2,000 Robotic Applications Using the National Instruments CompactRIO Embedded Control System

Vincent Wilczynski, Dean of Engineering, U.S Coast Guard Academy, New London, CT
Jonathan S. Mittelman, MD, MPH, Medical Director, Dominion Nuclear of Connecticut, Waterford, CT
Nathan Lim, Development Engineer, PVI Systems, Niantic, CT

Abstract— *Robotic control is a challenge for all applications, and simultaneous wireless control of multiple robots is an even greater challenge. In 2009, the FIRST Robotics Competition migrated to a new control system that relied on wireless communication (802.11n standard) between the controllers and the robots. This new control system was developed and implemented over the course of a single year, and nearly 2,000 teams competed with the new system during the 2009 FIRST Robotics Competition season. The National Instruments (NI) CompactRIO embedded controller served as the center of the control system, and this controller was augmented with additional hardware to interface with the robot and its controllers. The specific components of the 2009 FRC control system consisted of a driver control system, a wireless communications system and an on-board robot controller. The controller was programmable in LabVIEW and C/C++ using the WPI Robotics Library that was written to support the new controller. This paper describes the components of the 2009 FIRST Robotics Competition Control System and explains how the system enabled the use of vision systems, feedback control, autonomous control, and system monitoring. The experiences of an award winning team that participated in the 2009 FIRST Robotics Competition provide examples of the utility of this new robot control system*.

I. Introduction

In 2009, FIRST (For Inspiration and Recognition of Science and Technology, a 501(c)(3) organization dedicated to inspiring youth to pursue technical careers) challenged itself in much the same way it inspires students world-wide. The FIRST Robotic Competition undertook a total makeover of its robotic control system with a pre-determined deadline, a limited budget and limited manpower. In the process, FIRST completed its mission and managed to increase the level of computing power available for robotic competitions by an order of magnitude. This paper is a description of that accomplishment.

The FIRST Robotics Competition is an international contest that engages nearly 2,000 teams with a unique robotic based challenge. Over the course of six weeks, the 100,000 participants in FIRST design, manufacture and test robotic platforms capable of competing in a mechanical sport. The teams then gather at regional and championship competitions to compete with their 120 pound robots, and in the process show young people that a future in science and technology is accessible, achievable and rewarding. The competition is one of inspiration and motivation, as the program encourages students to pursue education and careers in the fields of engineering, science and math. At the competitions, alliances of three robots compete as a team against the opposing alliance in a series of matches over the course of a weekend. A full description of the FIRST Robotics Competition and the engineering applications of the program are detailed by other technical publications (references 1 and 2).

At the beginning of the season in January, FIRST Robotics Competition teams learn about that year's contest and begin their six-week design, construction and testing process. Each year a new game is created for the competition and this new challenge helps keep a level playing field between new and returning teams. Teams are provided with a common set of components from which to construct their robots, though additional mechanical and electronic components may be procured or fabricated. The contest is one that requires ingenuity, technical competence and effective planning, and the most successful teams are those that develop the strongest partnerships between the students and mentors.

The provided kit of parts includes all components of the robot control system. Typically, robots operate in both autonomous mode for the first part of a match, and in tele-operated mode for the remainder of the match. Because six robots compete on the playing field for each match, each robot control system must interface with a competition control management system. The competition control management system is called the field control system and enables coordinated on/off control of all six robot control systems while allowing individual control of the functions on each robot.

In 2009, the FIRST Robotics Competition migrated to a new control system that incorporated wireless technology, commercial data acquisition and control components, and two independent programming methodologies. From 2000 to 2009, an Innovation First, Inc. (reference 3) control system was used for this competition. The IFI system developed over time and resulted in a very reliable 900 MHz wireless, user programmable micro-controller that was C-programmable. This legacy system established the baseline for FIRST Robotics Competition control and its features were the starting point for the 2009 FIRST Robotics control system. This paper details the components of the 2009 FIRST Robotics control system and documents how the system was used for robot control.

II. THE FIRST ROBOTICS COMPETITION

To provide context for understanding the FIRST Robotics Competition (FRC) control system, it is fitting to briefly review the design challenges a typical FIRST team faces. In 2009 FRC teams had to design and construct robots that could navigate a slippery 27 ft x 54 ft playing surface while pulling a trailer behind. The task for the robots consisted of capturing 8 inch balls from the playing field surface and depositing those balls in an opponents' trailer while avoiding getting balls dumped into your own trailer. Matches consisted of 15 seconds of autonomous action followed by two minutes of driver-controlled play. Six robots competed in each match where three robots formed an alliance to compete against the other three robots.

On average, 48 FRC teams competed at each of 45 regional competitions, and 348 teams competed at the FIRST Championships in Atlanta. Each competition spanned three days, with the first day devoted to practice, the next day and a half devoted to qualification rounds (where teams were randomly assigned to alliances) and the finals. During the finals, the top 8 teams chose their two alliance partners for double elimination tournament play. It is reiterated that the purpose of the competition is not centered on wins and losses on the playing field, but rather it is to connect youth with mentors that can assist them in education and career pursuits. In fact, the highest award at the competition is presented to the team that best achieves the goal of inspiring youth to pursue technical careers.

