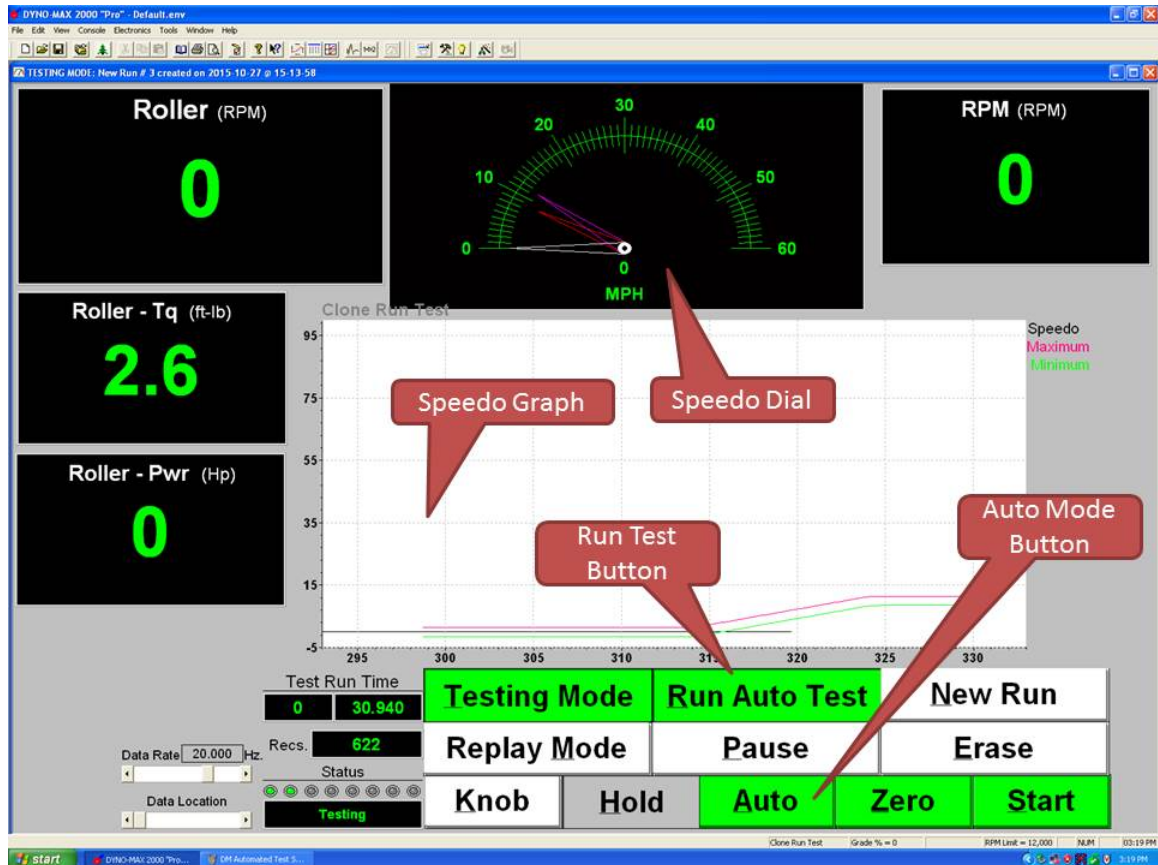


Figure 6.12: Dyno Max chassis cell console window. The current vehicle speed, along with upper and lower speed limits, is shown on the speedo graph and dial. This image contains material under copyright ownership of Land & Sea. Permission to use this material can be found in Appendix A.

#### 6.6.4 Operating the CHEV on the Dyno

The CVT on the CHEV makes it difficult to operate it on the dyno. The dyno cannot be powered on while starting the CHEV from a standstill. The starting torque on the roll causes the belt to slip. The CHEV has no issues starting in the real world so the issue starting on the dyno is unknown at the moment. To resolve this issue, start the vehicle with dyno powered off. Have someone power on the dyno when the vehicle speed reaches 10 MPH. As long as this procedure is repeated for every test all results should be consistent.



Changes to the mass and coast down values should be done on a percentage basis. The values for mass and drag vary between the simulation and dyno so the relative changes between the two needs to be the same. The drag changes are already on a percentage basis so only the mass changes require a calculation to find the percentage change. For example, the mass of the CHEV in the simulation is 575 kg so a 200 kg addition would be a mass increase of 34.8%. The same mass change on the dyno would result in the mass being changed from 499 kg to 672 kg. To make things go quicker during the dyno sessions, it is recommended to have the students do this calculation for you prior to their session.

## **6.7 Electric Machines Seminar Activity**

The electric machines seminar activity is designed to be a short 20 – 30 minute demonstration to highlight some electric machine concepts. The demonstration has several variations, but essentially a speed and load sweep is conducted. This can be done manually or through an automated test. Please refer to the electric machines experiment in section 6.1 on how to setup the test cell for the activity.

Due to the limited amount of time, the discussion following the activity is brief. Some of the topics that should be covered are DC motor power curves, and the relationships between speed, torque, voltage, and current. One easy calculation that can be done is to determine the motor torque/speed constant. This can be done by noting the torque, voltage, and current values on the whiteboard as the test is executed.

This activity could use some improvement. From past seminars, it was hard to get the students engaged in the activity and it always felt rushed. Finding a way to get the participants more engaged is difficult and might require the use of tools outside of the test cell. One such suggestion is using the power electronics module for the NI ELVIS platform. This will allow the participants to see the motor spinning and changing according to their input rather than just a bunch of numbers on a screen. Having the motor in front of them might make them feel more connected to the activity and get them more engaged.

## **6.8 Power Electronics Seminar Activity**

The power electronics seminar activity is designed to be a short 20 – 20 minute demonstration to highlight some power electronics concepts. In its current state, the demonstration utilizes the motor in the test cell with some additional instrumentation. An oscilloscope with a differential voltage probe and a current clamp on the armature and field wires is used to monitor the switching of the power poles. The oscilloscope and probes used are from Dr. Wayne Weaver from the Electrical Engineering Department. The scope from Dr. Weaver has a VGA output that is used to display the scope window on a TV outside of the test cell. A 50' VGA cable located in the Lab can be routed through the pass-through in the bottom of the cabinet along the front classroom wall to connect the TV and oscilloscope together.

After setting up the TV, cable, and oscilloscope with the correct display settings, a speed and load sweep can be run to show how the power electronics control the motor. Please refer to the electric machines experiment in section 6.1 on how to setup the test cell for the activity. Some of the concepts that should be discussed are PWM signals, switching frequency, and inductor current decay. It is not necessary to record any data during this activity as everything can be observed on the oscilloscope display.

This activity could use some improvements as well. For one, finding a way to display the oscilloscope window on the TV screens mounted on the expandable wall will vastly improve the experiment and reduce the amount of time required for setup. It will also make it easier to perform the activity with larger groups. Just like the electric machines activity in section 6.7, it is hard to get the participants engaged in the activity. Part of it is due to the short length but also there is not a lot for them to do. As suggested with the electric machines activity, using the NI ELVIS power electronics board might resolve this issue and get the participants engaged.

## **6.9 Battery Fundamentals Seminar Activity**

The battery fundamentals seminar activity is designed to be a 15 minute or less demonstration to highlight series, parallel, and some battery thermal concepts. As it

stands right now, the activity utilizes a cDAQ with a thermocouple and analog input card, a few AA batteries, and some low resistance ( $1 - 10\Omega$ ) resistors. There isn't a formal setup for this activity, but a couple battery holders and a breadboard can be used for a quick temporary setup. A few presentation slides that cover some background information as well as the activity can be found on the portable drive at the path: *~\Mobile Lab\Courses\Seminars\Ford HEV Seminars\2014\Battery Fundamentals*.

