LabVIEW Developer Education Day

Drive Your Potential

ni.com/devday

NATIONAL INSTRUMENTS

NI Linux Real-Time

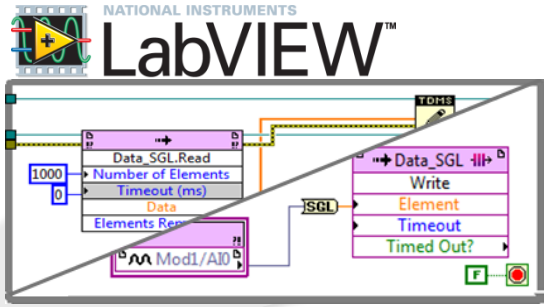
Agenda

1. Hardware Overview

2. Introduction to NI Linux Real-Time OS
   - Background & Core Technology
   - Filesystem
   - Connectivity and Security

3. Advanced New Features
   - C/C++ Support
   - OPKG package manager

The Redesigned CompactRIO System



NI LabVIEW System Design

Program with LabVIEW Real-Time and LabVIEW FPGA modules
Quickly port existing LabVIEW applications

High Performance and Throughput

Dual-Core ARM 667 MHz processor
Xilinx 7 Series FPGA fabric with 85k logic cells
16 DMA FIFO channels for data streaming

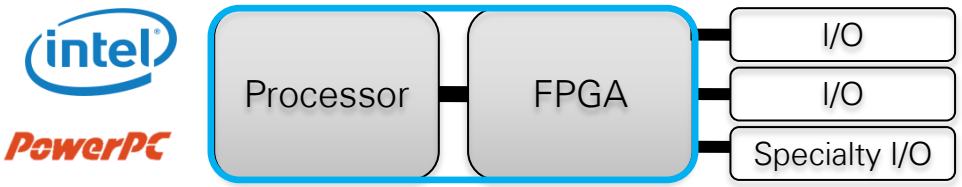
Ultra Rugged

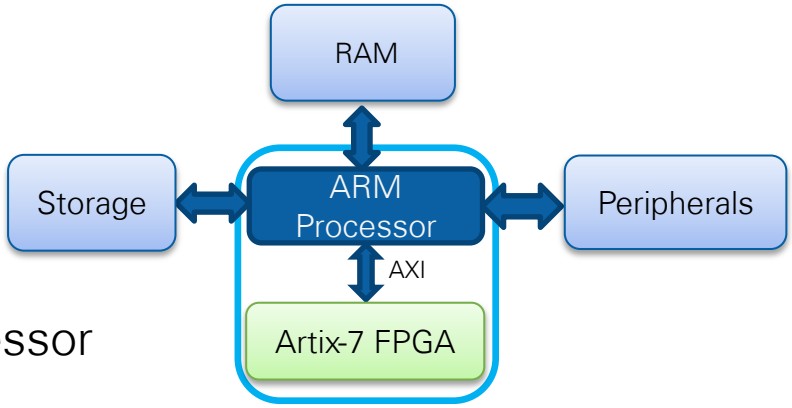
-40 to 70° C operating temperature range
50 g shock and 5 g vibration tolerance

Community and Code Reuse

NI Linux Real-Time Operating System
Integrate existing applications and libraries
Develop, debug, and deploy C/C++ code

