# AC 2010-1026: DATA ACQUISITION IN A VEHICLE INSTRUMENTATION COURSE

**David McDonald, Lake Superior State University**

David McDonald is a Professor of Electrical Engineering at Lake Superior State University and the ASEE Campus Representative. He also does consulting in the area of test cell development for electric vehicle engineering.

# Data Acquisition Applications in a Vehicle Instrumentation Course


## Abstract

The paper introduces instrumentation and data acquisition instruction in a course on vehicle instrumentation. The goal is to build students' skill set with the technology while nurturing their skills and confidence in the design and implementation of testing processes and procedures.

Data acquisition instruction focuses on applications of MATLAB/Simulink, LabVIEW, and Controller Area Network (CAN) hardware and software. The instructional activities introduce typical industrial applications such as the concept of bench marking while engaging the students in the design of the testing process. It also introduces students to modeling and model validation when evaluating the acquired data from the device under test.

The specific course, EGEE365 Vehicle Instrumentation, was piloted twice as a special topics class, and is now a regular offering within a new Vehicle Systems Option in the Electrical Engineering and Mechanical Engineering Plans of Study. An overview of the course and it's placement within a vehicle system option in electrical and mechanical engineering is outlined as a context for the data acquisition and control laboratory activities. Course instruction presents vehicle data acquisition applications while including discussions on the operation and testing of a generic electric vehicle drive train. An internal combustion vehicle and a vehicle chassis dynamometer are also used in the laboratory experience.

A sample laboratory project and assessment discussion is presented. An assessment data summary is also provided for the previous offering of the course along with the larger setting of engineering professionalism data in electrical and mechanical engineering.


## Introduction

The application of modern instrumentation is important in engineering education to provide students with critical skills for use in research and industry. Providing interesting and motivational learning opportunities in engineering laboratory experiences builds students' enthusiasm while teaching critical skills in modern instrumentation and engineering problem solving. It is relatively easy to provide students with interesting instrumentation activities today by using low cost data acquisition hardware and software, and to explore interesting data acquisition applications while implementing group, project-based instruction. Vehicle instrumentation applications today embrace a large spectrum of applications with the increased emphasis on CAN communications and emerging areas such as the growth in electric vehicle development and vehicle-to-vehicle communications. This technology provides an avenue to teach core concepts and techniques of data acquisition while focusing on modern applications within vehicle engineering including electric vehicle applications.

## Instruction Approach

Project-based learning is effective in improving learning outcomes and increasing students' retention for courses and programs. The use of projects in both lower and upper level courses can increase students' interest and success provided the level of difficulty of the project matches the students' skill level. Active discovery and engineering problem solving techniques, including projects and laboratory experiences, have been shown to increase learning success for the course objectives. Duesing et al[1]. Spinelli, et al[2], reports that the development of a discovery based system laboratory using LabVIEW and MATLAB empowered students to 'discover' some properties in the laboratory before they were discussed in the lecture which can lead to a greater appreciation of the material. Frank, et al[3] indicated that keeping students engaged through a teaching studio, software, and class discussions was preferable to lectures.

Providing students with more open-endedness in their learning experiences causes them to become more actively engaged and to exhibit a higher level of satisfaction with the course as indicated by Pape[4]. Casey et al[5] reported that, while project work was always seen as an integral part of later semesters in the curriculum, the need became evident to apply project-based learning (PBL) earlier, primarily to motivate early-stage students that otherwise failed to recognize the applicability of what they were studying to their future professions.

## Electric Vehicle Instruction

There is tremendous interest in electric vehicles today. Several major automotive manufacturers are developing an electric car for mass production, and the United States is on the eve of mass producing an electric car for the first time in history. US Economic Stimulus funding and similar activities are encouraging and fostering new technical development, and the engineering education community needs to evaluate its role in this process[6].

There are many recently documented examples of electric vehicle applications within engineering education. Three course experiences on instrumentation, electric vehicles and project activities were reported by Rizkalla et al[7,8,9]. A summary of the outcomes from these three experiences are that the students were very satisfied, learned technical content not covered in other courses, and felt that the course helped prepare them for the real world of engineering. From an instructor viewpoint the course(s) relied heavily on industrial cooperation, and included hands-on experiences. The authors also noted that an industrial-based course in a new technical area may require heavy industrial collaboration.

Two interesting LabVIEW-based electric vehicle projects that were reported include the development of a Virtual Hybrid Electric Vehicle Simulator using LabVIEW (Laio et al[10]), and another project, reported by Winstead, et al[11], to convert a stock Toyota Prius to a plug-in hybrid having enhanced electric only range capability. The project used National Instruments LabVIEW software and hardware. One assessment outcome was that the project benefited by having both EE and EET students on the team. Parten, et al[12] reported on a project to convert a GM Equinox into an alternative fueled, hybrid electric vehicle. The project outcomes indicated

that allowing students to participate in project-based helped students in the areas of interfacing, decision making and cooperation.

Efforts by Macomb Community College and Wayne State University include an ATE-NSF project Hybrid Electric Vehicle (HEV) curriculum to develop specialized HEV courses as reported by Yet et al[13]. Additional joint activities as reported by Rathod et al[14] includes courses in Energy Sources and Conversion, Control Systems for Vehicles, Fuel Cell Technology, Hybrid Vehicle Technology, Applied Vehicle Dynamics and Advanced Manufacturing Processes.

The US Government has been proactive in supporting new energy related curriculum and course development. One example is a joint effort by the University of Michigan and General Motors to create a program and laboratory to educate automotive battery engineers[15, 16]. Another aspect of the government funding includes a joint program involving several Indiana based universities to educate and train the work force needed to design, manufacture and maintain advanced electric vehicles and the associated infrastructure.[17]

Using project-based instruction in modern data acquisition and instrumentation tools and processes helps prepare students for today's engineering challenges. Several authors have reported positive experiences from initiating an electric vehicle course or focused project, and many new courses are in the midst of the development process. It is hopeful that project-based learning experiences in the context of electric vehicle development will help to draw out of students a realization that they can make a meaningful contribution to something bigger than themselves.

## Setting

The School of Engineering, Technology and Development degree program offerings include EAC/ABET accredited programs in computer, electrical and mechanical engineering and a TAC/ABET accredited program in Manufacturing Engineering Technology. The interest in developing offerings in the vehicle systems area while maximizing the use of resources by offering courses to multiple majors led to the development of new Vehicle Systems Options for electrical and mechanical engineering[18,19].