A thermocouple is placed on the battery as a load is connected to it. This load can be varied to draw different currents from the battery cell by interchanging resistors or putting more in parallel. The larger the current draw, the more rapid rate of temperature change. This temperature change can be monitored by using the *Battery Fund Exp v2.vi* virtual instrument. The VI can be found on the portable drive in the same directory as the presentation slides above. The front panel of the VI can be seen in Figure 6.13. The cell temperature chart includes a red line that indicates the original cell temperature when the VI was first started to help visualize the temperature rise. There are also charts for the cell voltage and current so that these can be monitored as the load changes. The voltage is measured across the load and the current is calculated based on the voltage drop and the given resistance. It is important to update the resistance control under the stop button when it is changed to ensure the correct current value is displayed. Additionally, the results of adding cells in series and parallel can be observed through this VI.

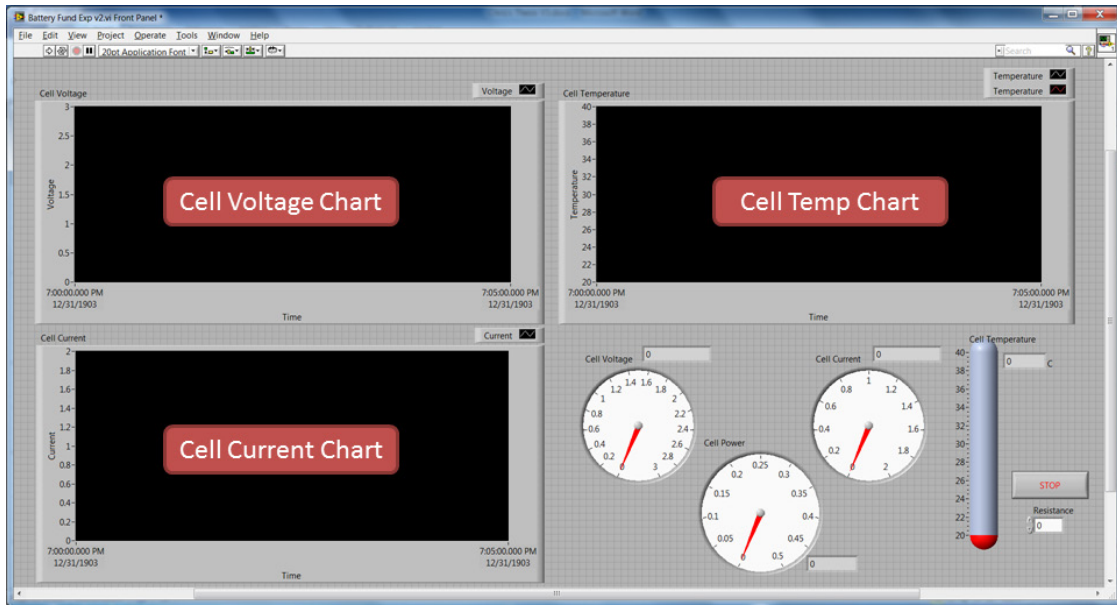


Figure 6.13: Battery fundamental activity VI front panel. This image contains content under copyright ownership of NI. Use of this material is courtesy of NI according to the agreement found in Appendix B.

This activity does very little to engage the participants and is better suited to be a demonstration. If more time is permitted for the activity, possibly having the participants make their own batteries and make observation about them would be better since it would engage the participants. Something like this is already done for the Battery Lab in MEEM 4296. Further details about this can be seen in the lab handout as well as in section 6.2.

## 6.10 Battery Applications Seminar Activity

The battery applications seminar activity is designed to be a 20 minute or less demonstration to highlight some application specific battery concepts. This activity utilizes the 48V battery pack in the front test cell to determine the pulse power capability of the pack. No additional equipment is needed and the battery experiment setup in section 6.2 should be followed to setup the test cell for this activity. A presentation that covers some of the theory and calculations needed to calculate the pulse power capability can be found on the portable drive at the path: *~\Mobile Lab\Courses\Seminars\Ford HEV Seminars\2014\Battery Applications.*

Even though this activity has the participants conducting a few calculations, more could be done to get them engaged and lengthen the activity. It would also be nice to delve into the different chemistries and thermal management. It is unclear how to design an activity to cover these topics and get the participants more engaged.

### **6.11 Vehicle Dynamics Seminar Activity**

The vehicle dynamics seminar activity is a longer 45 – 60 minute activity where the participants determine the center of gravity location of a vehicle. The best vehicle to use for this is the Chevy Tahoe because part of the experiment calls for the movement of some ballast weight and the large cargo space of the Tahoe makes this easier. The point of this activity is to have the participants go through the exercise of measure the corner weights of the Tahoe and calculate the position of its center of gravity. This activity requires a bit of space outside of the lab as well as some additional equipment. This space should be as level as possible. To conduct this activity, the vehicle scales and computer, scale ramps, measuring tape, two sturdy risers of approximately 10”, and a bottle jack and some 2x4’s are needed. If it is desired to show the effect of component placement, then approximately 260 pounds of ballast weight and a wooden crate or something sturdy that is approximately 3’ tall is needed as well. The heavy duty truck ramps work well for the risers.

This activity requires a bit of manual labor and time to complete. First, the scales and ramps should be setup in the front of each tire and zeroed out. This would be a good time to record the vehicles wheelbase, track width, and tire diameter for the calculations. All of these measurements should be taken from the center of the tire contact patch. Once this is done, the vehicle can then be driven onto the scales and weighed in a level position with the ballast weight in its lowest position. After this is done, the ballast weight should be moved to a raised position and the vehicle corner weights measured again. After this, the front of the vehicle needs to be raised so that the vehicle is weighed on an incline. This is where the bottle jack, scrap wood, and the heavy duty ramps come in. The stroke of the bottle jack is just barely enough to get the ramps and scales underneath the tires so use the wood to fill the gap between the bottle jack and the forward cross member. Once the vehicle is on an

incline, measure the corner weights again with the ballast weight in the raised and lowered positions.

After all of the measurements have been taken, the X-Y-Z location of the CG can be calculated. The level measurements are used to calculate the X and Y locations and the inclined measurements are used to calculate the Z location. A spreadsheet has been created that calculates the CG for this experiment and is shown in Figure 6.14 below. The spreadsheet is located in the *Documents* directory of each workstation as well as on the portable drive in *~\Mobile Lab\Courses\Seminar Files\Ford HEV Seminars\2014\Experiments\Vehicle Dynamics*. The X location is reported as the distance from the rear axle, the Y location is the eccentricity about the centerline of the vehicle and the Z location is the height above the ground. A height change is also calculated for the ballast weight in the raised position.

Level Corner Weights						Dimensions				
LF	1488	676	RF	1440	655	Wheel Base	116	2.95		
LR	1472	669	RR	1385	630	Track Width	65.5	1.66		
Total		2629.55				Tire Radius	15	0.38		
Front Axle		1331		Rear Axle		1299		Incline Height	10.5	0.27
Left Side		1345		Right Side		1284				
Inclined Corner Weights						CG Location				
LF	1452	660	RF	1416	644	Distance from rear axle	b	1.49		
LR	1516	689	RR	1413	642	Eccentricity	e	-0.02		
Total		2635				Height	h	0.75		
Front Axle		1304		Rear Axle		1331		Height shift	h'	0.83
Left Side		1349		Right Side		1286				
Inclined Corner Weights w/ Shifted Ballast										
LF	1448	658	RF	1407	640					
LR	1527	694	RR	1416	644					
Total		2635								
Front Axle		1298		Rear Axle		1338				
Left Side		1352		Right Side		1283				

Figure 6.14: Screenshot of the CG Calculator spreadsheet. The results and grey boxes are in metric units of kilograms and meters while the green boxes are in imperial units of pounds and inches. Only the green boxes should be edited.