For equity, FRC teams are limited to the motors, servo-motors, solenoids, compressor, and 12V 17 Amp hour battery provided in the kit of parts supplied to each team. The robot controller is responsible for energizing the motors, compressor and solenoids based on feedback from on-board sensors or driver commands.

Each motor is powered with a relay or speed controller that receives instructions via a PWM signal created by the control system. Typical sensors used on a FIRST Robotics Competition robot include a camera, optical encoders, potentiometers, limit switches, range detectors, and Hall effect sensors.

Each of these digital or analog sensors is monitored by the robot controller, with the state of the input signals triggering functions as determined by the robot control program. Teams are provided with default code in the robot control program which they can modify to optimize performance for their system.

Figure 1 is an example of a FIRST Robotics Competition robot created for this competition. In this design, each of the four drive wheels is independently powered, and each set of front and back wheels is independently steered. Wheel speed is monitored with quadrature encoders, and the steering direction is monitored with a rotary potentiometer. A Proportional-Integral control loop was designed and implemented to position the wheels in response to the driver commanded location. The ball retrieval and delivery system is a series of belt driven tubes, powered by a collection of motors. A camera is mounted on the robot to help detect the location of an opponent's trailer in both autonomous and tele-operated modes. This example typifies the sophistication of the FIRST Robotics Competition machines.

Figure 1 - FRC Team 236 Robot

III. 2009 FIRST ROBOTICS COMPETITION CONTROL SYSTEM

The 2009 FIRST Robotics Competition Control System is composed of three subsystems: the driver console, the wireless communication system and the robot controller. This system was designed and tested using a wide team of developers that understood the competition demands. While some components of the new control system utilized commercially available items, others were modified commercial items, and the remaining components were original designs created and manufactured for the competition. Some components of the control system were added to make use of FIRST-standard interfaces such as speed controllers and voltage relays. The components and their relationship to each other are illustrated in Figure 2.

The driver console, also known as the Driver Station (DS), was specifically designed for this competition and served as the interface between the human drivers and the robot. The heart of the DS is an ARM9 based single board computer running Linux. This unit accepted up to 4 USB joystick or handheld controllers and had capability to accept team-built electronic interfaces for robot control. An LCD screen was incorporated into the driver console to display parameters such as the team's identification number, system status and battery voltage.

The console included switches to place the robot in either an operating or disabled state, and additional switches to command robot actions in both autonomous and remote control mode. The driver console included a connection that coupled each unit with the field control system at a competition to coordinate robot power cycles. An Ethernet port for wireless programming of the robot controller provided a tethered connection between the driver station and the on-board robot control system for off-field testing.

The wireless communication system, composed of a commercially available wireless game adapter on the robot and a commercially available wireless router, served as the un-tethered link between the driver console and the robot. The upper 5 GHz band with the 802.11n wireless protocol provided 54Mbs data transfer between the robot and the driver console. IP addresses were assigned to each component of each team's control system using the team number as a base, as indicated in Figure 3.

The robot controller consisted of a subset of components: the National Instruments CompactRIO programmable automation controller (PAC), a power distribution board which provided fused power to all of the robot's electrical components, and interface modules to connect the control system to sensors and actuators. The National Instruments CompactRIO programmable automation controller, comprised of a field-programmable gate array (FPGA) and a 32-bit 400 MHz Freescale Power PC processor, served as the backbone of the robot control system. The NI CompactRIO used in the FIRST Robotics Competition was a special edition of this rugged and reliable mobile device controller.

Memory capacity of the CompactRIO included 64 MB pf system memory, 128 MB of nonvolatile flash storage, and