The Degree Plans of Study for both Electrical Engineering and Mechanical Engineering have designated options (concentrations) in which students take a prescribed cluster of courses. Their final diploma then designates both the engineering major and the option specialty. Electrical Engineering students can select from a prescribed set of courses that comprise a Digital, Robotics, Mechanical, Vehicle Systems or General Option. All five options require 11 credits. The Vehicle Systems Option is outlined below in Figure 1: Electrical Engineering Vehicle Systems Option. Electrical Engineering majors are not required to take dynamics as part of their core graduation requirements, but it is required for the Vehicle Systems Option.

| Electrical Engineering Vehicle Systems Option | | | |
|---|---|---|---|
| Course Number | Course Title | Credits | Laboratory Experience |
| EGEM 320 | Dynamics | 3 | None |
| EGME 310 | Vehicle Development & Testing | 2 | Software (50%) & Hardware (50%) |
| EGEE 365 | Vehicle Instrumentation | 4 | Software (50%) & Hardware (50%) |
| EGME 415 | Vehicle Dynamics | 2 | None |

Figure 1: Electrical Engineering Vehicle Systems Option

The Vehicle Systems Option for Mechanical Engineering students is outlined below in Figure 2: Mechanical Engineering Vehicle Systems Option. Mechanical Engineering majors who select the Vehicle Systems Option also take EGEE280 Introductory Signal Processing course for elective credits.

| Mechanical Engineering Vehicle Systems Option | | | |
|---|---|---|---|
| Course Number | Course Title | Credits | Laboratory Experience |
| EGME 240 | Assembly Modeling and GD&T | 3 | Software (100%) |
| EGME 310 | Vehicle Development & Testing | 2 | Software (50%) & Hardware (50%) |
| EGEE 365 | Vehicle Instrumentation | 4 | Software (50%) & Hardware (50%) |
| EGME 415 | Vehicle Dynamics | 2 | None |
| EGME 425 | Vibrations & Noise Control | 4 | Software (25%) & Hardware (75%) |

Figure 2: Mechanical Engineering Vehicle Systems Option

This new option includes three, new specialized courses that place a high value on laboratory instruction. The three courses are 1) EGME310 Vehicle development & Testing, 2) EGEE365 Vehicle Instrumentation, and 3) EGME415 Vehicle Dynamics. In addition, some students who have selected a different option will also take one or more of the courses either because of a personal interest or to enhance their engineering skill set in preparation for industry.

## Vehicle Instrumentation Course

The EGEE365 Vehicle Instrumentation course introduces instrumentation hardware and software that support the development, operation, and testing of vehicle systems. The course has evolved through two offerings of a special topics course, and is now a designated electrical engineering course. General topics include vehicle networks, data acquisition and control systems, modeling and simulation, and hardware and sensor interfacing.

The prerequisite courses are EGEE210 Circuit Analysis (DC & AC Circuits), and EGNE265 C-Programming. A two credit course, EGNR140 Numerical Analysis (with MATLAB) is a prerequisite for the C programming course. The students are usually juniors or seniors, and have also completed a course on numerical analysis applications using MATLAB. The electrical circuit analysis course provides students with an electrical background to wire circuits and use test equipment. The C programming background helps prepare students for learning the CAN software which uses C syntax. The programming background also enables covering LabVIEW topics, such as data types and arrays, relatively quickly because students have previous experience with these items.

The course is structured as three, 50-minute classes and a three-hour lab each week. The classes may follow a lecture format to explain new material, but many classes tend to be of a discussion format. Initial laboratory instruction is taught in a traditional electronics laboratory with computers on each student work bench, and a chassis vehicle dynamometer is used for some exercises later in the course. In addition to scheduled laboratory times, students may have after-hours access to the lab with the instructor's permission.

Instructional resources include the book LabVIEW for Everyone 3rd Edition by Jeffrey Travis and Jim Kring (ISBN 0-13-185672-3), and the Programming With CAPL manual for CANoe[20]. The CAPL manual is downloaded from the Vector CANtech website, and reprinted on campus with written permission. An appendix in the CAPL manual is used for instruction on the CAN bus operation. Other handouts and laboratory exercises are instructor generated based on academic and industrial experience.

Graded items include homework, quizzes, exams, laboratory exercises, and project documentation. Students are required to maintain an engineering notebook with an industrial format flavor. Students write formal, business-style memos on major projects in the Organizational Purpose (OP), Technical Task (TT), and Rhetorical Purpose (RP) format. The writing also includes journals with both free-write and assigned topics. Finally, there are some written User Manuals and Testing Procedure assignments to give students an opportunity to experience that type of writing.

## Real-World Topics

The vehicle instrumentation course provides students with a general data acquisition and analysis foundation that will serve the student in future study and in industry. The acquisition and analysis of component data is important because data is required throughout the life expectancy of a product. Test data is essential in the design and validation process, performance evaluation, durability testing, production end-of-line testing, and in-service use of the product. Data can be obtained directly from sensors attached to the device, or, in the case of vehicles, it may be available from existing in-vehicle networks. Therefore, data acquisition from in-vehicle networks is also introduced in the course.

While many students who are interested in vehicle applications have some background in internal combustion engine vehicles, few students have had any exposure to electric vehicle applications so this area is also introduced in the course.

General Data Acquisition Topics

General hardware topics include an overview of sensors, connection methods (single vs differential inputs), ranging, offsets, linearity, isolation, and general signal conditioning. Basic analog amplifiers are reviewed, and digital fundamentals are covered for those students who have not taken digital electronics. Data acquisition topics such as A/D converters, time and amplitude resolution, triggering, averaging, sampling, sample rate, and signal aliasing are discussed. The analysis of data includes an introduction to frequency analysis and filtering techniques.

In-Vehicle Network Topics

Applications of Controller Area Networks (CAN) and other in-vehicle networks are discussed in the course. In-vehicle networks are becoming increasingly more sophisticated in terms of the number of controllers, communication speed, and the types of data signals thus making in-vehicle networks an attractive source of critical data. The course provides a foundation for the types of data acquisition challenges that are involved in the development and testing of vehicle components along with a foundation in CAN. CAN systems are found in cars, heavy-duty vehicles, autonomous vehicles, off-road vehicles, ships, and many other applications.

Course topics include the investigation of typical vehicle networks including acquiring vehicle data via CAN. This includes an introduction to the use of the On Board Diagnostic (OBD-II) CAN interface located in the center dash area at the carpet level in new vehicles. The course also looks at other data acquisition approaches including simultaneously acquiring network data with direct sensor measurements, wireless data acquisition, and GPS-based acquisition techniques. Data acquisition instruction includes traditional personal computer based systems with an introduction to the use of a stand-alone, real-time data logger for real-time data acquisition, analysis, display and storage of data.

Detailed investigation of the CAN network includes network operation, setup of message data base information, CAN message layer structure and setup, CAN network layout and connections, programming message transmission and reception, and programming simulation of network nodes. Post-test analysis includes techniques to search CAN data Log files to find key points, and exporting data to MATLAB for analysis and presentation.

Electric Vehicle Topics

The course introduces discussions of various vehicles including the electric vehicle drive such as the diagram shown below in Figure 3: Electric Vehicle Drive Train. In the diagram, the battery

is a cluster Lithium Ion cells, and supplies 300+ volts DC voltage at a high current to the power electronics stage. The power electronics stage inverts the battery DC voltage into three-phase AC voltage at the proper frequency and voltage for the motor to meet the requested speed and torque. The motor is a high efficiency (90%) three phase ac motor such as an AC Induction Motor or Permanent Magnet Synchronous Motor. The motor can operate in all four quadrants of the torque-speed map and provide both accelerating and braking torque. During braking the motor is operated in a regeneration mode that provides braking torque and causes current to flow back into the battery. The computers of the Vehicle Control Unit, Battery Controller and Motor Controller communicate with each other using Controller Area Networks (CAN) or similar communications systems.

