

Work In Progress

Introductory LabVIEW Real Time Data Acquisition Laboratory Activities

David McDonald
Lake Superior State University, dmcDonald@lssu.edu

Abstract - This Work In Progress paper provides an overview of initial activities to use the National Instruments single-board Reconfigurable Input Output (sbRIO) development unit for instructional laboratory activities. The paper reviews some on-going LabVIEW and LabVIEW Real Time applications from both industry and education, provides an overview of real time, and explains setting up the hardware and software. The paper also discusses software strategies for applying the sbRIO, provides an overview of the sbRIO capability, and discusses an example application in education.

Index Terms - LabVIEW Real Time, Data Acquisition, Computer-Based Laboratory Activities

INTRODUCTION

Computer-based activities can be effective tools to engage students in laboratory learning experiences by leveraging their existing computer skills while introducing new concepts and skill sets. Laboratory activities in data acquisition and control applications can be helpful tools to cultivate students' interest and motivation while introducing valuable laboratory approaches, skills, and techniques.

Most academic data acquisition activities use low cost data acquisition computer boards and software tools. The transition to introductory real-time experiences will require additional resources, but it is still relatively straight forward. Even though new hardware and software resources are required, steep academic pricing discounts help tremendously to offset the cost, and there is significant documentation with examples available to help make the journey to real time applications successful.

APPLICATIONS OF LabVIEW REAL TIME

National Instruments offers a variety of Compact Reconfigurable Input Output (cRIO) [1] units that are designed for rugged and tough industrial applications. National Instruments also offers a similar unit as a single board RIO (sbRIO) [2] for development applications. These units have seen wide use in industry, and with academic pricing incentives, are appropriate for academic use.

Pittsburgh, PA

1. Examples From Industry

Efficiency of electric motors is important in efforts to improve energy independence of the United States. One area to improve efficiency is to use more efficient motors. Crouch [3],[4] indicates that data from the US Department of Energy estimates that switching to a motor with a four to six percent higher efficiency rating can pay for itself in just two years if the motor is in operation for more than 4,000 hours a year. High efficiency Brushless DC and Permanent Magnet Synchronous AC Motors are available; however, the power electronics uses a Field Oriented Control strategy that requires many computations of complex control algorithms making MCU implementation more challenging. Crouch reported that BAE Systems Avionics used RIO with Xilinx FPGAs to squeeze 15 percent extra performance from its existing motors while saving weight in avionics products by reducing motor mass.

Hopkins [5] discussed how Lockheed Martin used LabVIEW Real Time and Simulation Interface Toolkit to build a prototype integrated avionics unit (IAU) and a hardware-in-the-loop (HIL) simulator to test the light detection and ranging (LIDAR) control software functionality. They chose LabVIEW rather than traditional text-based, real-time development tools because LabVIEW Real Time enabled faster system development time by a small staff of engineers.

2. Examples From Education

Pretolani [6] from the School of Engineering at the University of Bologna used cRIO to develop a helicopter platform capable of autonomous flight to be used for control and navigation research in a university setting. The cRIO was easy-to-use and reliable enough for helicopter control.

Currier [7] reported that the DARPA Urban Challenge required a ground vehicle to autonomously navigate through an urban environment. To complete the course, Virginia Polytechnic Institute and TORC Technologies LLC used LabVIEW Real Time and cRIO hardware to power a 2005 Ford Escape Hybrid for the Darpa vehicle challenge. It traversed the 60 miles in less than six hours while navigating traffic through roads, intersections, and parking lots.

March 26 - 27, 2010

Sundaran [8] reported on using LabVIEW virtual instrumentation to enhance the measurement and analysis of a MATLAB/Simulink real-time Hardware-In-The-Loop (HIL) controlled machine drive experiment.

Craig and Rosmarin [9] discussed the use of LabVIEW and cRIO to develop a Mechatronic System Design Project at Rensselaer Polytechnic Institute to prototype a two-wheeled robotic locomotion platform.