New Value CompactRIO Hardware Architecture

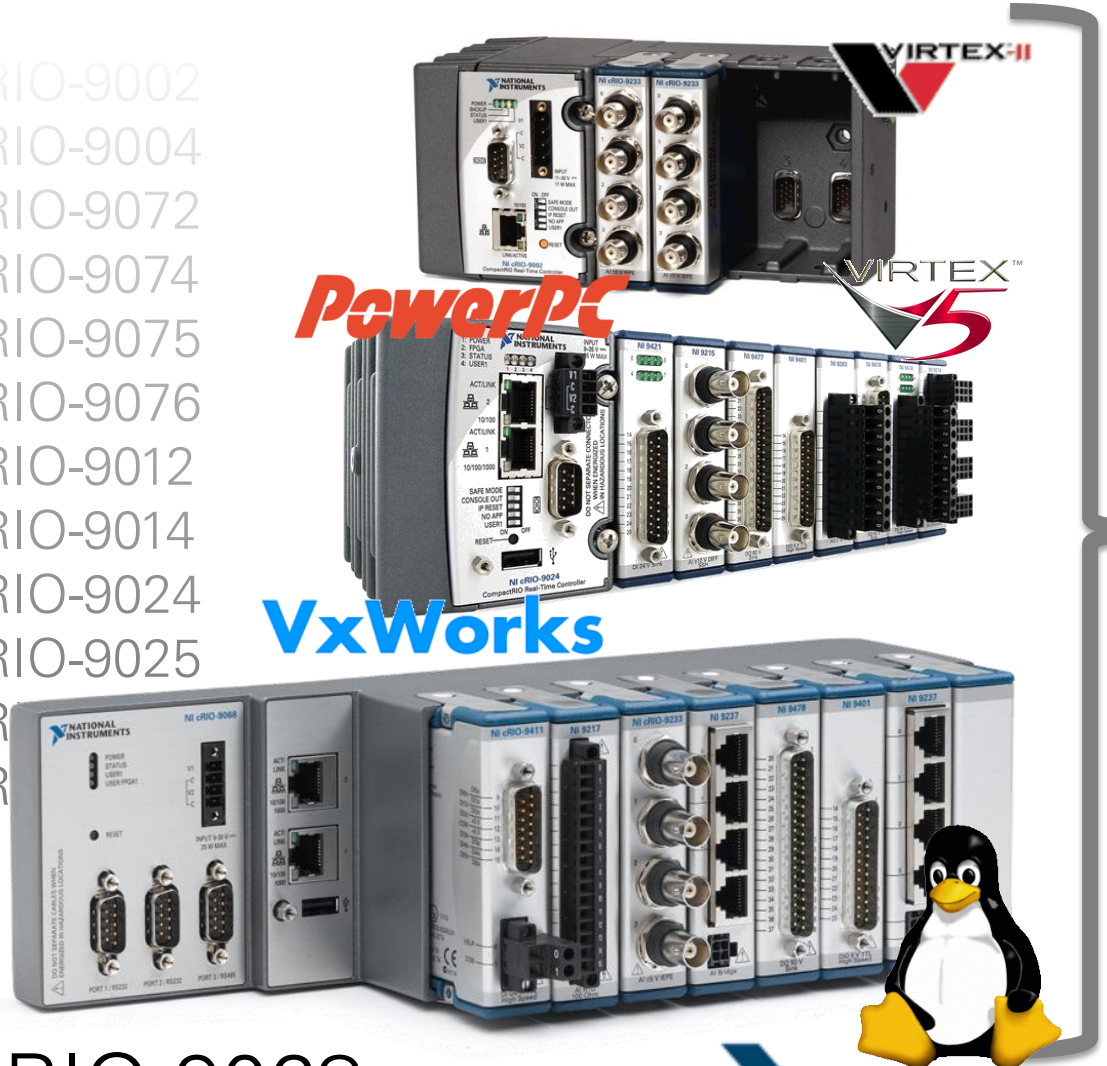




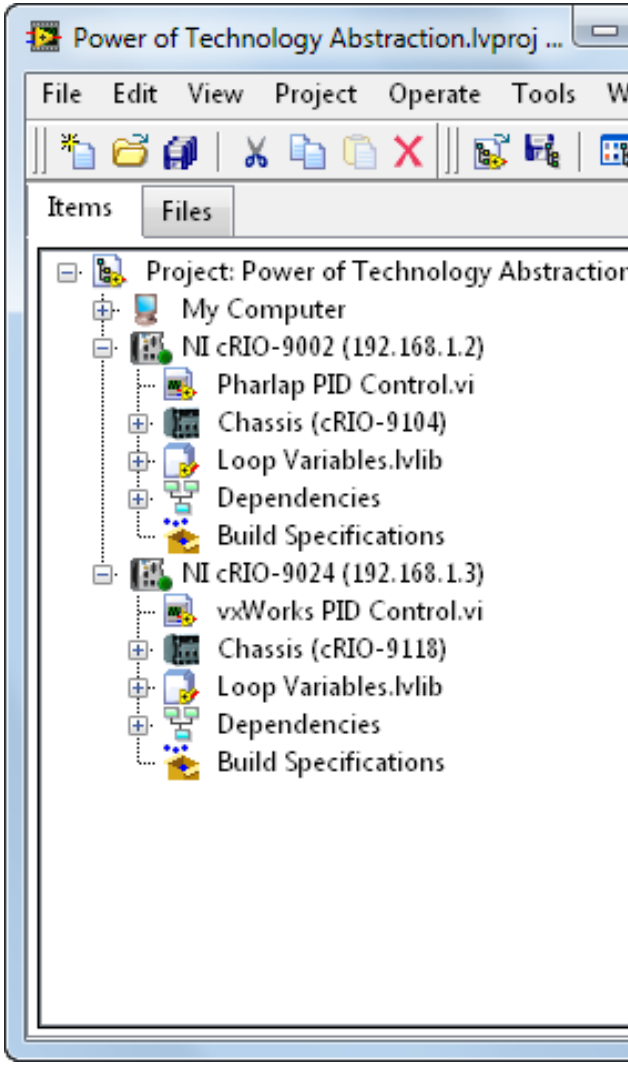
- AXI Bus between FPGA and processor
  - 300 MB/s throughput
- 667 MHz Dual-Core ARM Cortex-A9 Processor
  - Some peripherals routed through FPGA
- Artix-7 FPGA with 85K Logic Cells
  - 220 DSP Slices

LabVIEW Programmed NI CompactRIO

cRIO-9002
cRIO-9004
cRIO-9072
cRIO-9074
cRIO-9075
cRIO-9076
cRIO-9012
cRIO-9014
cRIO-9024
cRIO-9025
cR
cR



cRIO-9068



NATIONAL INSTRUMENTS

"Within 24 hours of receiving a cRIO-9068 controller, we ran our existing LabVIEW application software without any problems."