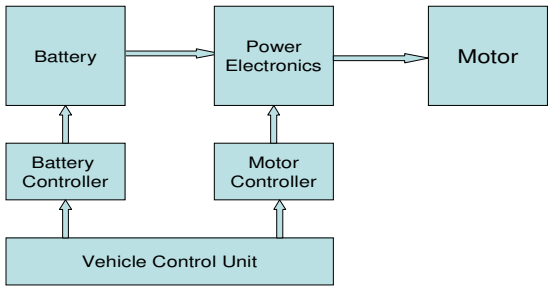


Figure 3: Electric Vehicle Drive Train

Vehicle Component Testing

The course topics include discussions of test cell applications of vehicle component testing including components used in electric vehicle systems. Example discussions include outlining development tests, durability tests, and drive-cycle tests.

## Overview of Software Topics

Laboratory instruction focuses on instrumentation software and hardware of the type that students are likely to encounter to control a testing environment or acquire vehicle test data. This could occur via the vehicle internal CAN network, instrumentation mounted on the vehicle, or software and hardware that is used to control a component under test in a dynamometer test cell or similar test location.

The 'CAN Goes to College' program offered by Vector CANtech[20] has been an extremely helpful source to obtain professional level CAN hardware, software and training with a very attractive academic discount. The program includes professional CANoe software that is used to cover the basics of the CAN bus through classroom and laboratory activities.

Students perform exercises in a laboratory environment that: 1) introduce the operation of the CAN bus and CAN message Data Base, 2) use the CANalyzer bus analysis software to monitor/send/receive/log messages and signal data, 3) use CANoe to simulate a Node and create

a GUI to send and receive messages, and 4) CAPL programming to control message transmission/reception/logging.
National Instruments [21] has been very helpful and offers hardware, software and training with academic discounts. It is entirely possible that graduates will be involved in the setup and operation of controlled testing environments such as engine or vehicle dynamometer test cells. During the classroom and laboratory sessions the students are provided with basic instruction in LabVIEW for data acquisition and control applications. National Instruments data acquisition hardware, including CAN hardware, is used to control test applications and acquire data.

A Vehicle Chassis Dynamometer has been commissioned and installed for instructional use in the Vehicle Option courses. The control application is built in LabVIEW Real Time software, and is a scaled version of a professional level package from Revolutionary Engineering [22]. A National Instruments Single-Board RIO system has been acquired from National Instruments. It has a microcontroller and FPGA and uses LabVIEW Real Time. It will be investigated for use as a data logger system.

The learning experiences also introduce MATLAB and Simulink which are frequently used to analyze data as well as in the Model-Based Design process of creating simulation models and then validation of those models.

**Data Acquisition Instruction: Measurement Automation and Explorer (MAX)**

Measurement Automation and Explorer (MAX) is the first software tool that students are introduced to in the course. MAX is part of LabVIEW, and can be selected for installation when installing LabVIEW software. Even if LabVIEW is not used as part of a data acquisition activity, MAX can be used to ensure the hardware is configured and functioning properly.
The instruction in MAX includes discussions of different types of signals, use of Test Panels, proper wiring and settings of the hardware, and related configuration issues.

MAX is vital for use in data acquisition hardware and measurement setup, and provides the programmer with vital information on the status of the hardware and also shows the related software that is installed. It is helpful to begin laboratory data acquisition topics with instruction on the use of MAX so that when difficulties occur with a programming application the students can verify proper hardware operation using Test Panels and checking pin connections. This enables the students ensure that the hardware is functioning properly before troubleshooting the software. If the data is available in MAX, then the problem must be in the software.

Key information must be initially set up in advance in MAX when LabVIEW is used in CAN applications. Both the CAN hardware settings and the CAN message database information must be structured in MAX before the LabVIEW code will work. CAN *.dbc files can be imported into MAX if that CAN database has been created in another application. If the signals are not imported from an existing database, then that information must be entered manually in MAX when the Channel API programming mode is to be used. CAN hardware settings of transmission speed, type of bus, and related items must also be set up in MAX before the hardware will function properly.

**Data Acquisition: LabVIEW**

LabVIEW is a popular professional programming environment for data acquisition and instrumentation. It is integrated into the course mainly as an environment that lends itself to the control of testing environments such as vehicle component testing.

LabVIEW instrumentation and control software from National Instruments is based on optimizing data flow as opposed to optimizing sequential structure as in tradition programming. The programming uses a graphical format with a Front Panel that represents the user's GUI. The Block Diagram contains the code with the programming performed in a graphical format. Programs in LabVIEW are called Virtual Instruments or VIs.

LabVIEW programs should be coded to follow standard design patterns to organize the code, enhance functionality, and foster efficient troubleshooting. Example standard design patterns include simple loops, Master/Slave loops to separate data acquisition and data processing, State Machines, Event Structures, and other software design patterns. The use of Local and Global is not recommended because data in these variables may be over-written. User-written Functional Global Variables or FIFO Queues are example techniques recommended to ensure data integrity.

CAN communication can be performed in LabVIEW provided proper hardware is installed. Both Frame and Channel API are provided. Channel programming is easier for beginning users, but the Frame API does provide more control of some items such as buffer size. Examples of CAN programming are included in the Example Finder when CAN hardware has been installed.

A simple CAN application from the LabVIEW Example Finder is shown below. The Front Panel is shown in Figure 4: Example CAN Application Front Panel, and corresponding Block Diagram is shown in Figure 5: Example CAN Application Block Diagram.

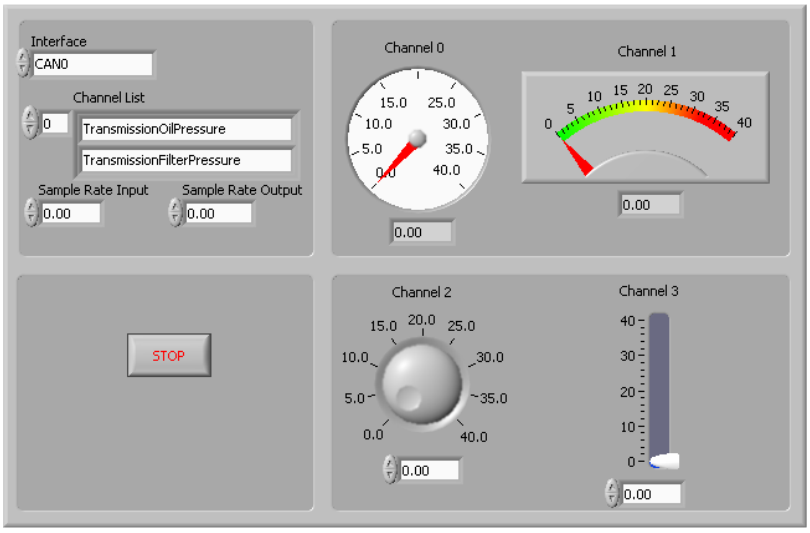


Figure 4: Example CAN Application Front Panel

This example uses a standard loop pattern and CAN Channel API. The program is initialized before starting the loop, and the loop runs continuously with a 1ms wait each loop as set by the timer icon. The loop will only stop if the STOP on the Front Panel is pushed, or if an error occurs during the loop. Once the loop is stopped, the tasks are cleared and resources released.