Muecke [10] reported on using LabVIEW Real Time to develop a robotic system that acted as a research platform for studies on novel locomotion and to serve as the first United States entry into RoboCup, a soccer competition for autonomous robots. Muecke said that “LabVIEW helped create expandable, adaptable software that eased and accelerated research while supporting the robot to perform higher-level tasks, such as playing competitive soccer.

Sharad [11] outlined that LabVIEW virtual instrumentation can serve as a cross-discipline platform to teach fundamental engineering concepts across disciplines.

Ganesan [12] reported that LabVIEW played an important role in the instructional aspects for validation of automotive embedded controls in a course on automotive instrumentation.

LabVIEW graphical programming can be an effective tool to enhance instruction in a wide range of educational courses and projects. LabVIEW has found acceptance in instructional areas. LabVIEW Real Time has been used for special applications, but does not appear to have found wide acceptance for instructional laboratory activities in engineering education. As will be illustrated throughout this paper, introducing LabVIEW Real Time is possible and affordable with current academic discounts.

The following section introduces LabVIEW Real Time, and further sections provide information on getting started using this technology.

LABVIEW REAL TIME

The National Instruments LabVIEW Real-Time Module allows the user to combine LabVIEW graphical programming with the power of a real-time operating system. This empowers the user to develop data acquisition and control applications that occur within a set time period. In a non- Real Time application the user is not ensured that a measurement or control response will occur within a given time period.

Real time does not necessarily mean fast, but it can be fast. Real time does mean the application will respond on or within in a predictable time. The ability to respond at a set period of time is referred to as determinism, and the system is considered to be deterministic. Because deterministic systems are predictable then the user is guaranteed that measurements will be taken at set, predictable times.

In many data acquisition applications repeated measurements are made in a loop, and the time it takes a loop to process is called the loop cycle time. Even with Real Time the loop execution time can vary. This variation in

loop cycle time is called jitter, and the maximum amount of time variation is referred to a maximum jitter.

A computer application that operates within a larger system is an embedded application. The embedded application may operate without a keyboard, monitor or mouse, and therefore has no Human Machine Interface (HMI). These embedded applications are considered to be operating headless, and frequently have a limited amount of memory, physical space and other resources.

Windows-based applications do not run within specific time limits because of the need for windows to balance requests from multiple programs, screen savers, anti-virus software, and service interrupts from input/output devices such as USB ports, keyboard, mouse, Ethernet card, etc. Thus, because processor time is shared it can preempt high-priority data acquisition and control programs and determinism is not guaranteed. Windows applications are therefore non-deterministic

The LabVIEW Real-Time Module operating on data acquisition and control hardware targets ensures that high-priority loops execute first, and do not require user input from peripherals such as keyboard, etc.

Symmetric Multi-Processing (SMP) allows multi-core processors to run a single instance of the operating system and connect to a common main memory while running code in parallel. LabVIEW Real Time (LVRT) includes extensions for SMP to take advantage of high-performance multi-core processor technology. [13]

LabVIEW has a feature for Timed Loops that ensures that Time-Critical loops run on time. When the programmer sets up a Timed Loop he/she can assign loop priority and can even assign a specific processor to that loop to help manage the processor load when using a multi-core processor.

TARGET CONFIGURATION

In the discussion below, the Host is the personal computer or laptop that will be used by the programmer to communicate with the real time system. The Target is the specific real time computer, Real Time Embedded Controller within a PXI chassis, a compact RIO controller, or a Single-Board RIO development board system.

1. Ethernet Connection

Before starting to configure the software, there needs to be an Ethernet crossover cable connection between the PC – Host and the Real Time Target. The IP addresses needs to be assigned, and both the Host and Target need to be on the same Subnet.