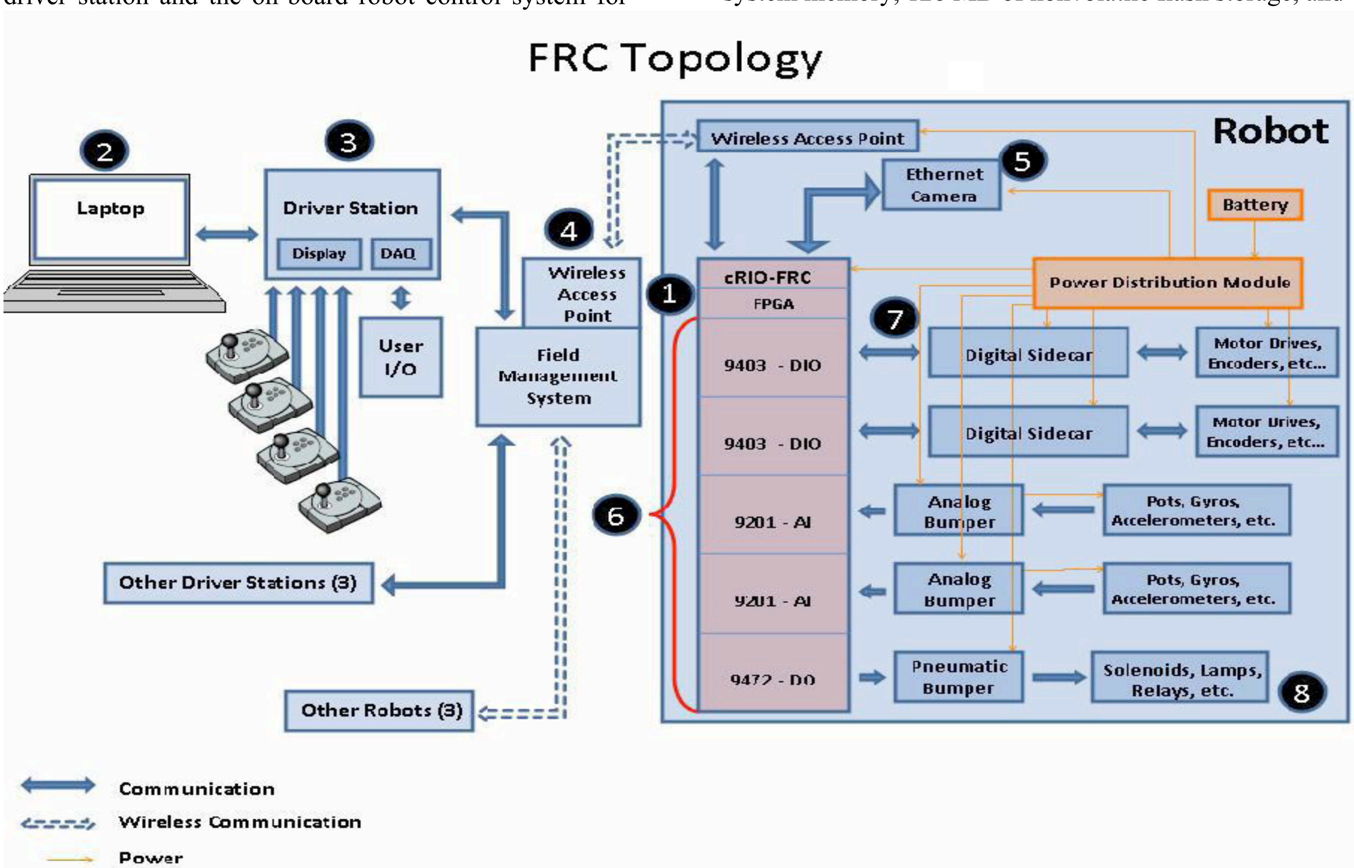

Figure 2 - FRC Control System Components (reference 4)

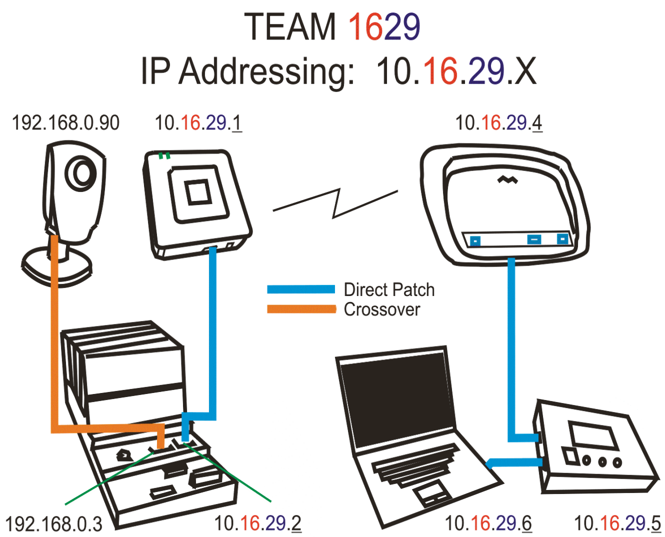

Figure 3 - IP Address Allocation (reference 5)

a 2M gate FPGA. The FPGA enabled the input/output processing to be separated from and run in parallel to the controller software, thereby increasing the operating speed of each function. This was accomplished with (Direct Memory Access) transfers from the FPGA to the Power PC Processor over a high-speed PCI Bus. Transferring data directly from the FPGA to the Power PC's onboard memory allowed the processor to perform other tasks such as communication and signal processing.

The CompactRIO could be configured for programming in either the NI LabVIEW language or the WindRiver C++ IDE. The hardware-specific code, which was downloaded to the FPGA, was developed in a partnership between the Worcester Polytechnic Institute (WPI) and National Instruments. Development software and default programs were provided for each programming environment.

Five standard data acquisition and control modules plugged into the CompactRIO to receive and transmit signals, and each module included a custom made interface to optimize its use with the FIRST Robotics Control system as illustrated in Figure 4. 16 analog input signals could be monitored with two NI-9201 Analog Input modules. Analog breakout boards were designed and manufactured to easily and securely attach sensors and to provide a 5V DC power source from the CompactRIO for analog sensors mounted on the robot. One input on the module included a jumper to designate a specific channel of the module to monitor the robot battery's voltage.