– Bob Leigh, CEO of LocalGrid™

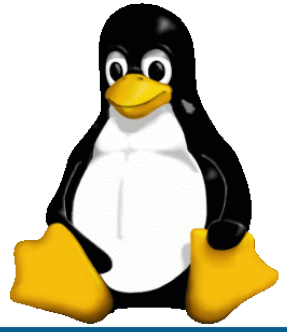
Power Monitoring

LocalGrid

LabVIEW Support for NI Linux Real-Time OS®

LabVIEW 2013 Real-Time Module supports developing, debugging and deploying applications to the NI Linux Real-Time OS® deterministic operating system

- For users familiar with Linux, unlock the vast Linux ecosystem on the new CompactRIO controller
- Reuse C/C++ code in and alongside LabVIEW Real-Time built applications on the redesigned CompactRIO controller
- Freedom in Connectivity
  - Expanded LabVIEW design flow for open web service creation
  - Secure file transfer with WebDAV
  - Improved network interface



NATIONAL INSTRUMENTS

NI Linux Real-Time Operating System

- Why Linux?
  - Support across CPU architectures (ARM, x86, etc)
  - Offers better security
  - Not vendor tied, not proprietary
  - Large ecosystem
- NI investment
  - Reliable, real-time performance
  - Future NI maintenance and management

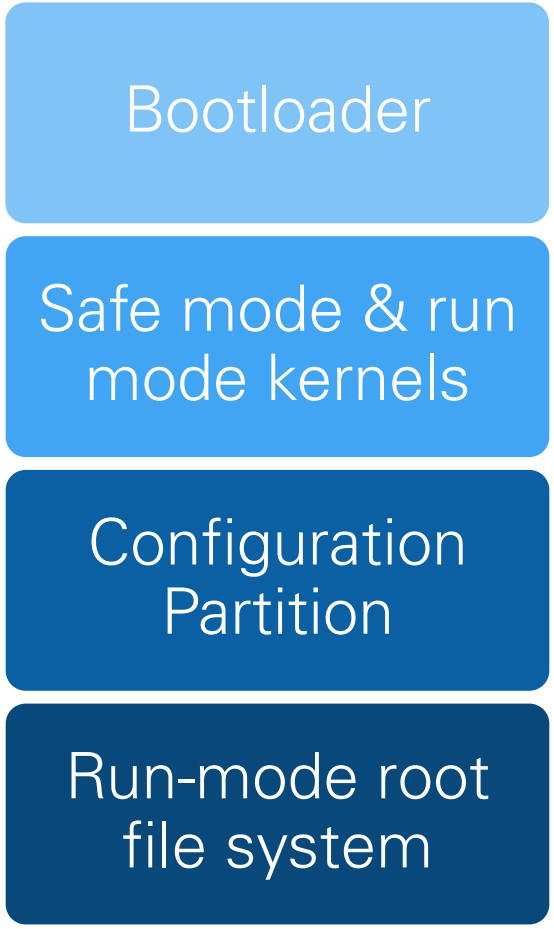
Real-time reliability WITH usability/ecosystem of a general-purpose OS

NI Linux Real-Time Operating System

- PREEMPT_RT
  - Enables pre-emption and priority inheritance
  - Improvements commonly applied to Linux mainline kernel
  - Standard approach to real-time on Linux over last few years
- Dual Mode
  - Increased system resilience and robustness
  - Improved system security
- Scheduler
  - Two schedulers: one for real-time tasks, one for all other tasks
  - Improved system throughput with more efficient scheduling

Boot Modes and Partitions

- Run Mode (read and write)
  - Has access to the root filesystem
  - This is the mode in which LabVIEW runs
- Configuration Partition
  - Holds networking configuration
  - Houses Firewall and VPN settings if set
  - Can be used to communicate between modes
- Safe Mode (read only)
  - Securable with NI-Auth
  - System Web Server is available
- Mode to restore to factory default without RMA



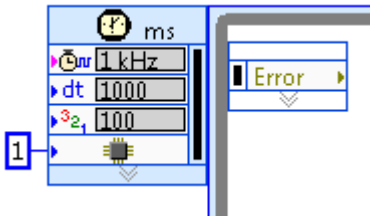
Multicore Performance with the cRIO-9068