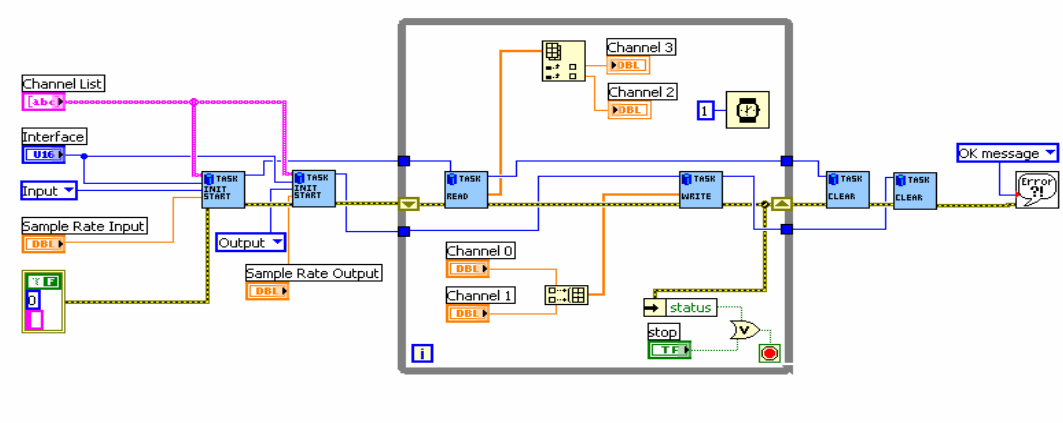


Figure 5: Example CAN Application Block Diagram

Before using this example the corresponding CAN message and signal information would need to be entered in MAX. It is likely that CAN hardware will need to be installed before this example will be visible in the Example Finder.

LabVIEW Real Time

The engineering programs include a well established senior design project sequence. An example project that helps support the new vehicle instrumentation course is the Vehicle Chassis Dynamometer that a senior design team refurbished during 2008-2009. It is now finding use in the laboratory experiences of several courses. The dynamometer is shown below in Figure 6: Senior Design Project: Vehicle Dynamometer.

Figure 6: Senior Design Project: Vehicle Chassis Dynamometer.

The dynamometer instrumentation and control software is a limited version of REPS, a professional, LabVIEW-based, Real Time, dynamometer software package from Revolutionary Engineering[23] The dynamometer control console contains the acquisition and control hardware and a Real-Time industrial PC running LabVIEW Real Time software.

## Controller Area Network (CAN) Software: CANoe

Controller Area Network (CAN) applications harness the power of distributed computing systems in the form of independent micro-computer modules referred to as Engine Control Units (ECU). CAN networks and software are used in the automotive, heavy equipment, ship, and other vehicle industries. CANoe is a development platform for professional network development software that includes CANdb (a message and signal database), CANalyzer (bus analysis tool), and CAPL (C-based bus programming environment).

Students learn to use CANoe to simulate CAN network operation or to obtain data from a CAN network. They can also simulate nodes on an existing network and implement test scripts. A CANoe panel is shown below in Figure 7: CANoe Environment. The software configuration that is demonstrated is a very simple demo configuration, called EASY, that is supplied with the software. There are two messages, LightState and MotorState, being transmitted and received on the bus. The message information can be seen on the far left window under Messages, and on the bottom right Trace window.

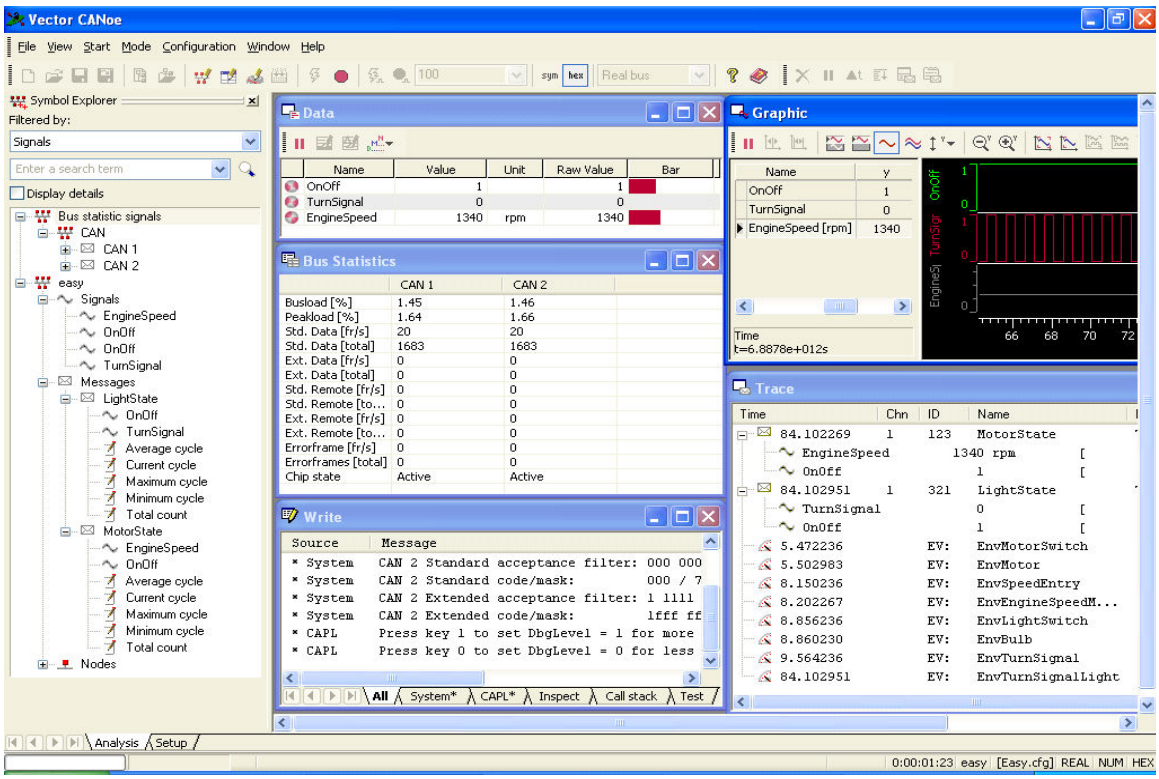


Figure 7: Vector CANoe Environment

The CANdb database for the Easy Confirmation is shown below in Figure 8: Vector CANdb Database. The CANdb contains the message and signal data.