Two NI 9403 Digital Input/Output modules and their associated digital side cars (designed as an interface between the module and the digital components) allowed 32 digital signals to be connected to the control system. The interface was used with inputs, such as sensors, or outputs, such as PWM connections to relays, motor controllers, and servos. Jumpers on this module provided 6V power to each output that was connected to a servo. A primary function of these modules was to provide a circuit for PWM signals that directed speed controllers and voltage relays.

The NI 9472 Digital Output module and breakout board supplied 12V, 3/4 amp power on each of 8 channels. This interface supplied power to pneumatic solenoids, thereby

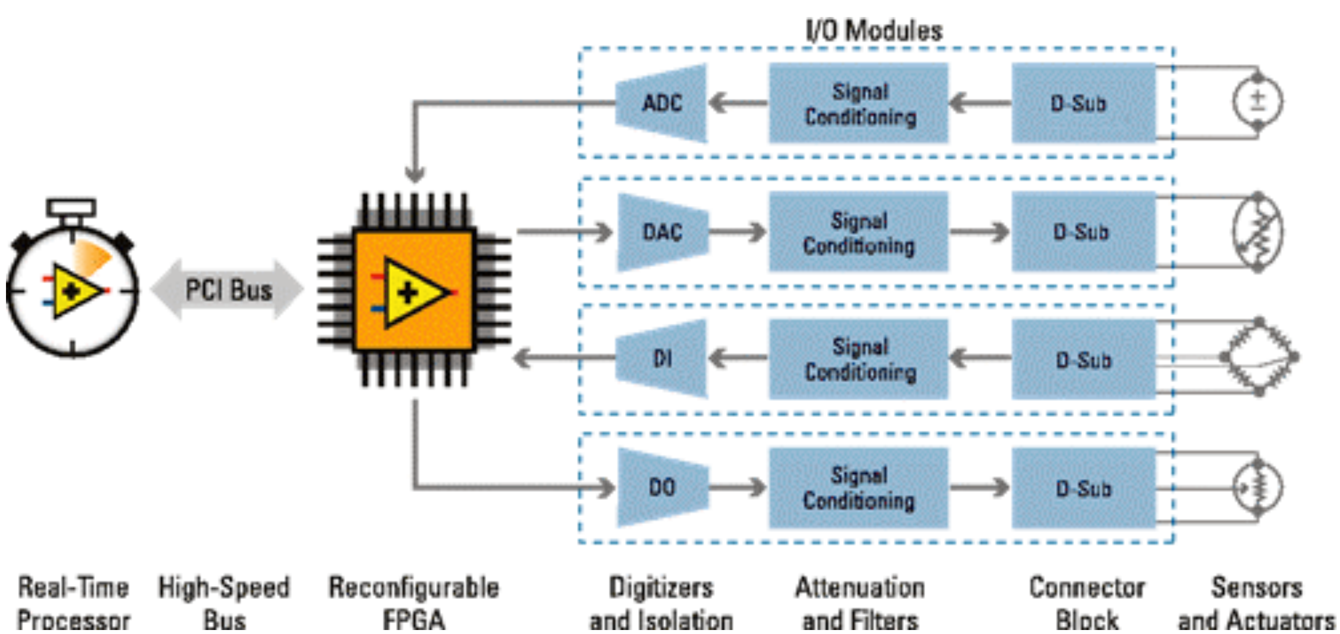

Figure 4 - NI CompactRIO I/O Modules (reference 4)

allowing these actuators to be used without requiring separate relays to activate solenoids.

The need to provide multiple regulated voltage levels as well a central wiring hub led to the development of the Power Distribution Board for this control application. Connected to the robot battery (normally used for golf cart applications) by a 120 ampere main breaker switch, the Power Distribution Board provided 12 V to 8 outputs through 40A auto-resetting breakers and 12 outputs via 30A auto-resetting breakers.

The on-board voltage regulator also provided a 24V/1.5A boost supply for powering the CompactRIO, a 5V/3A buck supply for powering the Axis Ethernet camera, and a 12V/2A boost supply for the WiFi adapter. The boost supply tracked battery voltage when the battery is fully charged and greater than 12 V. All regulated supplies had over-current and reverse battery protection. Wago connectors enabled crimp-free connections to the non-regulated outputs and were a great feature of this board.

III. Vision System - AXIS Camera

The AXIS 206 supplied in the FIRST Kit of Parts delivered crisp and clear images using progressive scan CMOS image sensors and advanced signal processing techniques. This camera has a built-in web server and was connected to the CompactRIO by an Ethernet cable. The AXIS 206 provided Motion JPEG images at up to 30 frames per second in all resolutions up to VGA 640x480 pixels. It could operate in light conditions as low as 4 lux and had a manual focus.

Camera settings were addressable by the WPI FIRST Software Library (WPILib) function, such as white balance, brightness, sharpness, color level, compression, exposure priority, exposure, image size and frame rate. JPEG compression, a function of the latter two parameters, was performed by the AXIS Camera.