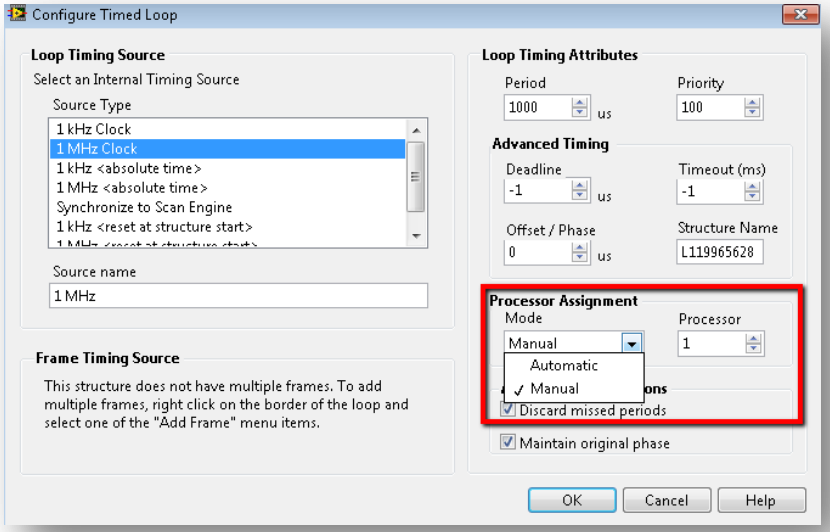
- Control Applications
  - One core for critical code, rest on other core
  - Introduction to LabVIEW Real-Time Symmetric Multiprocessing (SMP)
- High-end processing
  - Load balancing across cores
  - Specifying the Set of CPUs Available for Automatic Load Balancing in LabVIEW Real-Time
- Streaming applications
  - Producer/Consumer architectures

- Take advantage of LabVIEW to more intuitively approach multicore programming

Optimizations for Multicore Programming

- Set Processor Affinity
  - Time critical code on one core
  - Normal priority on another core



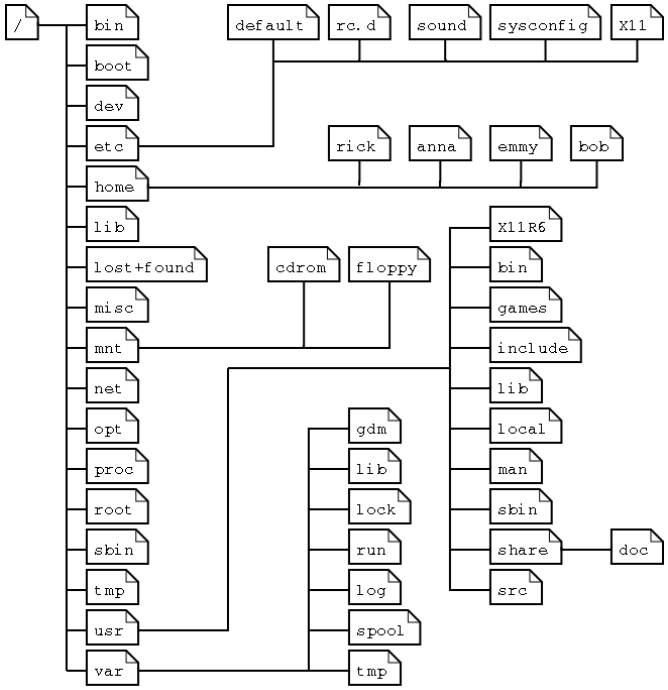


- Best Practices reminder: Avoid 100% CPU use with real-time priority for extended periods of time
  - Lower priority OS threads need access every so often for housekeeping
  - Can affect system performance if housekeeping is starved