2. Measurement Automation and Explorer (MAX)

MAX software installation occurs when LabVIEW is installed on the host computer. This utility is used to setup the hardware interface between the Host and Target, and to

install software from the host to the target. When MAX is opened, Remote Systems can be expanded to show the hardware and software of the Target. The IP address of the Target can be obtained automatically using *Obtain an IP address automatically* (DHCP), or an IP address can be assigned. The Subnet Mask must be the same on both the Host and Target.

The Remote Systems connection shown below in **Figure 1: Measurement and Automation Explorer** can be expanded to show the hardware system. In this case the system has been named NI-PXI-Chassis, and the IP address has been set to 10.10.10.82. The PXI-8110 Embedded Controller is a very powerful 2.26 GHz, Intel quad-core processor that lists for about \$5,600. It has build-in GPIB and Serial ports along with monitor, keyboard, printer, and USB connections.

In addition to the controller the PXI-Chassis has a National Instruments NI-PXI-6259 M Series Multifunction DAQ input-output board with 32 Single / 4 Differential analog inputs, 4 analog outputs, 48 digital I/O and two counter/timers. The PXI-chassis also has two Controller Area Network (CAN) computer boards. Each CAN card has two high-speed ports that can be used to receive/send CAN network messages.

3. Software Installation

The software tab under Remote Systems is used to install Real-Time on the Target from the Host computer.

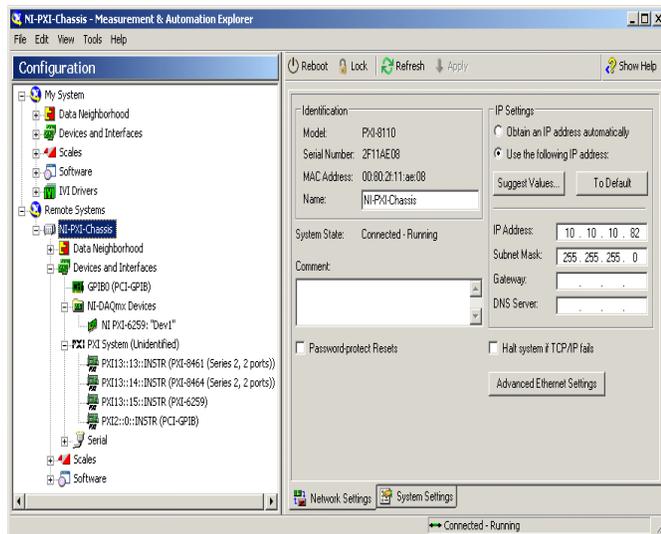


FIGURE 1
Measurement & Automation Explorer: Example Target Configuration

REAL TIME PROJECT

A Real Time Project must be used when programming LabVIEW real time applications. A Project enables the user to setup folders and libraries to organize files, manage targets that represent systems that LabVIEW can be run on, and build executables or use source control.

As shown below in **Figure 2: Real Time Project**, the project contains information on the LabVIEW programs (VIs) that will be running on the Host and on the Target. It also shows the Real Time Shared Variables that are used in the Target VIs.

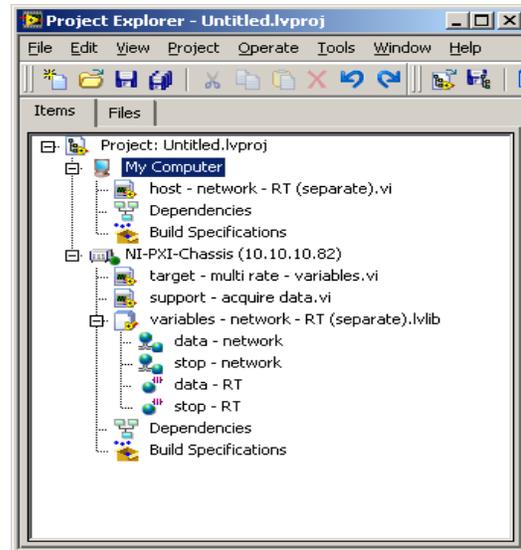


FIGURE 2
Real Time Project

SHARED VARIABLE

In addition to local and global variables, LabVIEW includes shared variables specifically for Real Time applications. These may be single-user (single computer) or network deployable. In this example, the data-network and stop-network variables are used across the network to communicate information on the measured data and the STOP control. These variables can be used on both the Host and Target, but will be housed on the Target because they were created on the Target.