The FIRST Kit of Parts software suite also included vision software developed by National Instruments: Vision and Vision Assistant. Vision is the underlying software module that provided an API to perform hundreds of image processing functions for programmers using LabVIEW or C++. Vision Assistant provided users with a quick, interactive environment to prototype ideas using 'scripts' of the vision functions without having to write code. These 'scripts' could then be used to generate LabVIEW code that could be inserted directly into existing projects.

In addition, teams were provided with examples of vision code in both LabVIEW and C++ that detected the two-color targets being towed by each robot on the playing field. Teams were free to use or modify the examples, or design their own algorithms from scratch.

IV. Software

The CompactRIO devices used in the FIRST Robotics Competition used essentially the same low level software (e.g. VxWorks operating system) as existing commercial CompactRIO devices. If a team used LabVIEW, the LabVIEW Real Time run-time engine was installed on the CompactRIO system. If a team used C++, the compiled application ran directly on the VxWorks operating system.

Figure 6 illustrates the software systems used in the CompactRIO. At the Library / Runtime Layer, teams were provided with the FIRST Robotics Competition Virtual Instrument (VI), comprised of two expansive libraries, each available in both LabVIEW and C++. One of these libraries, the FIRST Vision palette, provided image acquisition via the kit of parts AXIS 206 camera as well as a variety of image processing functions. The WPI Robotics Library interfaced with the CompactRIO device to interface with the robot's hardware devices.

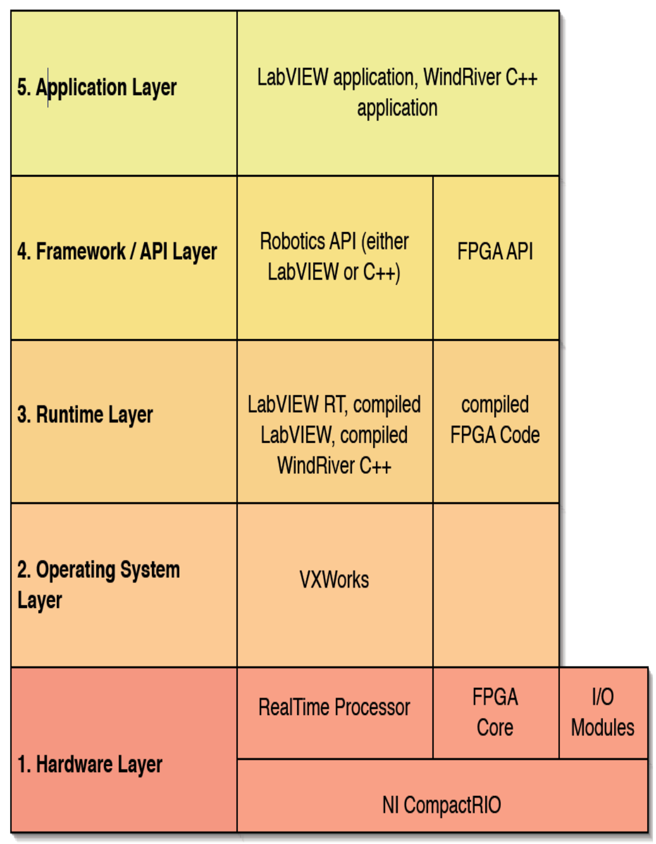

Figure 5 - NI CompactRIO Software Overlay (reference 4)

Among the many functions available in these libraries were C++ classes and LabVIEW VIs (virtual instruments) that addressed:

- I/O protocols: Analog, Digital, Serial Protocols: I2C, SPI (Serial Peripheral Interface - a synchronous serial protocol), and RS-232 (asynchronous serial protocol)

- General purpose counter/timers
- Analog triggering capabilities
- General purpose interrupts
- DMA channel that can source data from any subsystem
- Hardware Quadrature Decoding Sensors: Accelerometers, Compasses, Gyros, Encoders, Ultrasonic range-finders
- Vision: acquisition and analysis
- Actuator control: PWM Motor Controllers, Relays, Solenoids
- Driver's Station communications
- Control algorithms including PID controllers with auto-tuning
- User Watchdogs

The software used in this application resulted from a collaboration of individuals from many different organizations. Brad Miller from Worchester Polytechnic Institute was the primary developer of the C++ library, with significant help from National Instrument's Joe Hershberger (who developed the FPGA code and the C++ interface to it). Beth Finn of BAE Systems was primarily responsible for a C wrapper of the NI Vision Library, and Ken Streeter, also of BAE Systems, was primarily responsible for an out-of-box experience and control system documentation development as well as leading the beta program.

The specific software contributions of each collaborating organization included:

- WPI Library: Actuator and sensors
- BAE Systems : Vision
- National Instruments: Image processing & Control software (LabVIEW)
- WindRiver: C++ IDE

In both the LabVIEW and the WindRiver Workbench, a software framework was provided to the teams consisting of two project templates: an FRC Robot Project Template and an FRC Dashboard Project Template.