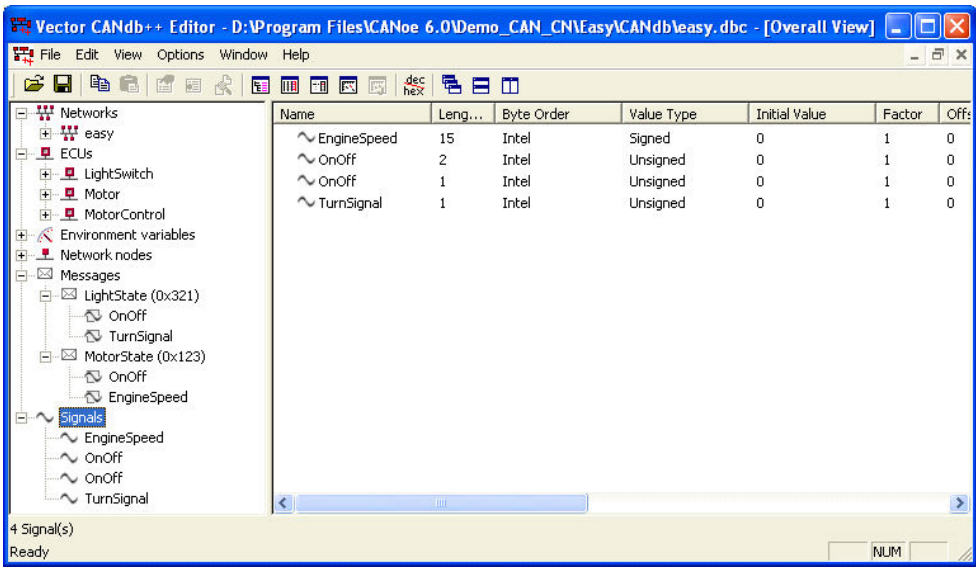


Figure 8: Vector CANdb Database

The bus Simulation Setup is shown below in Figure 9: Vector CANoe Simulation Setup , and shows the simulated ECUs labeled LightSwitch, Motor, and MotorControl. This simulation capability makes it easy to do network analysis and troubleshoot network problems. Students can also display and log bus signal data, insert pass or block filters, or generate test messages.

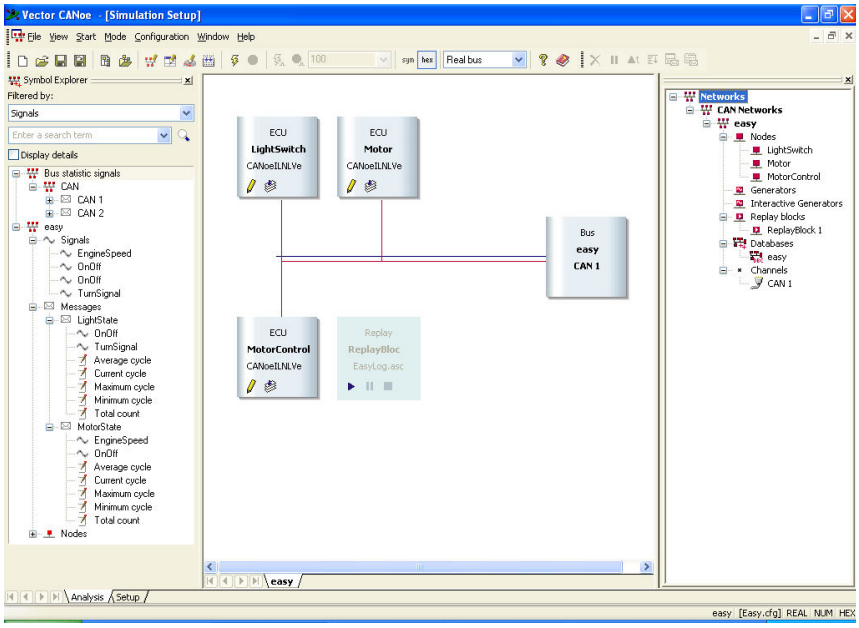


Figure 9: Vector CANoe Simulation Setup

Two GUI Panels are part of this simulation. The Control GUI is shown in Figure 10: Simulation GUI Control, and Display GUI is shown in Figure 11: Simulation GUI Display. The status of the switches and controls on the Control GUI determine the signal values that are transmitted in the bus messages. This signal information is captured by Display GUI. This figure shows the simulated ECUs labeled LightSwitch, Motor, and MotorControl. This simulation capability makes it easy to observe, analyze, and supplement data traffic on the bus.

As part of the CAN instruction students learn how to create the database, setup a bus analysis or simulation session, program node response using CAPL, and program the GUI displays.

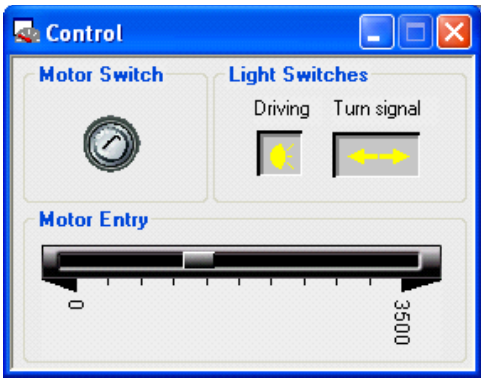


Figure 10: Simulation GUI Control


Figure 11: Simulation GUI Display

CAN Access Programming Language (CAPL) is an Event-based language that allows simulation of a node on the network. CAPL is a C-based language that interfaces with the CAN Database, CANdb, and therefore has access to message and signal information. The CAPL Browser shows the Browser Tree, Global variable declaration, script test, and message window.

Portions of the CAPL Browser for the simulation GUI Control are shown below in Figure 12: CAPL Browser for Control. As mentioned earlier CAPL is an Event-based program. It runs continuously while the simulation is active, and responds to Events such as receiving a specific message or a change in data of a specific signal. In this example the Browser is showing the CAPL code that would execute when message LightState is on the bus.