The data-RT and stop-RT variables are used on the Real Time Target VI (or LabVIEW program). These variables are First In First Out (FIFO) Data Queues that are used to avoid over-writing or other data loss when communicating data between two loops on the Target.

Shared Variables can be enabled as Real-Time FIFOs. When defined this way the user specifies single-element or multi-element size of the FIFO. Other Real Time data types include programmed Functional Globals [14] and Real Time FIFOs.

REAL TIME PROGRAM ARCHITECTURE

The programming of the Real Time Target begins with good architecture [15], and involves the need for a deterministic program that will respond to an external event within a given time and predictably meet deadlines. This section looks at the need for determinism of the Host and Target programs.

1. Deterministic

Deterministic applications normally perform a critical task in an iterative fashion with all iterations taking the same amount of processor time. Deterministic applications should reliably respond to inputs and supply outputs with little jitter. An example application is a time-critical loop which is monitoring a process and needs to adjust variables or stop the process at specific times.

2. Multitasking and Multithreading

Multitasking is the ability of the operating system to quickly switch between tasks where the task is likely to be an entire application such as LabVIEW. Multithreading is an extension of the idea of multitasking into applications so that specific operations within a single application can be subdivided into individual threads each of which can run in parallel. The operating system then can divide processing time among different applications, and among different threads within an application.[16],[17] To ensure determinism and minimize jitter in LVRT applications tasks are separated into Time-Critical Tasks and Non-Critical Tasks. As shown in Figure 3: Host Computer - Real Time Target, the Host computer is responsible for non-deterministic tasks including: Communication with the Target, User Interface, and file tasks.

The program on the Real Time Target is concerned with a combination of time-critical priority tasks and normal priority tasks. In order to accomplish this, the Target Real Time program must multitask by being able to handle multiple software threads or complete multiple loops.

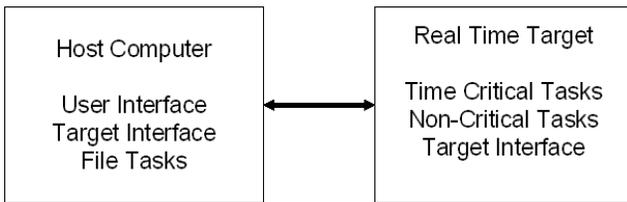


FIGURE 3
Host Computer - Real Time Target

Time Critical Tasks include items such as control loop or safety monitoring tasks that must be completed on time, every time. These items or tasks are placed in a Time Critical Loop. Non-Time Critical Tasks include items such as logging data or communication with a serial device or GPIB device that might not respond within an expected period of time. These items or tasks are placed in a Non-Time Critical Loop because putting inside a Time-Critical Loop might increase loop jitter and thus compromise determinism. LabVIEW uses a combination of preemptive scheduling to give Time Critical Tasks the highest priority, and round-robin scheduling where Normal Priority Tasks occur in a pattern.

If a processor's time is dedicated to Time Critical Tasks, then other tasks will not get done. This means that the loop times must allow for the processor to have some Sleep Time to yield processor resources to complete Non-Priority Tasks. Without adequate Sleep Time in the High Priority Loop, the other tasks will not get accomplished and those tasks will experience Starvation due to lack of resources.