The provided FRC Robot Project Template was used to initiate communications with the CompactRIO, initialize a watchdog to detect unresponsive programs, and initialize the AXIS 206 camera. The template also provided each team an example of very basic motion and steering. The teams were then allowed to modify the code as needed to handle a considerable number of inputs, both from the human driver(s) and electronic sensors, implement advanced motion systems such as traction control (similar to an auto's anti-lock braking system, ABS), and actuate any other output devices that they had designed into the robot (such as other motors, pneumatics, hydraulics, and lights). The template was available in both a basic and advanced forms which allowed teams to tradeoff simplicity against additional flexibility.

The FRC Dashboard Project Template gave teams a simple out-of-box solution to view a display on the remote host computer. This image consisted of images from the camera and the state of the CompactRIO's inputs and outputs, thereby enabling an easy way to observe raw inputs and outputs while developing the robot. Because the Dashboard was a template, the teams also had the flexibility to customize the Dashboard as needed for their own use.

Once teams developed their code, they deployed them to the CompactRIO using one of the following methods:

Deployment for Debugging: In both LabVIEW and Workbench, the teams could elect to deploy their programs into the CompactRIO's RAM while maintaining a communications link with the host PC. LabVIEW or Workbench would then provide an interface that allowed the team to debug programs while the CompactRIO ran the program.

Standalone Deployment: In both programming environments, teams could deploy the programs to the CompactRIO's flash disk, which allowed the program to be permanently resident (until deleted) on the robot controller. At this point, the program would automatically execute upon boot-up of the CompactRIO. This is the mode that is used for competition.

V. System Integration: Tele-operated and Autonomous Control

Actual field play during the FIRST Robotic Competitions was controlled by the Field Management System (FMS). FRC game rounds consisted of a 15 second autonomous mode, followed by 2 minutes of tele-operated mode between 2 randomly generated alliances of 3 teams each. By controlling the value of a single byte, the FMS was able to control the state of field play.

The system cycled through the states in the following order:

Robot State	Sensor Input	DS/User Input	Motor/ Relay Output
Tele-op/Disabled	Enabled	Enabled	Disabled
Autonomous/Disabled	Enabled	Disabled	Disabled
Autonomous/Enabled	Enabled	Disabled	Enabled
Tele-op/Disabled	Enabled	Enabled	Disabled
Tele-Op/Enabled	Enabled	Enabled	Enabled
Tele-op/Disabled	Enabled	Enabled	Disabled

The Robot State variable was accessible to programmers and was used by both the LabVIEW and C++ programming environments to allow control of robot program structures. The provided control template code first established communications with the FMS and then read the single-byte variable Robot State. When not on the competition field, this Robot State variable was controlled by a switch on the Driver's Station and transmitted via a router to the robot. This allowed seamless integration between development and competition environments to enabled teams to develop and download their code without concern for the run-time environment.

The FMS connected by plugging into the Competition Port on the Driver's Station, where the 'Enable Dongle' was normally attached. When the robots were placed on the field and before play started in the Autonomous mode, Driver Station inputs could be used to provide initial conditions to the robot controller. Once the autonomous period began, all the user inputs of the Driver's Station to the CompactRIO was then disabled and the robot ran on code written to be enabled during Autonomous play. Switching into the 'Disabled Mode' between Autonomous/Enabled and Tele-op/Enabled allowed the programmer to zero encoders, set the bias of gyro chips and set default values of other variables. Once the mode was set to Tele-op/ Enabled, all user inputs were enabled. At the end of the Tele-op period when the match was finished, all motor and relay outputs were disabled.

The FMS software had many reporting functions in addition to running game play. It created the randomized match schedule as well as WPA codes to allow secure communications with the robots on the field. An additional output to the Team Pits Area outside of the Competition Arena allowed for concurrent display of team rankings as well as listing the teams needing to be queued up for play.

The FMS consisted of:

1. The 'Scorpion Case' which contained a dedicated PC. This PC hosted the Event Server which ran the Field Management Software and provided video feeds to the AV system. The audience (both local and internet-wide) could then observe select camera feeds as well as the real-time score and a count-down timer.

2. A Station Control Cabinet (SCC) was located at each end of the field; one for each alliance. Each Team's Driver Station was connected to the SCC both by an Ethernet cable and by a serial cable connected to the DS's Competition Port that controlled the mode of play. Three Scoring Controllers were also connected to each SCC by Cat5 cables to enable referees to score the event in real-time with hand-held devices. The SCC also provided power to the field devices and to the Driver Station for each alliance.

3. The Field Control User Interface was a custom unit located at the scoring table which allowed the Scorer to control the field functions directly including finalization of the match scoring.

4. The Field Radio, a Cisco wireless access point, connected to the FMS via an Ethernet connection to the Scorpion Case, allowed communication with the 6 robots on the field.

5. A separate laptop connected to the Scorpion Case was used to monitor the network and ensure that the robots were powered and their components properly connected.