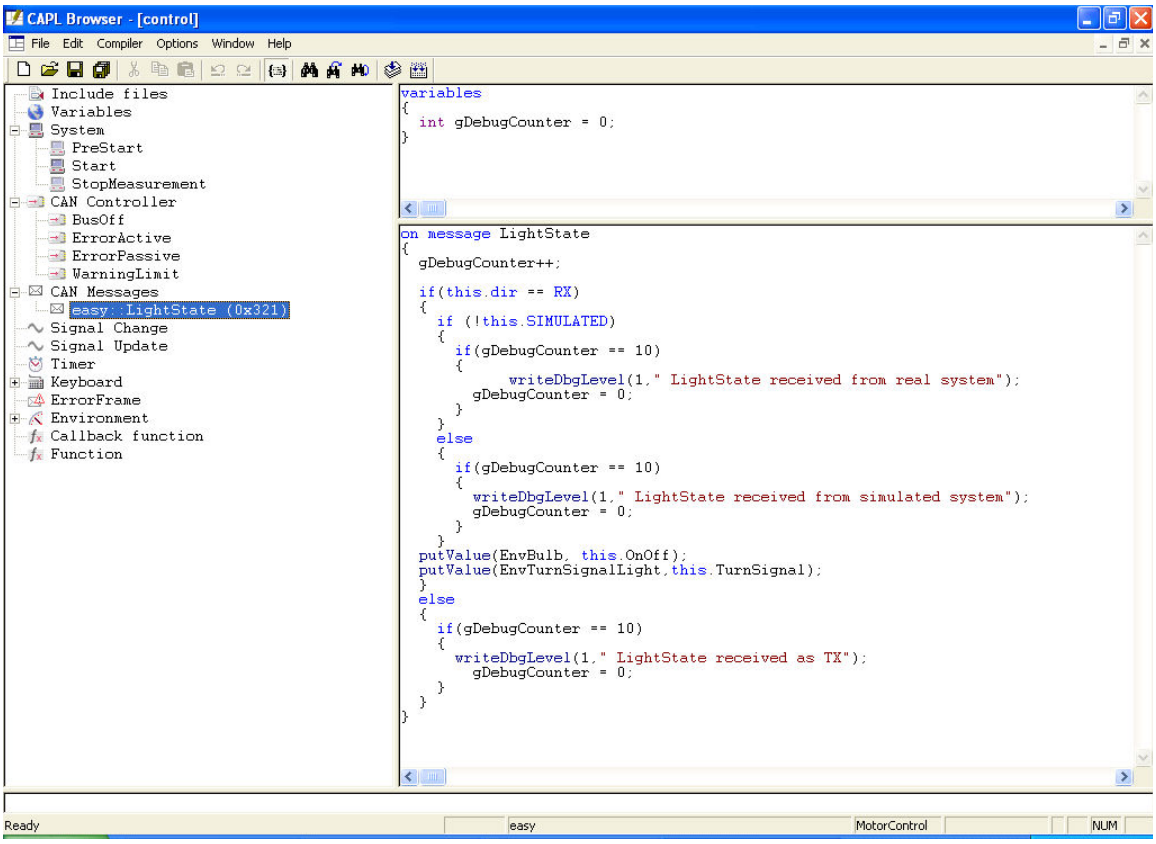

Figure 12: CAPL Browser for Control

An example of CAPL code is shown below in Figure 13: CAPL Simple Example that causes Message 1 to be periodically transmitted.

```
variables
        {
        message 0x555 msg1 = {dlc = 6};
        msTimer timer1;
        }
on start
        {
        setTimer(timer1,100);
        }
on timer timer1
        {
        msg1.byte(0) = msg1.byte(0) + 1;
        output(msg1);
        setTimer(timer1,100);
        }
```

Figure 13: CAPL Simple Example

CAN programming gives students the background to enter the industrial setting and acquire key variable data that is contained in message signals. In a testing environment this software is vital to acquire key Device Under Test (DUT) data such as Motor Temperature, Per Unit Power, Speed, or Per Unit Torque.

With this data acquisition approach the test engineer can obtain critical data from the device being tested. If an embedded sensor is transmitting temperature information via CAN to the bus, then the test engineer can obtain exact internal temperature via CAN message as opposed to mounting external sensors and measuring an external temperature.

## Data Analysis, Device Simulation, and Data Acquisition: MATLAB/Simulink

Mathworks [23] is very supportive of the academic community, and offers significant academic discounts for software and training. MATLAB/Simulink is a powerful tool that is used in industry for data analysis, model simulation, and rapid prototyping and embedded code generation for control module development in vehicle applications. This process can also be followed in education as Mohammadzadeh, A., et al[24], reported on the use of MATLAB/Simulink in the simulation of a vehicle suspension.

### MATLAB/Simulink/SimPowerSystems

Mathworks has developed physical modeling tools based on Simulink. SimPowerSystems, SimMechanical and SimElectronics can be used to develop models of electrical – mechanical components such as modeling a motor, mechanical system, or electrical circuit. A simple filter is shown below in Figure 14: Simulink/SimPowerSystems Model of a Filter. This model was developed using SimPowerSystems. Physical Modeling learning exercises can be helpful to illustrate technical concepts while building students' computing skills, interest, and motivation.[25]

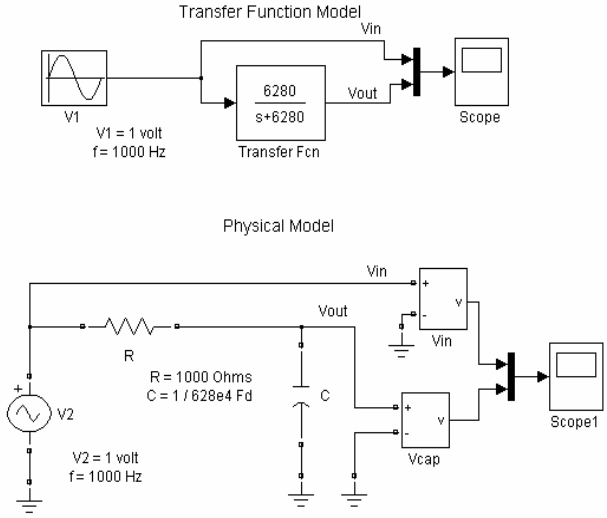


Figure 14: Simulink/SimPowerSystems Model of a Filter.

### MATLAB/Data Acquisition and Instrument Control Toolbox

The MATLAB Data Acquisition and Instrument Control Toolbox[26] enables the user to interface directly to a data acquisition computer board using MATLAB. This empowers the user to

acquire, visualize and analyze real world data all within the same script. Acquisition includes analog and digital I/O, and laboratory instruments such as signal generators and multi-meters.

MATLAB DAQ and Instrument control first defines an object to enable access to hardware functionality which is communicated to the programmer as properties. A typical data acquisition session consists of 1) creating a device object, 2) adding channels or lines, 3) configuring properties to establish the device object behavior, 4) acquiring or outputting data, and 5) usual clean-up to delete the function and clear the workspace. A simple example of analog input data acquisition is illustrated below in Figure 15: Example Analog Input using MATLAB DAQ.

```
%% Cell 1:  Create Analog Device Object w/ analog input channel
AI = analoginput('nidaq',1);          % Create analog input device object
chans = addchannel(AI,1);             % Samples signal connected to channel 1

%% Cell 2: Specify desired sampling parameters and trigger mode
fs = 1000;                            % Set sampling frequency = 1000 samples/sec
Ns = 200;                             % Set number of samples at 200

%% Cell 3: Set up Data Acquisition
set(AI,'SampleRate',fs);              % Set DAQ board acquisition rate
set(AI,'SamplesPerTrigger',Ns);       % Specifies number of samples
set(AI,'Triggertype','Manual');       % Start acquisition on trigger (AI)command
set(AI, 'InputType','SingleEnded');   % Sets DAQ Board amplifier as single input

%% Cell 4: Acquire Data
start(AI);                            % Start the data acquisition
trigger(AI);                          % Software trigger to acquire data
DataVal = getdata(AI);                % Get acquired data & assign to DataVal

%% Cell 5: Plot Data
TimeVal = (0:(Ns-1))*1/fs;            % Create time value data for plotting
plot(TimeVal,DataVal)                 % Plot data
```

Figure 15: Example Analog Input using MATLAB DAQ

MATLAB/Vehicle Network Toolbox

Mathworks has recently added a Vehicle Network Toolbox that supports sending and receiving CAN packets directly from MATLAB or Simulink. It enables encoding, decoding, and filtering of CAN messages, and allows working with industry-standard CAN database files. This supports test and analysis applications in MATLAB that use live data from CAN networks. It supports using live CAN data to validate Simulink models. The toolbox supports the Vector CAN interface hardware CANcaseXL that is used with the CANoe software.