An example of dividing tasks is shown below in Figure 4: Example Target Program: Time Critical and Non-Critical Loops. The higher priority (top) loop has been assigned a priority of 100 and the lower loop priority is set at 50. LabVIEW Real Time does provide debugging capability for programs on the target, and programs should be benchmarked to ensure proper operation.[18]

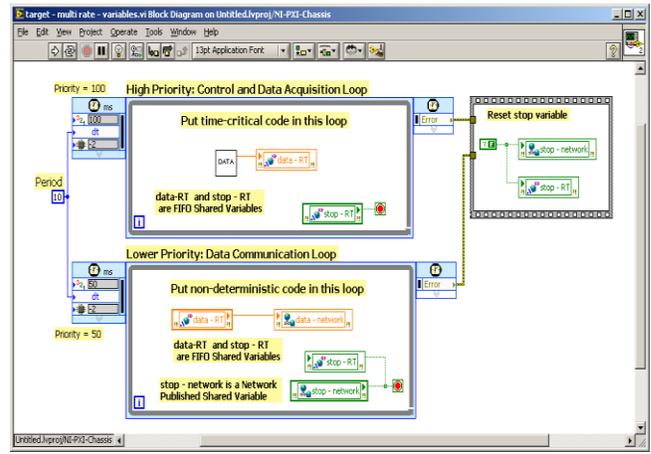


FIGURE 4
Example Target Program: Time Critical and Non-Critical Loops

HARDWARE SINGLE-BOARD RIO

The National Instruments Single-Board RIO is available in six configurations, and is appropriate for academic use. The sbRIO board [19] uses an industrial Freescale MPC5200 real-time processor, Xilinx Spartan-3 Field Programmable Gate Array (FPGA), analog and digital I/O, and ethernet and serial ports. A picture of the sbRIO is shown below in Figure 6: sbRIO 9632 Development Board

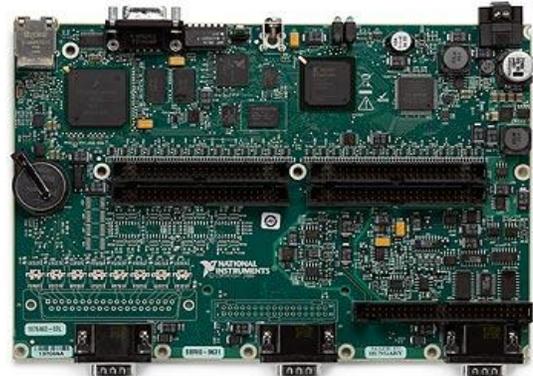


FIGURE 6
sbRIO 9632 Development Board

SOFTWARE AND SUPPORT RESOURCES

National Instruments provides academic bundle packaging. The bundle includes LabVIEW, LabVIEW Real Time, FPGA, Measurement Automation and Explorer (MAX) and other software. The software is first installed in the Host PC, and downloaded from the Host to the Target when MAX is connected to the Target.

Getting started in new hardware and software can be challenging. National Instruments offers academic discounts, and it is also possible to purchase hard copies of the training materials at a reduced price. The NI website, www.ni.com, has multiple support items for getting started with LabVIEW Real Time and FPGA. [20], [21]

LABORATORY APPLICATION

LabVIEW instruction is included in the EGEE365 Vehicle Instrumentation course.[22],[23] This course can be taken as an elective or as a requirement to complete the option in Vehicle Systems within Electrical or Mechanical Engineering. Other software instruction in the course includes MATLAB/Simulink and Vector CANtech CANoe, which is used for vehicle CAN bus analysis and simulation.