VI. FIRST Robotics Competition Control System Integration: FMS Robot Communications During Game Play

The Field was controlled by packets [UDP datagrams] sent in both directions between the FMS and the Driver Station. Packet sizes were the smallest size possible to minimize communications overhead and to ensure system reliability. FMS DS packets only used 9 bytes of their 74 byte length sent twice a second by the FMS. As previously described, the Robot State was controlled by a single byte: the software running on the CompactRIO would then enter the appropriate state based on that datum.

The DS FMS communication utilized all 50 bytes, transmitted every 100 ms. Data included robot identification (team number and IP address) and battery voltage, as well as information about the quality of communications (missed packet count and average round trip time). This information was critical to the FRC Technical Advisor who monitored the field and helped debug field functionality issues. A common cause of field problems was a team forgetting to plug their battery in or connecting their robot access point to the CompactRIO, a condition readily monitored by the FMS and FRC Technical Advisor.

VII. Dashboard Feedback

The Dashboard was a separate LabVIEW program which ran on a laptop and communicated with the robot to display the data via a customized graphic display. When ran outside of the Competition Field, the Dashboard program allowed display of video images captured by the AXIS 206 net-camera on the laptop. While the FMS data communications were very sparse, the Dashboard allowed monitoring of a much larger data set, as well as of the video image, which was transmitted separately.

The data set transmitted was user-definable, and allowed programmers to follow values of state variables, as well as sensor outputs. The default data set in the Dashboard template included PWM, relay, and solenoid outputs status, as well as analog and digital sensor inputs, the robot's battery voltage, and Driver Station inputs and outputs. The entire data set was accessible during competition matches, with the exception of the video data. This was done to ensure that the system's bandwidth was not exceeded and that robot function was unimpaired.

Data communication was carried out at 50 Hz (every 20 ms) with a 984 byte maximum. If the data set being transmitted was larger than 984 bytes, it was truncated with the remainder sent in the next packet.

VIII. SYSTEM MONITORING

The FRC Control System utilized two "watchdogs" to ensure that the CompactRIO defaulted into a fail-safe state.

System Watchdog: The system watchdog was always enabled and ensured that both the wireless and CompactRIO system were properly functioning. The system watchdog was programmed into the CompactRIO FPGA that was fed TCP packets through the Ethernet connection (wired or wireless) from the Driver Station to the CompactRIO. This happened by default and required no additional setup from the user beyond the normal IP address setup of the CompactRIO and Driver Station.

If anything disrupted the connection between the Driver Station and the CompactRIO (such as a loss of wireless network connection, or the competition dongle becoming detached from the DS) the FPGA turned off all PWM and actuator outputs. This immediately stopped the robot and prevented a run-away robot. Teams did not have any way to access or modify this watchdog. This watchdog was entirely transparent to the user and ensured correct system operation.

User Watchdog: The user watchdog was an optional feature that teams could elect to include in their program. The user watchdog was provided in the robotics palette for teams that wanted to implement their own watchdog into their code. This user watchdog would be used in addition to the system watchdog. If the User Watchdog was triggered, either by not being fed in time or by manual triggering, all motors and actuators were disabled in an identical manner to the System Watchdog.

IX. HARDWARE MONITORING

If the Watchdog (User or System) was triggered, the FPGA on the CompactRIO was programmed to put the Digital Sidecar Status LED into a blinking mode. This output was also connected to the Robot Status Light which provided a visual monitor of robot status on the field. The Watchdog status also appeared on the DS LCD screen, to assist debugging efforts.

X. ONE TEAM'S PERSPECTIVES: FIRST ROBOTICS TEAM 236 CONTROL SYSTEM

The 2009 FRC game, Lunacy, called for a unique challenge: since the field and required wheels produced a coefficient of friction of 0.05, teams were encouraged to design a robotic motion system that would avoid slipping and provide for maximal traction. An additional challenge was to use the supplied AXIS camera to locate an opponent's trailer and load it up with game pieces.

Based on prior experience with LabVIEW, FRC Team 236 decided to use the LabVIEW programming environment. LabVIEW's ease of use, coupled with a graphical data display, made it easy to get new programmers quickly up to speed.

To address the game challenge, Team 236's mechanical build team designed and fabricated a swerve steering system with independent pods. Each pod consisted of a CIM motor and a fixed ratio transmission held together by a custom CNC machined frame contained within a length of 5 inch diameter PVC. The 2 front and 2 rear pods were separately attached as pairs by a belt anchored to each pod, running over a drive pulley powered by the front and rear steering motors.

The software task was to allow for smooth control of the steering motors. Simple proportional control was used. A Logitech game steering wheel with foot controls for acceleration was chosen as the human control device to input the drive motor speed and steering set-points.

A potentiometer was used to measure the angle of rotation of the pod and provide the process variable to the control algorithm. The control system used these inputs to determine the PWM output signal that controlled the steering motor. This feedback loop was repeated for each steering motor with separate potentiometers measuring each steering angle.