Filesystem & Connectivity

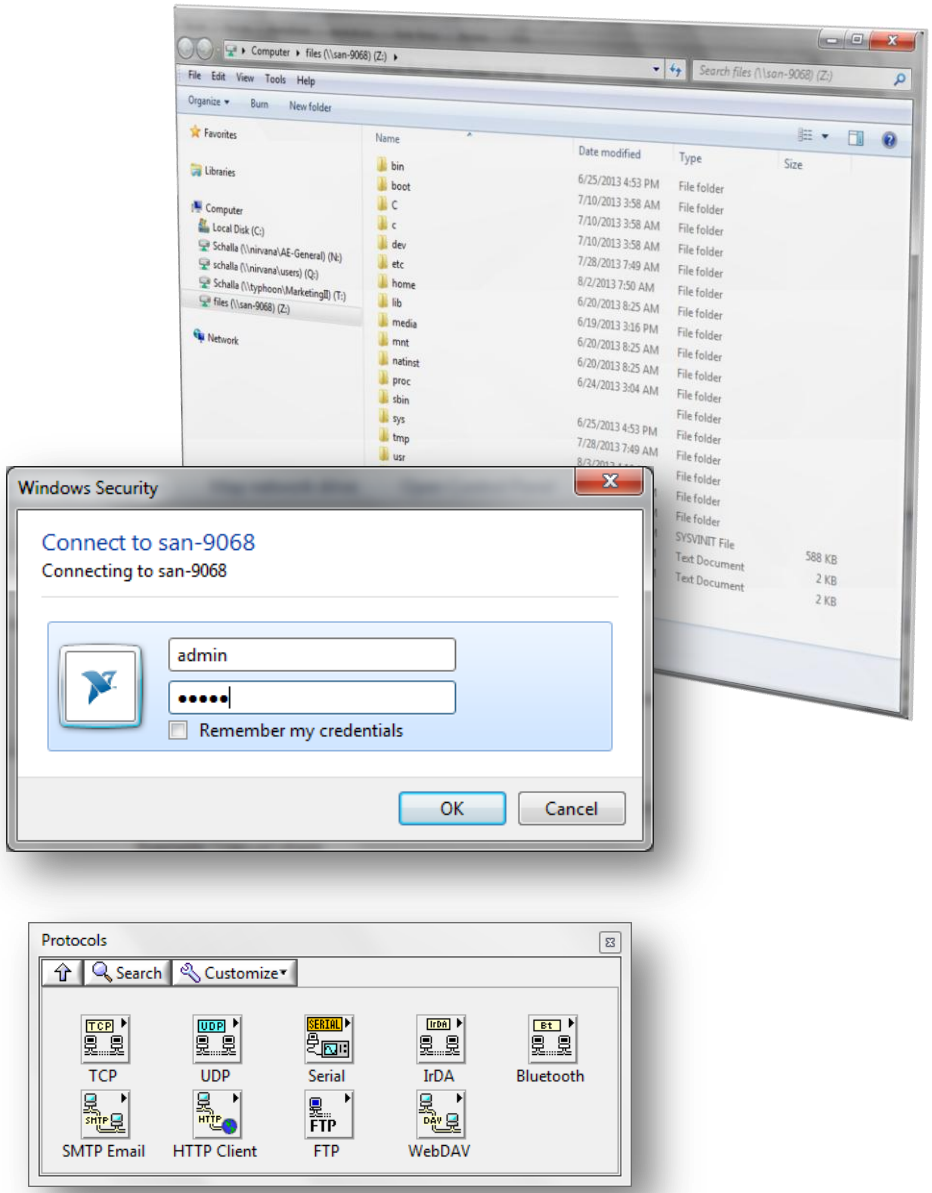
Filesystem on NI Linux Real-Time

- Compressed Filesystem (UBIFS)
  - Can store more data locally on device
- /tmp clears on reboot
  - RAMDisk, 64MB max size
- Filepath changes
  - Unix style
  - Locations of system files have changed



File Transfer: WebDAV

- Industry Standard Protocol
- Manage files on targets remotely over HTTP
- Secure File Access
  - Authentication
  - Encryption
- Supported by all OSes and Web Browsers
- New LabVIEW API for programmatic access



File Transfer: Unsecure FTP

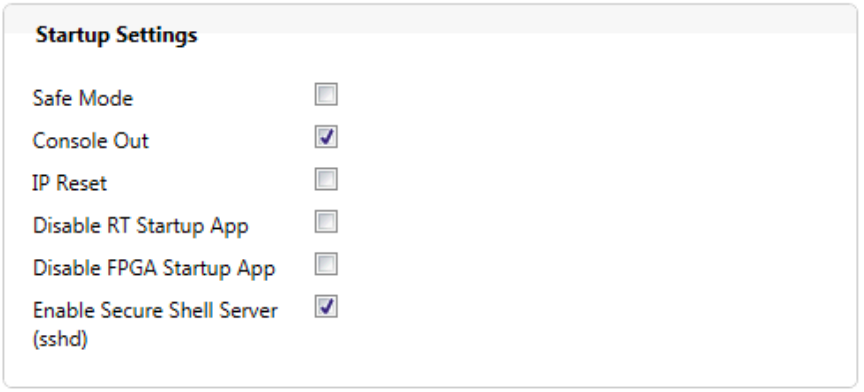
- No unsecure FTP server installed by default on NI Linux Real-Time systems

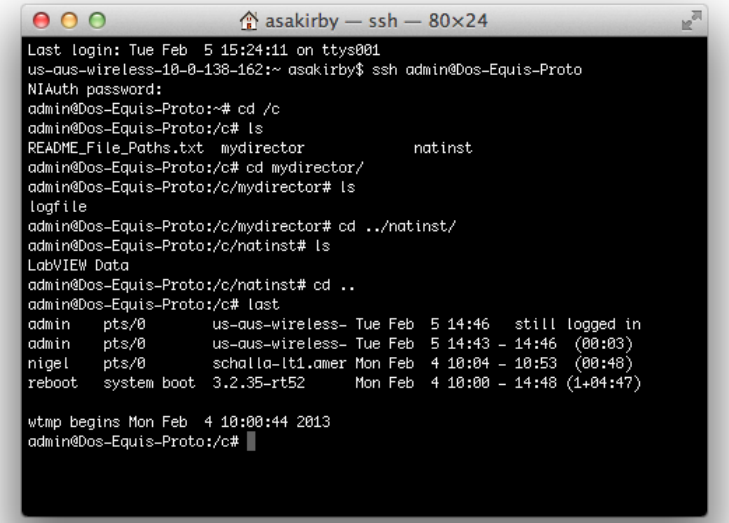
- Unsecure FTP server can be manually installed for compatibility
  - Must be accessed as the 'anonymous' user
  - Has root privileges similar to current cRIO