The following equations are used in the spreadsheet if it is desired to calculate the CG location by hand. The CG distance from the rear axle can be determined by

$$b = L * \frac{W_f}{W} \quad (6.1)$$

where  $L$  is the wheelbase of the vehicle,  $W_f$  is the front axle weight, and  $W$  is the total vehicle weight. The eccentricity about the geometrical centerline of the vehicle is found by

$$e = \frac{T}{2} - \left( T * \frac{W_L}{W} \right) \quad (6.2)$$

where  $T$  is the track width of the vehicle and  $W_L$  is the left side corner weights. A positive eccentricity value indicates that the CG location is on the driver side of the centerline. The height can then be calculated by

$$h = \left( b - L * \frac{W_{fr}}{W} \right) * \cot \theta + r \quad (6.3)$$

where  $W_{fr}$  is the front axle weight with the vehicle on an incline,  $r$  is the tire radius, and  $\theta$  is the inclination angle which can be calculated by

$$\theta = \sin^{-1} \frac{H}{L} \quad (6.4)$$

where  $H$  is the height that the front axle was raised.

## 7 Supporting Systems

This section will cover some of the support systems of the Mobile Lab. In particular, the router, CCTV, and wireless HDMI systems will be discussed.

### 7.1 Wireless Router and Network

The network on the Mobile Lab uses a Cradlepoint MBR1400 wireless router and several wired routers to connect all of the Ethernet devices together. A network diagram is shown in Figure 7.1. The devices on the network are each workstation computer, the PXI systems for both the test cells and the microgrid, the CCTV system, and the combustion cDAQs.



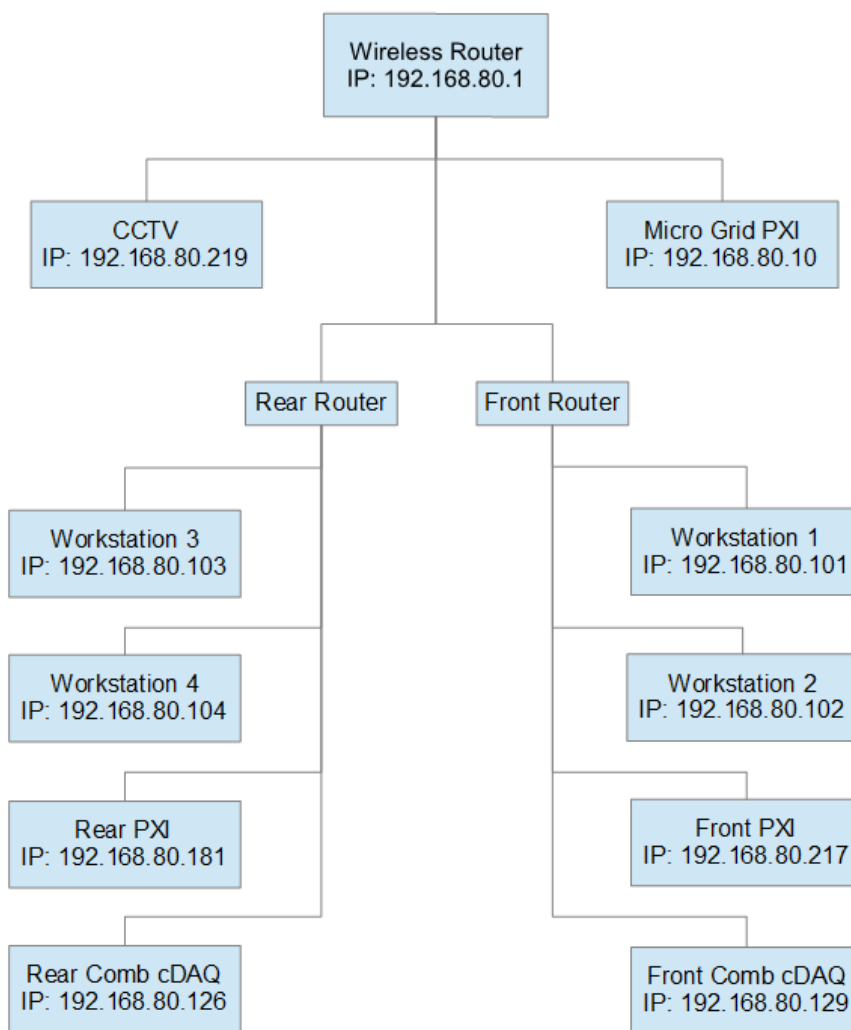


Figure 7.1: Network diagram for the Mobile Lab. The IP of each node is indicated on the diagram.

The administration username and password for the router is *admin* and *440fe31d* respectively. The username and password can also be found on a sticker on the bottom of the router. This will grant access to the router setup in the event of a connection issue. The router has been setup so that it automatically connects to the MTU guest wireless network and the default gateway has been changed so that the AT&T 4G modem can be used on the road. These settings can be modified in the *WiFi as WAN* settings page under the *Networking* tab. The default gateway should be *192.168.80.1*.

## 7.2 CCTV System

A CCTV system is used to monitor the test cells during operation. The camera feeds of the system can be monitored over the local network by typing in the IP address *192.168.80.219* into the address bar of Internet Explorer. There is also a desktop application that is installed on the front computer. A logon screen will show up first, there is no password so simply press login to monitor the camera feeds. Control of the cameras and feeds is then handled through a web interface.

If there is ever an issue connecting with the system, connect a monitor directly to the camera box mounted above the overheads above the third workstation. The menu then can be used to configure the IP address and diagnose any connection issues. It should also be noted that all of the cameras appear in the feed directly from the control box but not through the web interface as seen in **Error! Reference source not found.** where only 2 feeds are seen. This could possibly be fixed through the camera menu.

## 7.3 Wireless HDMI System

The Lab utilizes two Warpia wireless HDMI transmitters and receivers to broadcast images onto the TV screens mounted onto the expandable wall. The four workstation PCs are connected to a USB switch located on the wall above the handicapped side door. One of the transmitters can be used with the switch to connect the TV's to the workstation PC's. The PC's are connected to the switch so that workstation 4 corresponds to connection *A* on the switch and workstation 1 corresponds to connection *D*. The other USB transmitter can be used with a presenter's laptop or to display a different screen on each TV.

The two receivers are mounted on the expandable wall under the whiteboard. A HDMI splitter is also mounted in the same vicinity. The splitter can be used to display the feed from the wireless HDMI receiver on both TV's simultaneously. Or the two receivers can be connected to the individual TV's to display different images.

Due to issues transmitting some dynamic images and websites as well as video, it is recommended that a presenter's laptop be connected directly to the HDMI splitter via a HDMI cable. In addition, it is sometimes required to link the transmitter and receiver dongles by plugging them into the same computer until the Warpia software indicates that they have been linked. This will sometimes solve the issue of not being able to find the receiver on a new computer.

## 8 Summary

The Mobile Lab is a unique venue for engineering education. The lab fills a gap in continuing education and enables professional engineers seeking a master's degree or additional knowledge to learn through an interactive setting. This thesis has outlined the various hardware and software components within the Mobile Lab.

Procedures for operating the front and rear test cells were outlined as well as how to configure the equipment and software used in the cells. Additionally, setup for the key experiments in both the on-campus and professional development courses was discussed.

## 9 Future Work

There are several improvements that can be made to the lab. The first and most important improvement is integrating the cart system in the front test cell.

Currently, the belt drive system for the engine is broken and operation of the cell cannot continue until this is fixed. It will be easiest to just implement the cart system rather than fixing a system that is planned to be replaced.