The course includes testing and benchmarking laboratory activities along with the software instruction. LabVIEW Real Time 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. Both the temperature and the probe's position are recorded and plotted to provide the profile. A picture of the general setup is shown below in Figure 7: Dryer Temperature Setup

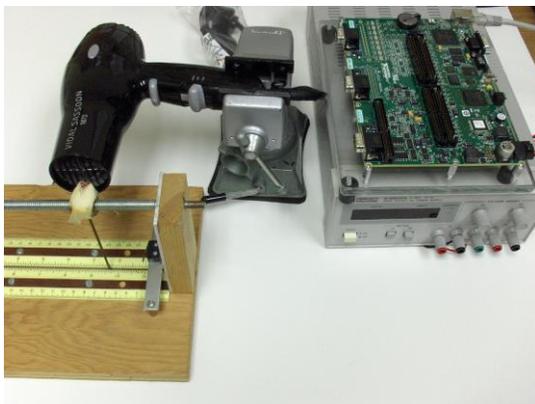


FIGURE 7
Dryer Temperature Setup

The activity includes instruction on the positioning motor, switches, temperature sensors, etc. used in the testing fixture. Students develop a data acquisition and control program for the test system in LabVIEW (Windows-based).

Later the exercise is repeated as an introduction to using LabVIEW Real Time and sBRIO Single Board hardware.

This exercise introduces the students to many different aspects of setting up a test cell. Benchmarking is introduced both in the form of testing different hair dryers, but also within the software design. Students are introduced to benchmarking various ways of transferring data within LabVIEW. If a temperature sensor and meter are used to measure temperature, then students learn to install and program instrument drivers, and benchmark how long it takes to acquire a reading from the instrument. The response time will be used to determine if the data is read in the time-critical or non-critical loop in order to reduce jitter and ensure a deterministic software application. Future plans for the course include investigating developing a data-logger [24].

From a Lessons Learned perspective, the prerequisite courses, EGEE210 Circuit Analysis and EGNR365 C Programming, appear to be appropriate and provide students with appropriate software and electrical background.

The Vehicle Chassis Dynamometer shown below in Figure 8: Senior Design Project: Vehicle Chassis Dynamometer is also used in the course. A senior design team refurbished the dynamometer during 2008-2009, and it is being used in some laboratory courses.



FIGURE 8
Senior Design Project: Vehicle Chassis Dynamometer.

The application software for the dynamometer is a limited version of REPS, a professional, LabVIEW-based, Real Time, dynamometer software package from Revolutionary Engineering. [25]

The course was last offered as a Special Topics course at three credits. Assessment outcomes were positive with overall student assessment for the course laboratory at 82%. The Professionalism Component for the 3-credit Special Topics version was Design – 0.3 credits, Software – 2.2 credits, and Communications – 0.4 credits. The course is now a permanent course offering at four credits.

CONCLUSION

LabVIEW graphical programming is a powerful industrial tool that is available for academic use. LabVIEW Real Time is an extension of LabVIEW and provides an avenue to introduce students to real time programming experiences.

Programming in LabVIEW Real Time introduces real time projects, and requires students to think through the concepts of deterministic software design, multi-tasking and multi-threading, benchmarking, and design of software for reliable operation and reduction of data loss. Introducing this information during a standard course will equip students for more advanced senior projects or special project activities while expanding their skill set with experience in this area.

National Instruments Real Time has been used in industrial project as well as special academic student projects and as support for research projects. The goal of this paper was to encourage engineering educators to consider introducing real time using LabVIEW Real Time. The introduction of LabVIEW Real Time for academic use is available at a reasonable cost with academic discounts, and opens up opportunities for students to complete interesting work while building their professional skill set.

AUTHOR INFORMATION

David McDonald Professor, Department of Engineering and Technology, Lake Superior State University, Sault Ste. Marie, MI, 49783, dmcDonald@lssu.edu.