An example application of MATLAB/Simulink in this course would be to develop and validate the model of a component. The component is tested and appropriate CAN data is collected in a CAN log file. This data is post-test analyzed in MATLAB to develop a theoretical model of the component. The component model can then be developed in Simulink to predict it would behave

in a real environment test. The specific component is not important, and the test could be a simple temperature rise test of a small motor.

What is important is that, to the degree possible, the test instruction topics and applications of testing actually reflect industrial testing processes while using actual industrial testing analysis and simulation tools. The inclusion of MATLAB/Simulink for modeling and model validation enhances students' understanding by enabling them to develop analysis and design skills.

**Sample Project**

The course includes testing and benchmarking laboratory activities along with the software instruction. LabVIEW is currently being introduced as part of a laboratory project to measure the temperature profile across the face of a heater. For instructional purposes, a household, hand-held hair dryer is used. The dryer is mounted to hold it fixed, and a temperature probe is slowly moved past the dryer by using a stepper motor and driver controlling a threaded rod. A picture of the general setup is shown below in Figure 16: Dryer Temperature Setup

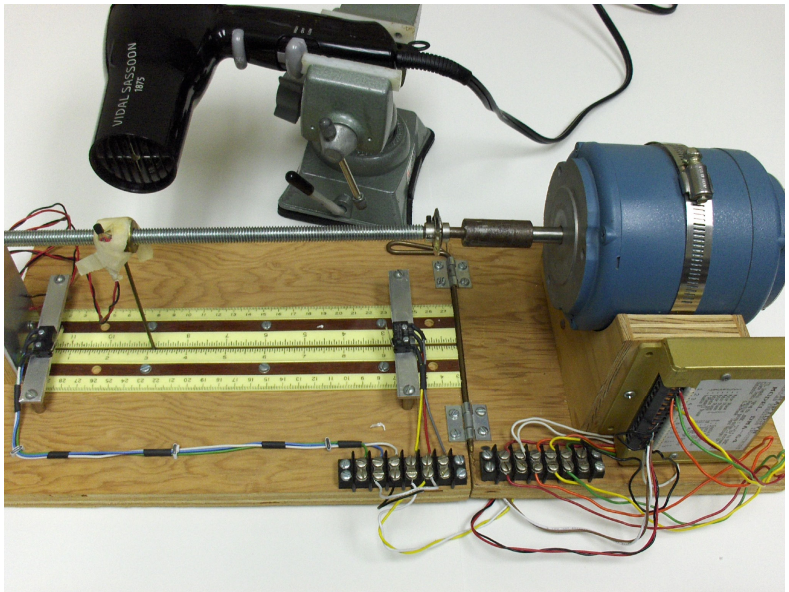


Figure 16: Hair Dryer Temperature Setup

In addition to basic instruction in LabVIEW, the activity includes instruction on the positioning motor, switches, temperature sensors, etc. used in the testing fixture. The students program control the test system in LabVIEW. The LabVIEW software measures the analog voltage of the sensor that is proportional to temperature. The program also includes two digital outputs to control the speed and direction of the positioning motor, and two digital inputs to determine if the end limit switches have been reached. The LabVIEW program uses a standard State Machine ( Case Structure within a While loop ) with a Type Definition for the Case States. The State Machine includes Initialize, Read Limit Switches, Move, Measure Temperature, Wait, and Stop states. The sensor voltage, sensor temperature, and sensor location are logged during the

test. From a Lessons Learned perspective, the laboratory exercise is a challenging but positive experience.

The computers that the students use in this course have a National Instruments E-Series General Purpose DAQ board installed in the computer. Other data acquisition hardware that students encounter in the course is shown below in Figure 17: Data Acquisition Hardware. Clockwise from Top Right:
Measurement Computing Personal Acquisition Device – Analog & Digital I/O
National Instruments PCI CAN Card, 2 Port, High Speed
Vector CANtech CANextender, CAN – Analog & Digital I/O
Vector CANtech CANStart card that ships with Can Goes To College Program
Vector CANcaseXL that ships with the Can Goes To College Program
The small DC motor/tach is also used in the course for some activities.

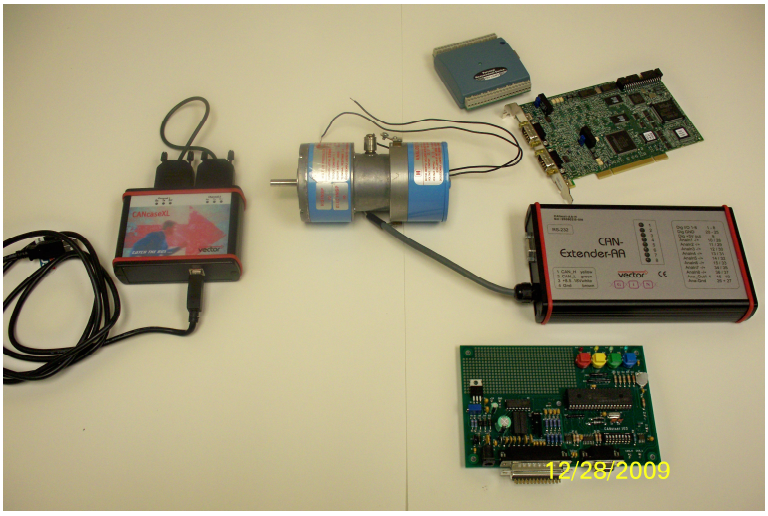

Figure 17: Data Acquisition Hardware

**Assessment**

The School of Engineering, Technology, and Development has an established assessment process that includes program and course quantitative assessment data. (Duesing[27] et al) The assessment process also includes tracking data in Engineering Professionalism, including Design, Software and Communication credits and weighted average grades.

Heater Benchmark Test - Assessment

The lab exercise was initially used as part of a laboratory for EE375 Electronic Circuits, which covered operational amplifiers and general electronic circuit applications. The assessment form included one specific objective for all laboratory activities, and the average student assessment for the representative years was 80%. In that particular laboratory this activity was the final project for the semester. Students were required to use memo form reports and give a design review with a PowerPoint presentation. The students felt the design reviews were a positive step. They also felt the final project was significantly more difficult that previous projects, and recommended having the project build up gradually with culmination in a final project.

A modified form of the exercise was also included in a former course, EE305 Analog and Digital Circuits, which covered analog and digital topics for mechanical engineering majors. The assessment form included one specific objective for all laboratory activities, and the average student assessment value for all laboratory activities for the representative years was 82%.

Course Level Assessment

The Vehicle Instrumentation Course was taught twice as a Special Topics EGEE300 Special Topics, and is now EGEE365 Vehicle Instrumentation. The Assessment Summary from the second offering of the special topics course is shown below in Figure 18: EGEE300 Course Assessment Summary. That specific offering did not include MATLAB activities.

EGEE300 Special Topics Spring 2008
Initial Offering of EGEE365 - Some objectives different

Student Population: ME Seniors and Juniors ( no EE or MfgET students this semester).
Course Prerequisites: C Programming, Signal processing or Analog & Digital Electronics.

General Comments:
This was the second offering of a Special Topics course. This year the course topic distribution was 45% learning CAN software and CAPL programming, 25% learning LabVIEW, and 30% applications of CAN, LabVIEW, or both combined. General student comments indicated that the large number of examples, hands-on environment, and small class size helped to enhance their condifence with the course material on use of the CAN network. The students enjoyed learning the software and anticipated it would prove helpful on a future job.