The second proposed improvement to the lab is upgrading the test cell control software of both cells to Veristand 2014 with the appropriate Inertia plugin. This will comonize the two test cells and make training and operation easier for the team members and course participants. Veristand 2014 is already installed on all of the computers and a deal needs to be finalized with WTI for the Inertia plugin. This update will also fix the issue of not being able to log data in Veristand 2011. The combustion analysis software should be updated at the same time. NI has since changed the name from DCAT to CAS or Combustion Analysis System and has versions for LabVIEW 2013, 2014, 2015. The licensing has since changed from a standalone Driven license to an NI license so there might be some issues there.

Additionally, to better represent the technology in the current market, an AC motor and Li-Ion electric propulsion system should be designed and implemented. There are several off-the-shelve AC motors and controllers that can be easily implemented. In addition, initial discussions with LG Chem have been started on acquiring a 48V Li-ion battery pack for use in the CHEV and test cell. These items will make our courses more relevant.

Lastly, the AVL Cruise model for the CHEV needs to be fixed. When the model was first created in the 2010 version of Cruise, everything worked. The model was able to be executed with the CHEV in all three operating modes. When IT services updated the license for Cruise, the 2011 version was installed. The model can be executed with the CHEV in electric and engine only modes but crashes on every drive cycle with the vehicle in hybrid mode. The cause of this error is still undetermined and should be resolved before the next 5296 final.

## 10References

- [1] Weaver, W., Anderson, C., Naber, J., Keith, J., Worm, J., Beard, J., ... & Hackney, S. (2011, September). An interdisciplinary program for education and outreach in hybrid & Electric Drive Vehicle Engineering at Michigan Technological University. In *Vehicle Power and Propulsion Conference (VPPC)*, 2011 IEEE (pp. 1-6). IEEE.
- [2] Worm, J. J., Beard, J. E., Weaver, W., & Anderson, C. L. (2012). A Mobile Laboratory as a Venue for Education and Outreach Emphasizing Sustainable Transportation. In *American Society for Engineering Education*. American Society for Engineering Education.
- [3] Arnold D. Kerr and R. Byron Pipes (1988). Why We Need Hands-On Engineering Education. *MRS Bulletin*, 13, pp 23-28. doi:10.1557/S0883769400064381.
- [4] Feisel, L. D. and Rosa, A. J. 2005. The role of the laboratory in undergraduate engineering education. *J. Eng. Education*, 121--130.
- [5] J. Bourne, D. Harris, and F. Mayadas, Online engineering education: Learning anywhere, anytime, *J Educ* 94 (2005), 131–146.
- [6] Webster, T.J., and K.M. Haberstroh. 2002. An interactive, video-teleconferenced, graduate course in biomedical engineering. *Journal of Engineering Education* 91 (2): 159–66.
- [7] Okkema, Timothy. Design and Implementation of a Mobile Test Cell. *Master's Thesis*. 2015
- [8] Heywood, John B. Internal Combustion Engine Fundamentals. McGraw-Hill, 1988, 668-676
- [9] *Kohler Engines: ECH730: Command PRO EFI: Product Detail: Engines*. September 9, 2015. <http://www.kohlerengines.com/onlinecatalog/productDetail.htm?productNumber=Command%20PRO%20EFI%20ECH730>
- [10] *Yanmar L100V*. September 9, 2015. <http://us.yanmar.com/products/industrial-engines/air-cooled/epa-certified/l-v-series/l100v/>

# Appendix A



Christopher Davis <ctdavis@mtu.edu>

---

## Copyright permission

2 messages

---

**Christopher Davis** <ctdavis@mtu.edu>

Mon, Nov 30, 2015 at 2:10 PM

To: support@land-and-sea.com

Hello,

I am submitting a thesis for Michigan Technological University that includes some screenshots of DynoMAX 2000 windows and consoles. I am not sure who to contact about this but I am wondering if I need copyright permission to include these images. They are used for the purposes of informing users what settings to use for some tests and how to enter them.

Thank you for your help,

Chris Davis

APS Labs

Michigan Technological University

---

**Ryan Loiselle** <ryan@land-and-sea.com>

Mon, Nov 30, 2015 at 2:44 PM

To: Christopher Davis <ctdavis@mtu.edu>

Chris,

You are more than welcome to use screen shots of DYNO-MAX 2000 for your thesis. Thanks for asking. Hope you do well!

Best regards,

Ryan Loiselle

Technical Support

***DYNOmite Dynamometer***

by Land & Sea, Inc

[ryan@land-and-sea.com](mailto:ryan@land-and-sea.com)

Phone: [+1-603-226-3966](tel:+1-603-226-3966)

[www.dynomitedynamometer.com](http://www.dynomitedynamometer.com)

[www.facebook.com/DYNOmiteDynamometer/](https://www.facebook.com/DYNOmiteDynamometer/)

---

**From:** Christopher Davis [mailto:[ctdavis@mtu.edu](mailto:ctdavis@mtu.edu)]

**Sent:** Monday, November 30, 2015 2:10 PM

**To:** [support@land-and-sea.com](mailto:support@land-and-sea.com)

**Subject:** Copyright permission

[Quoted text hidden]



# Appendix B



December 15, 2015

Christopher Davis  
51455 N Amygdaloid St  
Calumet Township, MI 49913  
ctdavis@mtu.edu

Re: Permission to Use Certain Copyrighted Materials of National Instruments Corporation ("NI")

Dear Chris:

In response to your request to use certain copyrighted materials of NI, by this letter, NI grants you the limited right to reproduce the screenshots as found in **ATTACHMENT A** for inclusion in your thesis to inform users about settings and how to use the pictured items (hereinafter "PURPOSE").

1. You may use the NI Materials only for the purpose(s) described above (the "Purpose"). Please note that (unless expressly stated above in the Purpose) this is a one-time use right, and any additional uses by you will require the separate prior written authorization of NI.
2. Other than resizing the NI Materials as necessary for the Purpose, you may not revise, modify, or otherwise change the NI Materials in any manner without the prior written authorization of NI.
3. In no event may you use the NI Materials in a defamatory, libelous, or unlawful manner or in any manner that disparages NI.
4. You must retain all copyright and other proprietary notices of NI in the NI Materials. Without limiting the foregoing, you must include in the verification procedure, a copyright notice in prominent type and font indicating: (a) NI's copyright ownership of the NI Materials, and (b) that reproduction and redistribution of the NI Materials is courtesy of NI.

Please note that your use right is non-exclusive and non-transferable. NI reserves all rights and licenses not expressly granted you in this letter. **THE NI MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, AND NO WARRANTIES (EITHER EXPRESS OR IMPLIED) ARE MADE WITH RESPECT TO THE NI MATERIALS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, OR ANY OTHER WARRANTIES THAT MAY ARISE FROM USAGE OF TRADE OR COURSE OF DEALING. YOU AGREE AND ACKNOWLEDGE THAT NI SHALL NOT BE LIABLE FOR ANY CLAIMS RESULTING FROM OR IN CONNECTION WITH ITS USE OF THE NI MATERIALS, AND YOU AGREE TO INDEMNIFY AND HOLD NI HARMLESS FROM ANY CLAIMS AGAINST OR EXPENSES OF NI ARISING OUT OF OR IN CONNECTION WITH ITS USE OF THE NI MATERIALS.**

Sincerely,

  
Pete Smits  
Sr. Intellectual Property Counsel  
National Instruments Corporation

11500 N Mopac Expwy • Austin, TX 78759-3504 USA  
Sales and Service: (512) 683-0100 • Other Calls: (512) 338-9119 • Fax: (512) 683-8411 • [info@ni.com](mailto:info@ni.com)