REFERENCES

[1] <http://www.ni.com/compactrio/>
 [2] <http://sine.ni.com/nips/cds/view/p/lang/en/nid/207169>
 [3] <http://www.xilinx.com/publications/archives/xcell>,
 “FPGA Powered Platform Controls Industrial Motors for Maximum Efficiency,” by Greg Crouch, posted November 7, 2009\
 [4] <http://zone.ni.com/cda/tut/p/id/10048>, “FPGA Powered NI CompactRIO Controls Industrial Motors using Field Oriented Control (FOC)”, by Greg Crouch, Publish Date: Oct 27, 2009.
 [5] <http://www.ni.com/academic/educators/> Hopkins J., Lockheed Martin Space Systems Company National Instruments, National Instruments, January 21, 2010.
 [6] <http://sine.ni.com/cs/app/doc/p/id/cs-692>, Pretolani , R., Teodorani, B., Saggiani, G. “Using CompactRIO to Develop a Rotocraft Unmanned Air Vehicle” as posted on January 31, 2010.
 [7] <http://www.ni.com/academic/educators/> Currier, P., Hurdus, J., Wicks, A., Reinholtz, C., “Team Victor Tango’s Odin: Autonomous Driving Using NI LabVIEW in the DARPA Urban Challenge,” National Instruments, January 21, 2010
 [8] Sundaram, Ramakrishnan, Mak, F., Tandle, S., “Virtual Instrumentation Interfaces for Real-Time Control and Disply of Electric Machine Drives, Proceedings of the ASEE Annual Conference 2008

[9] <http://www.ni.com/academic/educators/>Craig, K. and Rosmarin M. “Two Wheeled Balancing Transport Platform,” National Instruments, January 21, 2010.
 [10] <http://www.ni.com/academic/educators/>Karl Muecke, “Development of a Fully Autonomous Humanoid Robot for Novel Locomotion Research and as the First U.S. Humanoid Entry to RoboCup,” January 21, 2010.
 [11] Sharad, S. “ Teaching Engineering Concepts Across Multiple Disciplines Using A Single Platform Based On Virtual Instrumentation,” ASEE AC 2007. Paper 1708
 [12]Ganesan, S. , “A Course on Automotive Embedded System Analysis, Verification and Validation., “Improving Student Understanding of Instrumentation and Measurement in US Engineering Undergraduate Programs” ASEE North Central Section Conference Proceedings 2009, Grand Rapids, MI.
 [13] <http://zone.ni.com/devzone/cda/tut/p/id/6403>
 “Introduction to LabVIEW Real-Time Symmetric Multiprocessing (SMP).” Multicore Programming Fundamentals White paper Series. National Instruments. January 21, 2010.
 [14] <http://zone.ni.com/devzone/cda/tut/p/id/4338>
 Improving Determinism with Functional Globals, LabVIEW 2-Style Globals in LabVIEW Real-Time
 [15] <http://zone.ni.com/devzone/cda/tut/p/id/2833>
 LabVIEW Real-Time Architecture and Good Programming Practices
 [16] <http://zone.ni.com/devzone/cda/tut/p/id/6424>
 Differences between Multithreading and Multitasking for Programmers
 [17] <http://zone.ni.com/devzone/cda/tut/p/id/14>
 Multithreading in LabVIEW Real-Time
 [18] <http://zone.ni.com/devzone/cda/tut/p/id/5424>,
 Benchmarking a Typical Application using LabVIEW Real Time and NI-DAQmx, as posted on January 21, 2010.
 [19] <http://sine.ni.com/nips/cds/view/p/lang/en/nid/205899>, sbRIO 9632, as posted on February 8, 2010.
 [20] <http://www.ni.com>, “Getting Started with the LabVIEW Real-Time Module,” On-line document. Search Part Number 371375C-01.
 [21] <http://zone.ni.com/devzone/cda/tut/p/id/9988>,
 “LabVIEW Real-Time Module Training,” Free On-line training module. As posted on January 30, 2010.
 [22] McDonald, Hildebrand “Laboratory Learning Experiences in Vehicle Engineering,” SAE World Congress, 2010.
 [23] McDonald “Instrumentation and Data Acquisition in a Vehicle Instrumentation Course,” ASEE 2010 AC.
 [24] <http://zone.ni.com/devzone/cda/epd/p/id/5751>, “In-Vehicle Datalogger Reference Application for CompactRIO,” Example code and documentation. As posted on January 30, 2010
 [25] www.revoleng.com