Plans for Next Offering:
The CAPL programming manual is a free down load from Vector CANtech. Instructor will seek permission to reprint the manual for student use in the course. The LabVIEW for Everyone book worked well for LabVIEW instruction, and cuold be used again. It would be helpful to find a second textbook to support automotive topics.

| Objective # | Brief Description | Faculty Grade | Student Assessment |
|---|---|---|---|
| 1 | Create LabVIEW programs | 92 | 87 |
| 2 | Design patterns & structures | 92 | 76 |
| 3 | Interface Data Acquisition Hwdardware | 92 | 84 |
| 4 | CAN Basics / protocols | 87 | 89 |
| 5 | Analyze CAN networks | 86 | 89 |
| 6 | Simulate CAN networks | 87 | 82 |

Other Data:
Average Grade = 3.0
Design Component = 0.3 Credits
Software Component = 2.2 Credits
Communication Component = 0.4 Credits

Figure 18: EGEE300 Course Assessment Summary Spring 2008

The assessment is review by the department faculty each time the course is taught. The individual forms are helpful, but a more meaningful view can be obtained by looking at trends over time. This type of assessment is provided by the annual formal reports on the appropriate

Program Outcome Objective. All course assessment forms are reviewed by the faculty annually, and the Program Outcome Objective forms are reviewed by the faculty on alternate years.

## Summary

The EGEE365 Vehicle Instrumentation course provides students with learning experiences in data acquisition within the context of component testing and measurement for vehicle applications. The course was developed in response to student and faculty interest as a required course in a new Vehicle Systems Option for both Electrical and Mechanical Engineering majors. The course is also taken by some students as an engineering elective.

The course uses relatively low-cost data acquisition hardware and software that is available at greatly reduced academic pricing. The instructional approach uses group projects that are intended to model industrial practice and introduce students to equipment, processes and procedures that they will encounter in the work place. The instruction includes software programming, hardware interfacing, trouble-shooting, and benchmarking techniques that are consistent with engineers who are entering the workplace.

Course lecture topics include general data acquisition topics, in-vehicle network topics, electric vehicle drive train basics, test cell and in-vehicle data acquisition topics. The course introduces data acquisition hardware interfacing, Measurement Automation and Explorer, LabVIEW basics, LabVIEW design patterns as well as analog, digital, and CAN Bus interfacing. The course also covers CAN network analysis and synthesis using CANoe software environment, obtaining data using CAN signals, and using MATLAB for data analysis and device model validation.

Preliminary assessment results indicate that students enjoy and benefit from the data acquisition and CAN applications in the course, and that the course makes a significant contribution to the professionalism component of design, software, and communications in both the electrical and mechanical engineering programs.

## References


1. Duesing, P., Mokhtar, M. "Active Discovery and Engineering Problem Solving (EPS) Techniques – An Effective Approach to Teach a Freshman Level CAD Course", 2009 ASEE NCSection Conf 2009.
2. Spinelli, J, LaFerriere, K., "A Discovery Based Systems Lab using LabVIEW & MATLAB, ASEE AC
3. Frank, B., Carr, J., "Active Learning Using Guided Projects in an Upper Year ECE Course." ASEE AC2007.
4. Pape, David, "A Progressively Open Ended Laboratory to Promote Active Learning, ASEE Annual Conference, 2006. Rizkalla, M., Pfile, R., El_Antably, A., Yokomoto, C., Development of a Senior Elective for EE and EET Majors in the Design of Electronic Instrumentation for Electric Vehicles, ASEE Annual Conference, 1998.
5. Casey, A., Bratschitsch, E., Millward-Sadler, A., "Thinking Globally, Acting Locally: Strategies for Improving Int'l Experience & Employability Skills of Undergraduate Students of Vehicle Engineering, ASEE AC2008.
6. McDonald, D, Engineering and Technical Education for Electric Vehicle Development, ASEE AC 2010.
7. Rizkalla, M., Pfile, R., El_Antalbly, A., Yokomoto, C., Development of a Senior Elective for EE and EET Majors in the Design of Electronic Instrumentation for Electric Vehicles, ASEE Annual Conference 2000.

8. Rizkalla, M., Yokomoto, C., Pfile, R., Sinha, A., El-Sharkawy, M, Lyshevski, S., Needler, M., Using Senior Research, Design& Development Projects in the Dev of a Course in Electric Veh Tech, ASEE AC2000.
9. Rizkalla, M., Yokomoto, C., Pfile, R., Sinha, A., El-Sharkawy, M, Lyshevski, S., Al-Antably, A., Applications Computer-Based Power Electronics to Electric Vehicle Tech, Interdisciplinary Senior Course, ASEE AC08
10. Liao, G., Yeh C., Sawyer J., Design and Implementation of a Virtual Hybrid Electric Vehicle Simulator for Educational Purpose" ASEE Annual Conference 2008
11. Winstead, V., Applied Engineering with LabVIEW: Experiences from a Plug-In Hybrid project,ASEE AC2007.
12. Parten, M., Maxwell, "Monitoring and Control in Advanced Vehicle Engineering Lab, ASEE AC, 2007.
13. Yet, C., Liao, G., Sawyer, J. "A College-University Partnership for Developing a learning Environment for Hybrid Electric Vehicle Technology" ASEE Annual Conference 2007.
14. Rathod, M., Sheyman, V., Addressing the Alternative Energy Workforce Needs, ASEE Annual Conf 2005.
15. http://energysystemseng.engin.umich.edu/
16. http://engineeringtv.com 2/03/2009 report by Curtis Elizey.
17. http://news.uns.purdue.edu/x/2009b/090805CaruthersGrant.html
18. Hildebrand, R,Mokhtar, W,Bryan, S,"Vehicle Dynamics Con: Capitalizing Breadth &Duration. ASEE AC 2009
19. McDonald, D,Hildebrand, R, "Laboratory Learning ExperiencesVehicle Engr, SAE World Congress, 2010
20. http://vector-cantech.com
21. http://www.ni.com
22. http://www.revoleng.com
23. http://www.mathworks.com/products/daq
24. Mohammadzadeh, A. and Haidar, S., "Analysis and Design of Vehicle Suspension system using MATLAB and Simulink, ASEE Annual Conference, 2006.
25. McDonald D., Simulation Learning Experiences in Energy Conversion with Simulink and SimPowerSystems, ASEE Annual Conference 2006.
26. http://www.mathworks.com/products/vehicle-network/
27. Duesing, P., McDonald, D., Schmaltz, K., "Qualitative and Quantitative Assessment to Accomplish Continuous Improvement", National Assessment Institute Conference, 2005.

## Acknowledgements

Vector CANtech supported the acquisition of CAN software and hardware at a tremendous discount via the Can Goes to College grant program.

Kettering University engineering faculty and staff provided support and encouragement with information on the CAN Goes to College program and the dynamometer setup.

Revolutionary Engineering donated a chassis vehicle dynamometer and assisted a senior design team in refurbishing and commissioning the dynamometer for use in engineering instruction.

.