# **ATTACHMENT A**

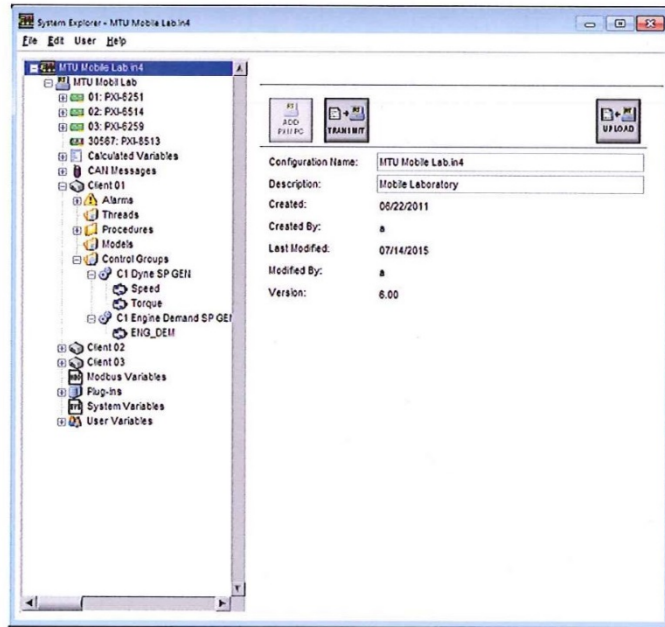
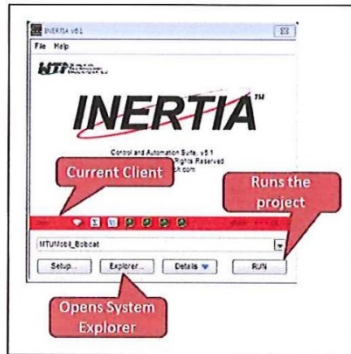


Figure Error! No text of specified style in document..1: Screenshot of the main window of the Inertia system explorer.

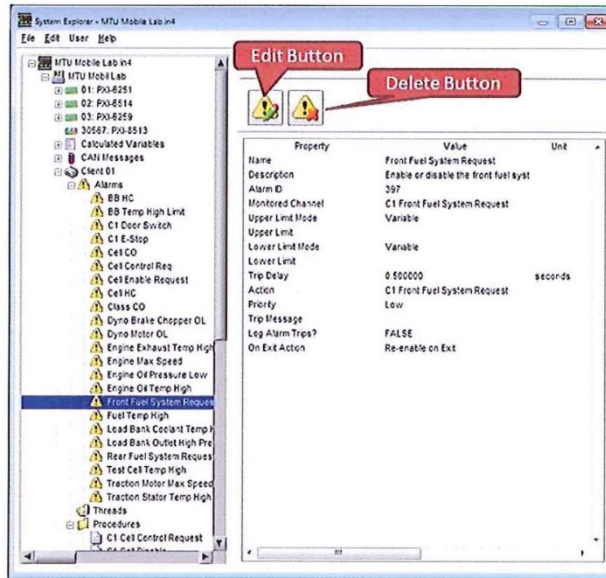


Figure Error! No text of specified style in document..2: Screenshot of Inertia system explorer in the alarm editing menu.

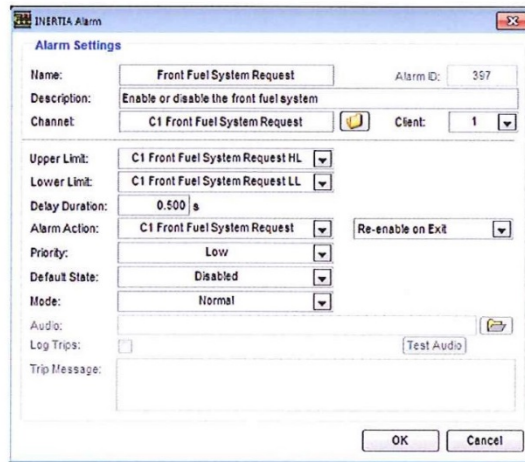


Figure Error! No text of specified style in document..3: Screenshot of the Inertia alarm editing window.

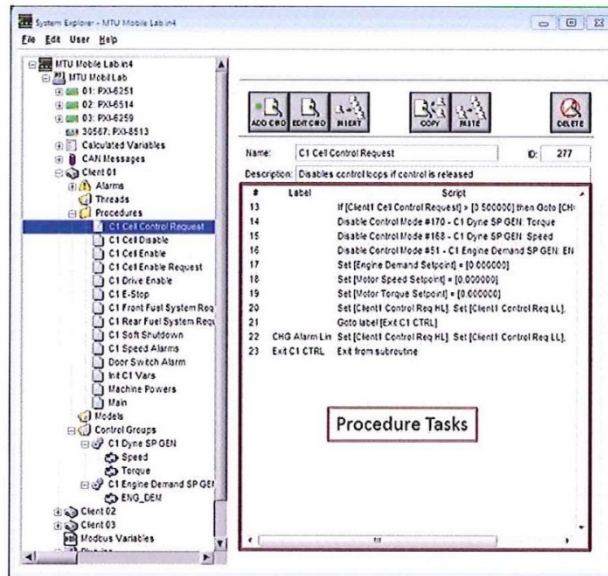


Figure Error! No text of specified style in document..4: Screenshot of the Inertia system explorer window under the procedure branch.



Figure Error! No text of specified style in document..5: Screenshot of the window to add or edit a task of a procedure.

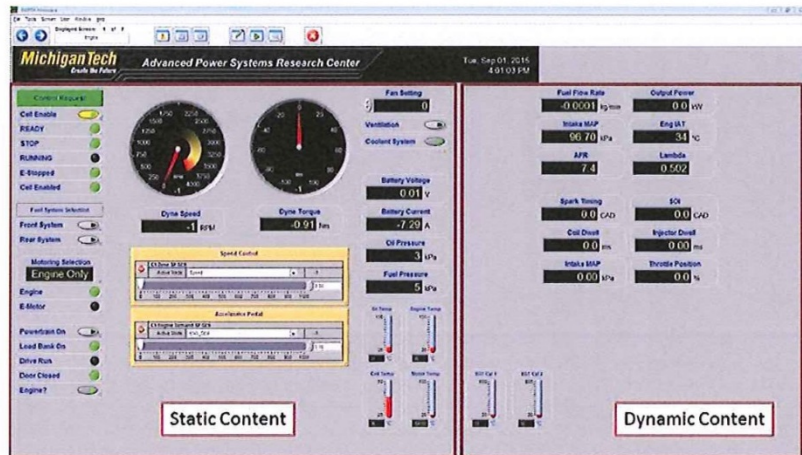


Figure Error! No text of specified style in document.:6: Screenshot of the Inertia 6.1 front panel. The static and dynamic window areas are highlighted.

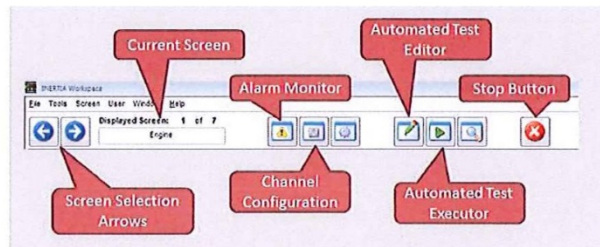


Figure Error! No text of specified style in document.:7: Labeled Inertia workspace toolbar buttons.





Figure Error! No text of specified style in document.8: Screenshot of the manual data logging control and editor windows.