Not Recommended

Secure Shell (SSH)

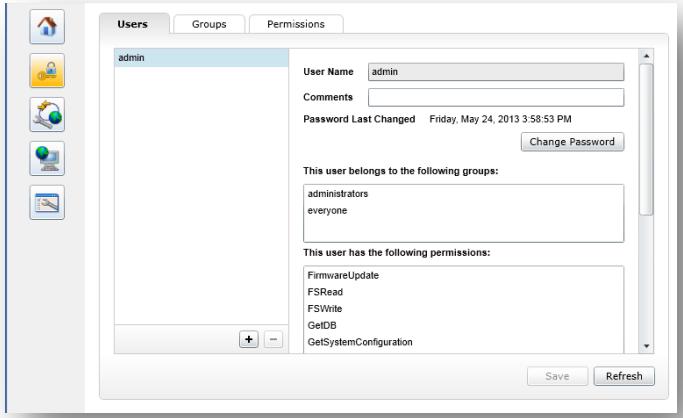
- Enable through MAX and/or Web Interface
- Can be used as a console
- Can be used to transfer files
  - Permissions based on login
  - SFTP
- Credentials synchronized with NI-Auth (Web Interface)





NI-Auth and NI Linux Real-Time Integration

- All user authentication goes through NI-Auth
- Use the Web Interface to manage users
- PAM Integration
  - Users in NI-Auth are users in Linux
  - admin user is superuser
- If admin password is lost, target must be reset to factory default
  - Must contact NI



Demo

CONNECTIVITY AND SECURITY ON NI LINUX REAL-TIME

Security on NI Linux Real-Time

- SSL enabled by default
  - Can programmatically install software over SSL
  - Can use public keys for SSH
- HTTPS only communication possible
  - Can turn off HTTP version of the System Web Server
- IPTables* available for setting up a firewall

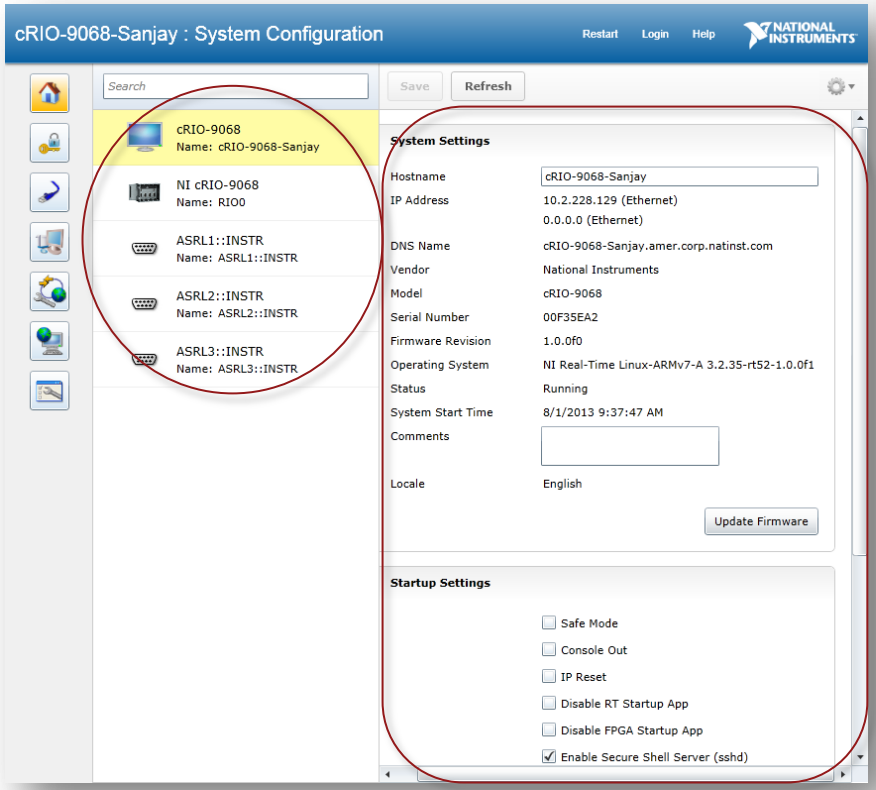
- OpenVPN* available for setting up a VPN