In order to maximize maneuverability in a crowded playing field, the driver could use different combinations of steering alignments. Steering modes used included steering with the front pods only; or front and rear pods in tandem which allowed crab or swerve drive; or alternatively the front and rear pods moving in opposite directions which facilitated the robot's ability to turn in place. Examples of the LabVIEW programming tools are presented in Figure 6.

Traction control was implemented by a low-pass filter which prevented rapid changes in acceleration when the driver depressed the accelerator pedal to its limit. Team drivers actually preferred to use a non-filtered version, as they found it easier to adapt their driving habits to the robot/field interactions than allow a drive-by-wire approach.

XI. Conclusions and Recommendations

The challenge of designing and implementing a new robotic control system over the course of a year was met. Nearly 1,800 FRC teams of mentors and students, both experienced and rookie teams, were able to successfully switch from using a custom-manufactured control unit to an off-the-shelf programmable automation controller, complete with 2 new programming language environments. This was all accomplished within a 6-week window of learning, design and actual construction of a competition robot.

The system development benefited from an open framework that invited a host of dedicated technologists to contribute to the project. This group worked closely with the FIRST Robotics Competition community to understand the customers' user requirements and had the ability to quickly prototype ideas. Beta versions of the hardware and software were shared with evaluation groups from across the country to ensure that the system was meeting their requirements. On-line discussion forums were created and used to collect feedback and share lessons learned on a real time basis. To augment the roll-out of this product to the entire community, regional trainers were created to visit local teams and bring them up to speed on the technology. In addition, a wide variety of web resources were assembled, including video training lessons, to assist the teams with the implementation of the 2009 FIRST Robotics Competition Control System.

The system has shown to be reliable and well suited for robotic control. While some improvements are needed with the driver station, and especially so in its ability to withstand electrostatic discharge, it is anticipated that this control system will evolve as it is used in future robotic competitions.

Acknowledgement

CompactRIO and LabVIEW are trademarks of National Instruments.

References

[1] “FIRST Robotics Competition: University Curriculum Applications of Mobile Robots,” Wilczynski, V., Flowers, W., 2006, *International Journal of Engineering Education*, Vol. 22, No. 4, pp. 792-803.

[2] “Creating Virtual Teams for Engineering Design,” Wilczynski, V., Jennings, J.J., 2003, *International Journal of Engineering Education*, Vol. 19, No. 2, pp. 316-327.

[3] www.ifirobotics.com

[4] FRC Getting Started Guide, National Instruments, 2009, http://decibel.ni.com/content/docs/DOC-2453

[5] FRC 2009 Control System Manual, FIRST Robotics Competition, Chapter 5, p. 6.

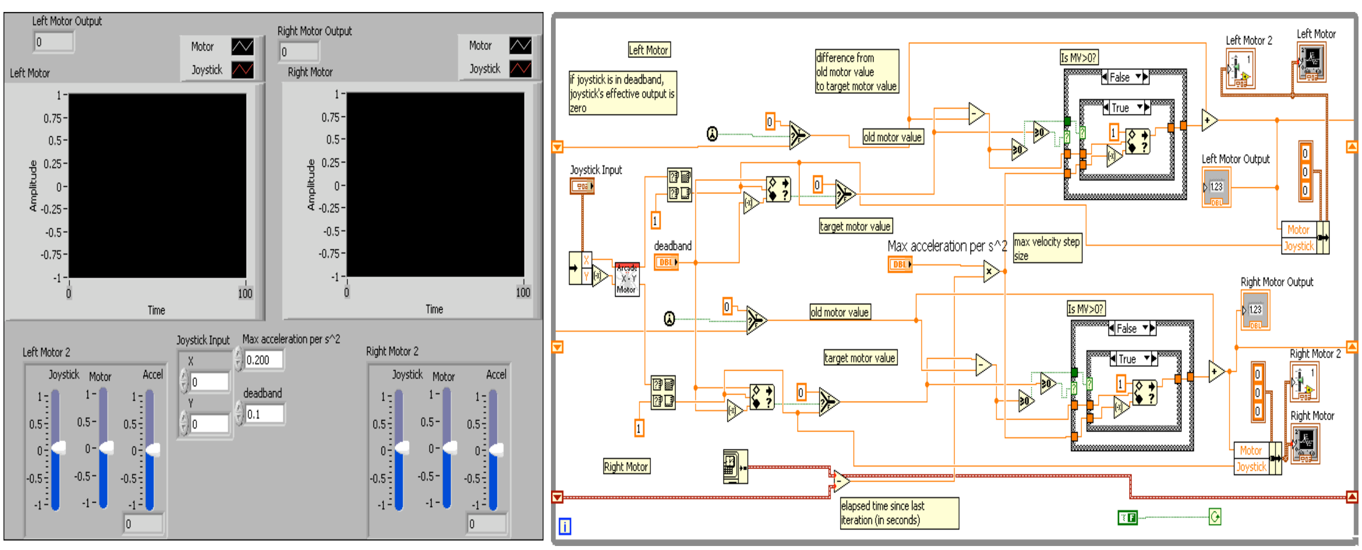

Figure 6 - LabVIEW with FRC Team 236 Programming