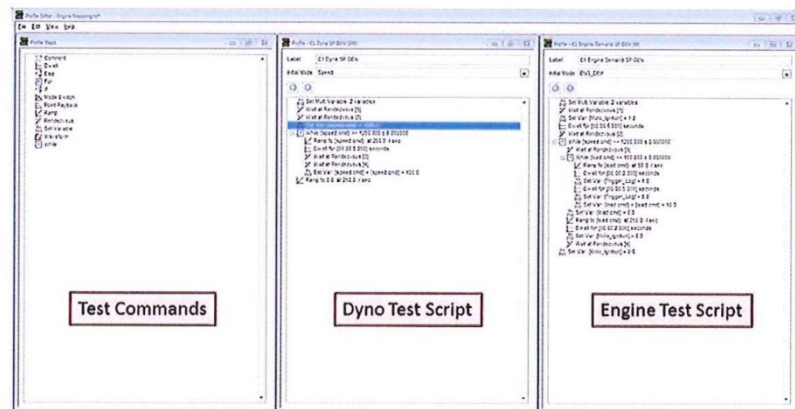


Figure Error! No text of specified style in document.9: Image of the Inertia automated test editor utility window. The test shown is to

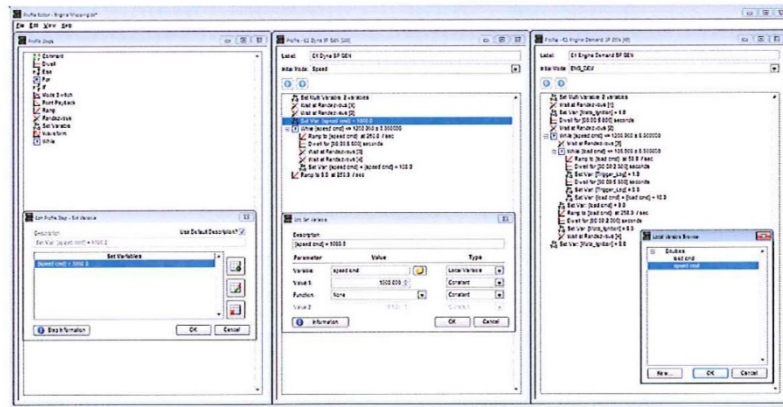


Figure Error! No text of specified style in document..10: Screenshot of the popup windows used to create a local variable.

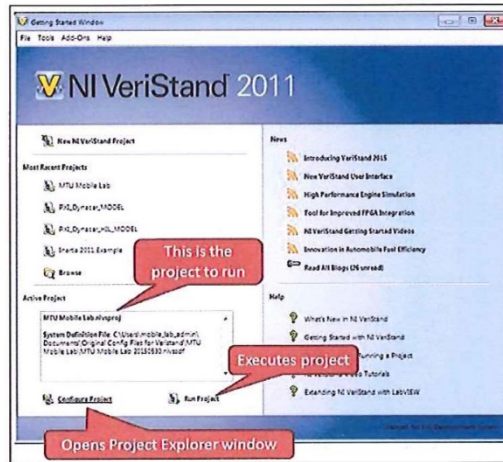


Figure Error! No text of specified style in document..11: Screenshot of the Veristand launch window.



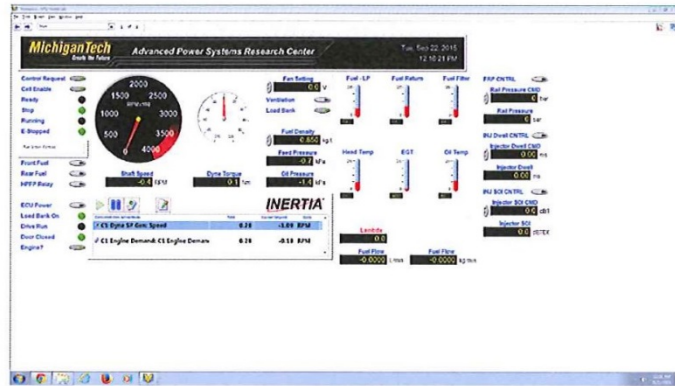


Figure Error! No text of specified style in document..12: Screenshot of Veristand workspace. From here, the test cell can be monitored and controlled.

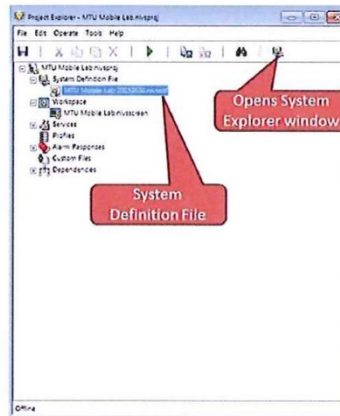


Figure Error! No text of specified style in document..13: Screenshot of the Veristand Project Explorer window.



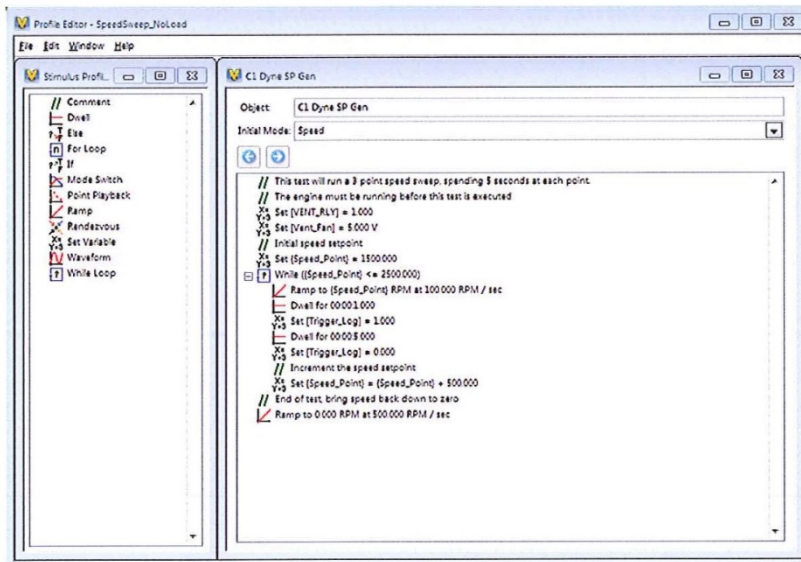


Figure Error! No text of specified style in document..15: Screenshot of the Veristand automated test profile editor. This tool is a part of the Inertia plugin.

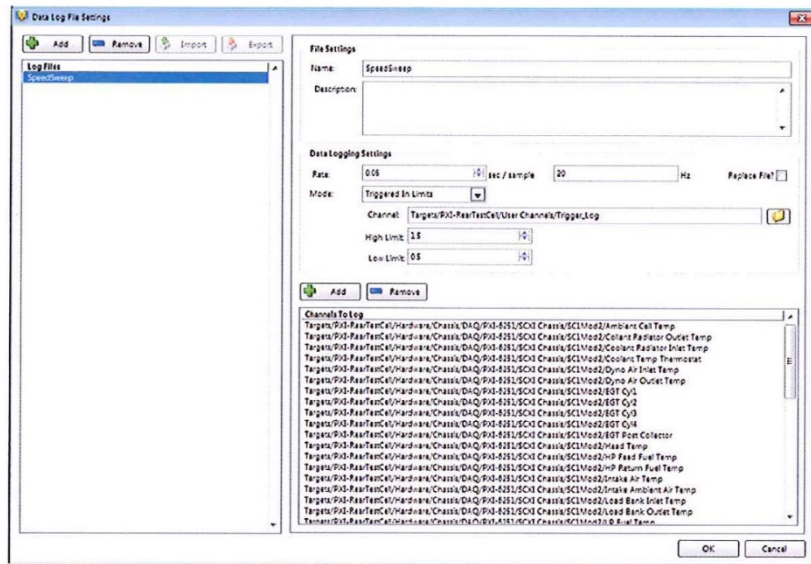


Figure Error! No text of specified style in document.:16: Data log file settings window for the Veristand automated test function.