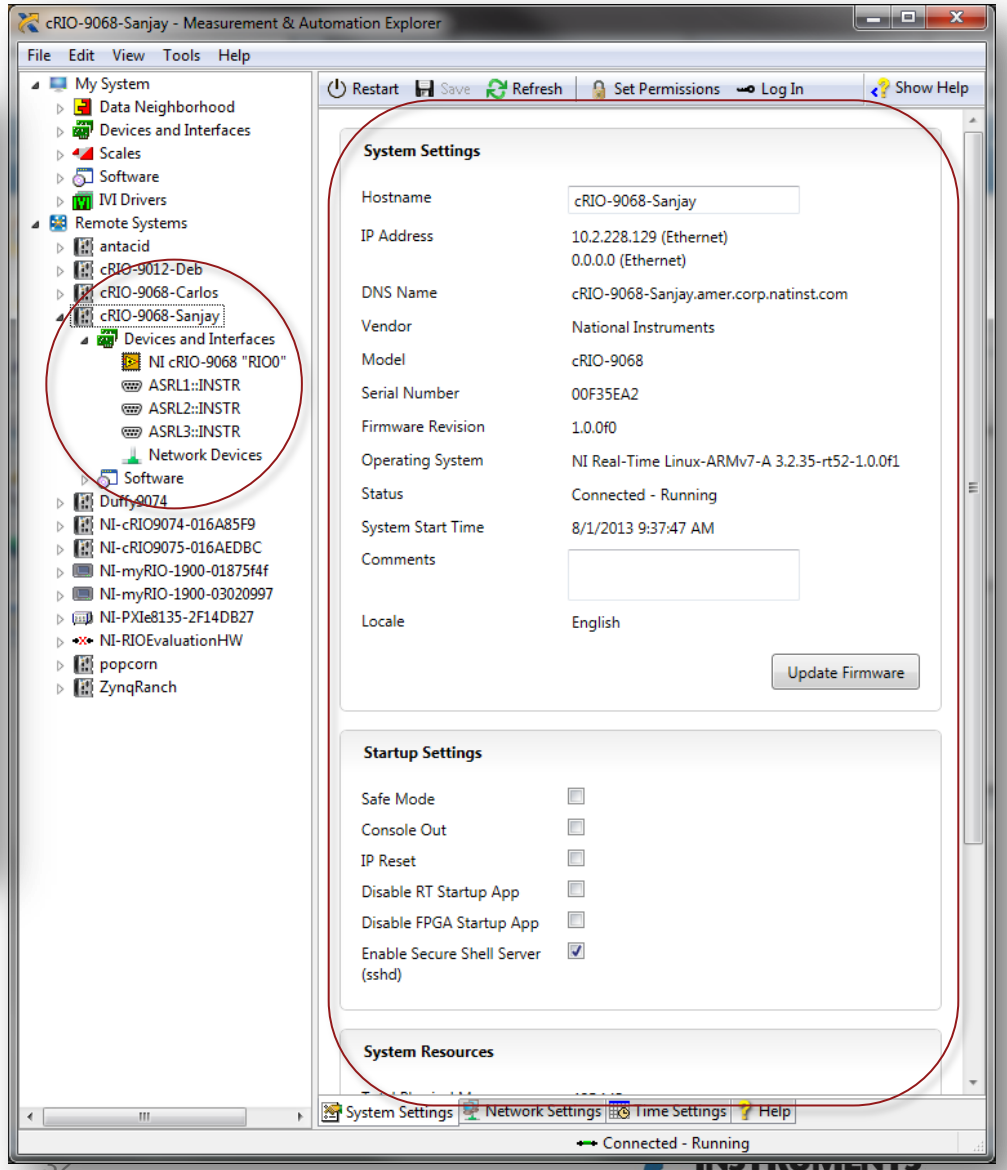
OPENVPN™

*Not supported by Applications Engineering. Requires experience. No LabVIEW API

III. Configuration and Deployment

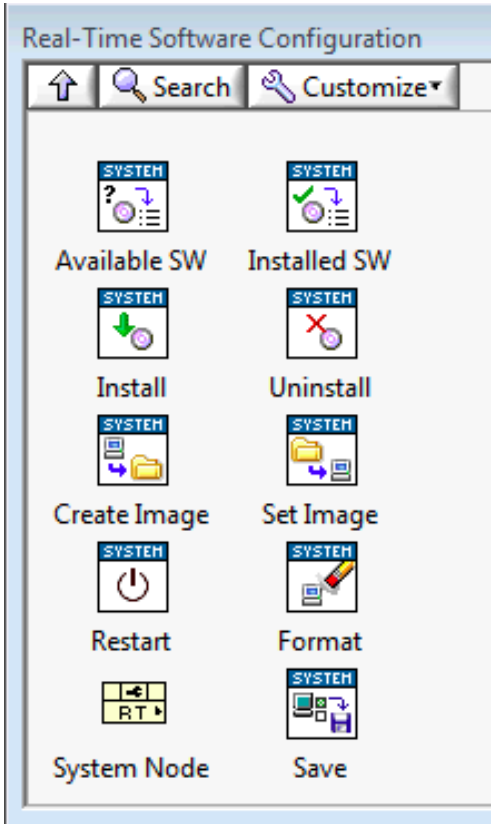
New Consistent Right-Hand View in MAX and Web Interface