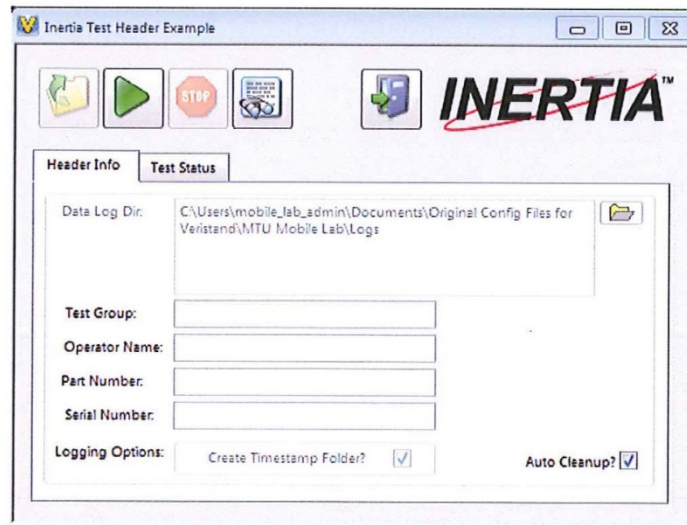


Figure Error! No text of specified style in document..17: Screenshot of the Inertia automated test execution tool in Veristand.

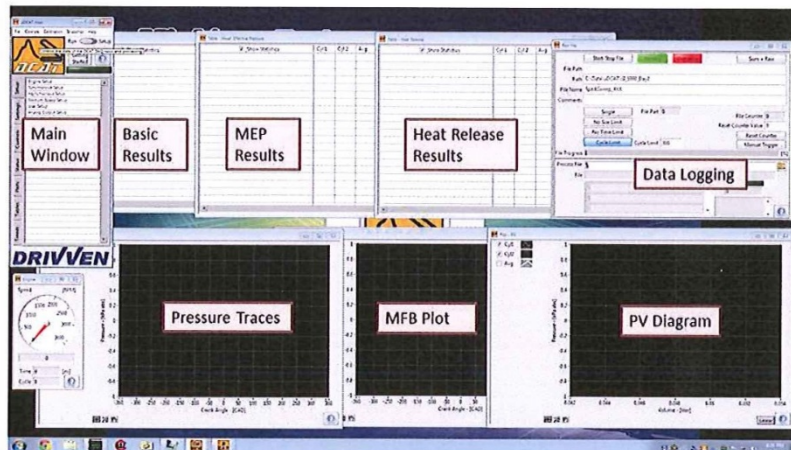


Figure Error! No text of specified style in document..18: Screenshot of the  $\mu$ DCAT workspace.

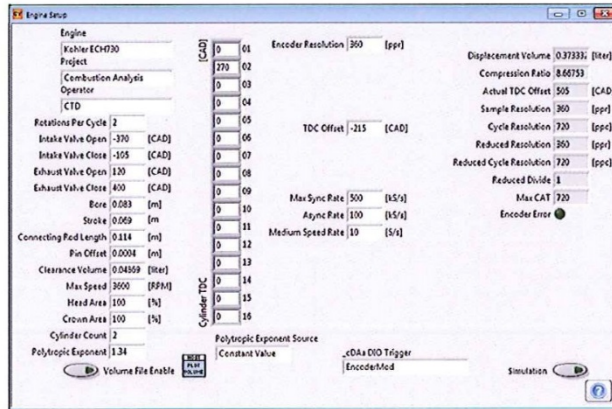


Figure Error! No text of specified style in document..19: Engine setup menu.

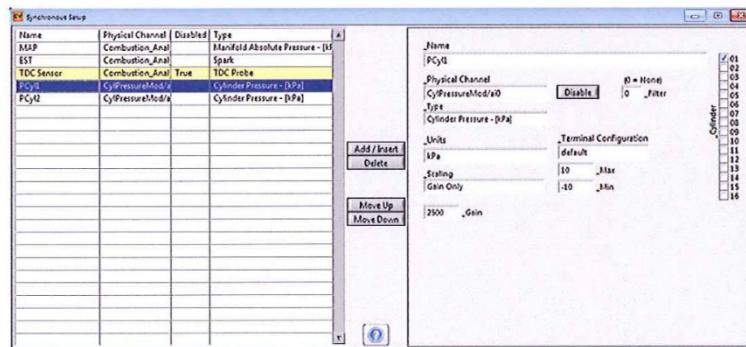


Figure Error! No text of specified style in document..20: Synchronous setup menu.



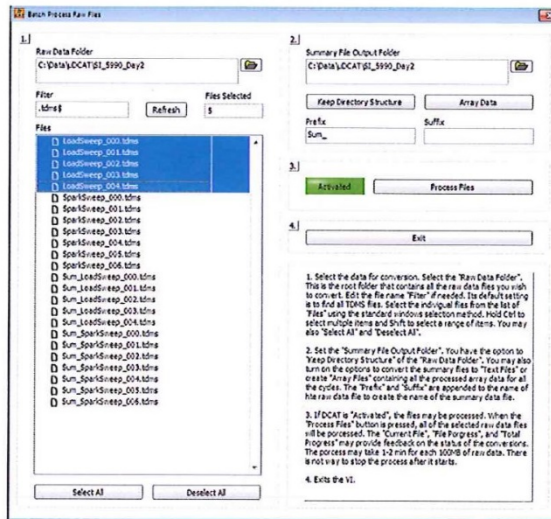


Figure Error! No text of specified style in document..21: Screenshot of the Batch Process Raw Files program. Instructions are in the text box. A prefix of *Sum\_* should be added to help identify the output files as summary files.

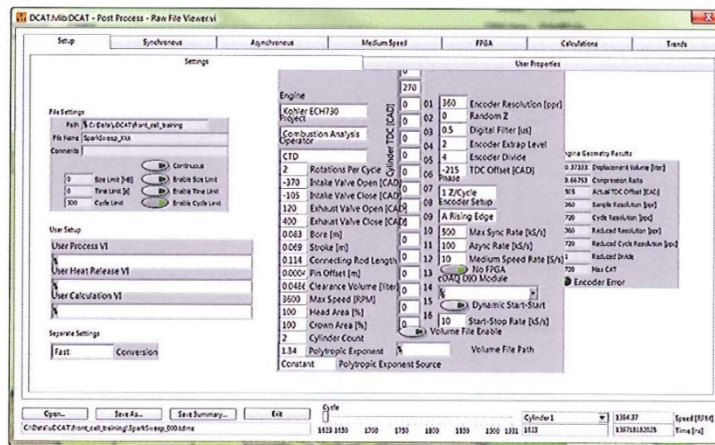


Figure Error! No text of specified style in document..22: Screenshot of the Raw File Viewer program.



Figure Error! No text of specified style in document..23: Virtual instrument front panel for logging data on the HE-HMMWV.



# Appendix C

## License and Copyright

### Copyright

Software Product: canlib 5.12

#### KVASER SOFTWARE LICENSE STATEMENT AND LIMITED WARRANTY

=====

##### IMPORTANT - READ CAREFULLY

This license statement and limited warranty constitutes a legal agreement ("License Agreement") between you (either as an individual or a single entity) and KVASER AB, Mölndal, Sweden, ("KVASER") for the software product ("Software") identified above, including any software, media, and accompanying on-line or printed documentation.

BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE, YOU AGREE TO BE BOUND BY ALL OF THE TERMS AND CONDITIONS OF THE LICENSE AGREEMENT. If you are the original purchaser of the Software and you do not agree with the terms and conditions of the License Agreement, promptly return the unused Software to the place from which you obtained it for a full refund.

Upon your acceptance of the terms and conditions of the License Agreement, KVASER grants you the right to use the Software in the manner provided below.

This Software is owned by KVASER or its suppliers and is protected by copyright law and international copyright treaty. Therefore, you must treat this Software like any other copyrighted material (e.g., a book), except that you may either make copies of the Software solely for backup or archival purposes or transfer the Software to a computer readable medium, e.g. a hard disk, provided you keep the original solely for backup or archival purposes.

You may transfer the Software and documentation on a permanent basis provided you retain no copies and the recipient agrees to the terms of the License Agreement. Except as provided in the License Agreement, you may not transfer, rent, lease, lend, copy, modify, translate, sublicense, time-share or electronically transmit or receive the Software, media or documentation. You acknowledge that the Software in source code form remains a confidential trade secret of KVASER and/or its suppliers and therefore you agree not to modify the Software or attempt to reverse engineer, decompile, or disassemble the Software, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

If you have purchased an upgrade version of the Software, it

constitutes a single product with the KVASER software that you upgraded. You may use or transfer the upgrade version of the Software only in accordance with the License Agreement.

GENERAL TERMS THAT APPLY TO COMPILED PROGRAMS AND REDISTRIBUTABLES  
You may write and compile (including byte-code compile) your own application programs using the Software, including any libraries and source code included for such purpose with the Software. You may reproduce and distribute, in executable form only, programs which you create using the Software without additional license or fees, subject to all of the conditions in this statement.

KVASER products may include certain files ("Redistributables") intended for distribution by you to the users of programs you create. Redistributables include, for example, those files identified in the accompanying printed or on-line documentation as redistributable files. In any event, the Redistributables for the Software are only those files specifically designated as such by KVASER. From time to time, KVASER may designate other files as Redistributables. You should refer to the documentation, including any "readme" or "deploy" files included with the Software, for additional information.

Subject to all of the conditions in this statement, you may reproduce and distribute exact copies of the Redistributables, provided that such copies are made from the original copy of the Software or the copy transferred to the hard disk, or obtained from KVASER's web site, <http://www.kvaser.com>. Copies of Redistributables may only be distributed with and for the sole purpose of executing application programs permitted under this statement that you have created using the Software. Under no circumstances may any copies of Redistributables be distributed separately. Regardless of any modifications which you make and regardless of how you might compile, link, and/or package your programs, under no circumstances may the libraries (including runtime libraries), code, Redistributables, and/or other files of the Software (including any portions thereof) be used for developing programs by anyone other than you. Only you as the licensed user (or the Named User for your entity) have the right to use the libraries (including runtime libraries), code, Redistributables, or other files of the Software (or any portions thereof) for developing programs created with the Software. In particular, you may not share copies of the Redistributables with other co-developers. You may not reproduce or distribute any KVASER documentation without KVASER's permission.

The license granted in this statement for you to create your own compiled programs and distribute your programs and the Redistributables (if any) is subject to all of the following conditions: (i) all copies of the programs you create must bear a valid copyright notice, either your own or the KVASER copyright notice that appears on the Software; (ii) you may not remove or alter any KVASER copyright, trademark or other proprietary rights notice contained in any portion of KVASER libraries, source code, Redistributables or other files that bear such a notice; (iii) KVASER provides no warranty at all to any person, other than the Limited Warranty provided to the original purchaser of the Software, and you

will remain solely responsible to anyone receiving your programs for support, service, upgrades, or technical or other assistance, and such recipients will have no right to contact KVASER for such services or assistance; (iv) you will indemnify and hold KVASER, its related companies and its suppliers, harmless from and against any claims or liabilities arising out of the use, reproduction or distribution of your programs; (v) your programs must be written using a licensed copy of the Software; (vi) your programs must add primary and substantial functionality, and may not be merely a set or subset of any of the libraries (including runtime libraries), code, Redistributables or other files of the Software; (vii) regardless of any modifications which you make and regardless of how you might compile, link, or package your programs, the libraries (including runtime libraries), code, Redistributables, and/or other files of the Software (including any portions thereof) may not be used in programs created by your end users (i.e., users of your programs) and may not be further redistributed by your end users; and (viii) you may not use KVASER's or any of its suppliers' names, logos, or trademarks to market your programs, except to state that your program was written using the Software.

The Software might include source code, redistributable files, and/or other files provided by a third party vendor (Third Party Software). Since use of Third Party Software might be subject to license restrictions imposed by the third party vendor, you should refer to the on-line documentation (if any) provided with Third Party Software for any license restrictions imposed by the third party vendor. In any event, any license restrictions imposed by a third party vendor are in addition to, not in lieu of, the terms and conditions of the License Agreement.

All KVASER libraries, source code, Redistributables and other files remain KVASER's exclusive property. Regardless of any modifications that you make, you may not distribute any files (particularly KVASER source code and other non-executable files) except those that KVASER has expressly designated as Redistributables. Nothing in the License Agreement permits you to derive the source code of files that KVASER has provided to you in executable form only, or to reproduce, modify, use, or distribute the source code of such files. You are not, of course, restricted from distributing source code or byte code that is entirely your own. Source code which you generate with a KVASER source code generator, such as Kingdom Founder, is considered by KVASER to be your code.

Contact KVASER for the applicable royalties due and other licensing terms for all other uses and/or distribution of the Redistributables.

#### LIMITED WARRANTY

Except with respect to the Redistributables, which are provided "as is," without warranty of any kind, KVASER warrants that the Software, as updated and when properly used, will perform substantially in accordance with the accompanying documentation, and the Software media will be free from defects in materials and workmanship, for a period of ninety (90) days from the date of receipt. Any implied warranties on the Software are limited to ninety (90) days. Some

states/jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

KVASER's and its suppliers' entire liability and your exclusive remedy shall be, at KVASER's option, either (a) return of the price paid, or (b) repair or replacement of the Software that does not meet KVASER's Limited Warranty and which is returned to KVASER with a copy of your receipt. DO NOT RETURN ANY PRODUCT UNTIL YOU HAVE CALLED THE KVASER CUSTOMER SERVICE DEPARTMENT AND OBTAINED A RETURN AUTHORIZATION NUMBER. This Limited Warranty is void if failure of the Software has resulted from accident, abuse, or misapplication. Any replacement Software will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. Neither these remedies nor any product support services offered by KVASER are available without proof of purchase from an authorized source.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KVASER AND ITS SUPPLIERS DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT, WITH REGARD TO THE SOFTWARE, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.

LIMITATION OF LIABILITY TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL KVASER OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE PRODUCT OR THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES, EVEN IF KVASER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, KVASER'S ENTIRE LIABILITY UNDER ANY PROVISION OF THIS LICENSE AGREEMENT SHALL BE LIMITED TO THE AMOUNT ACTUALLY PAID BY YOU FOR THE SOFTWARE PRODUCT; PROVIDED, HOWEVER, IF YOU HAVE ENTERED INTO AN KVASER SUPPORT SERVICES AGREEMENT, KVASER'S ENTIRE LIABILITY REGARDING SUPPORT SERVICES SHALL BE GOVERNED BY THE TERMS OF THAT AGREEMENT. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

#### HIGH RISK ACTIVITIES

The Software is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). KVASER and its suppliers specifically disclaim any express or implied warranty of fitness for High Risk Activities.

#### PATENTS AND TRADEMARKS

KVASER may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this Software. Except as expressly provided in any written license agreement from KVASER, the furnishing of this Software does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

#### GENERAL PROVISIONS

This statement may only be modified in writing signed by you and an authorized officer of KVASER. If any provision of this statement is found void or unenforceable, the remainder will remain valid and enforceable according to its terms. If any remedy provided is determined to have failed for its essential purpose, all limitations of liability and exclusions of damages set forth in the Limited Warranty shall remain in effect.

This statement shall be construed, interpreted and governed by the laws of Sweden. This statement gives you specific legal rights; you may have others which vary from state to state and from country to country. KVASER reserves all rights not specifically granted in this statement.