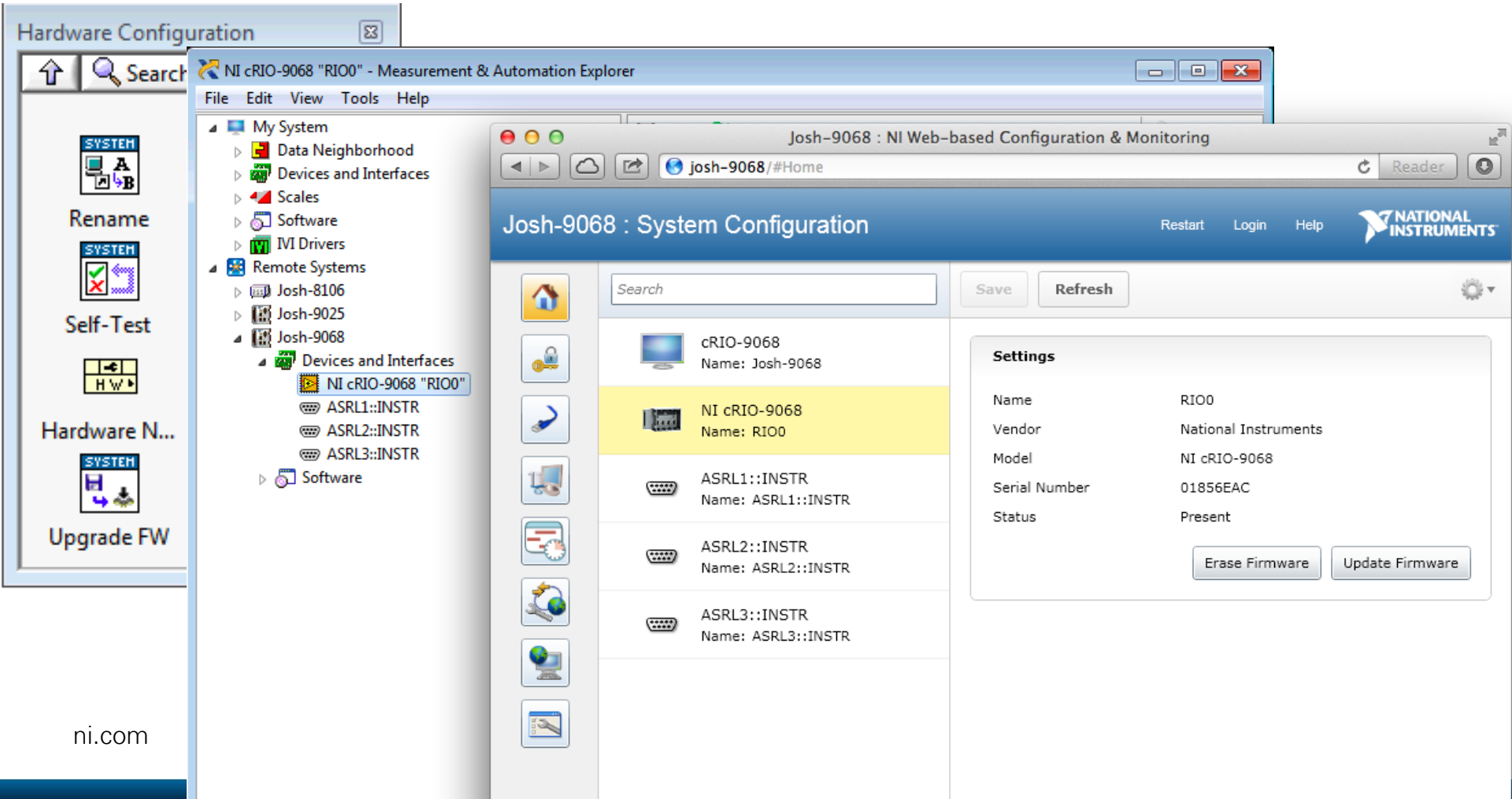
System Updates on NI Linux Real-Time

- NI Linux Real-Time targets can directly call "Set Image"
  - Enables targets to reimage themselves
  - Images can be pulled down from the network or stored on a USB drive
- Specify additional metadata when creating an RT image (title, version, description)
- Blacklist Wildcards with System Imaging*
  - Globbing: "*" and "?"
  - Character set matches: [abc]



Manage FPGA Bit Files

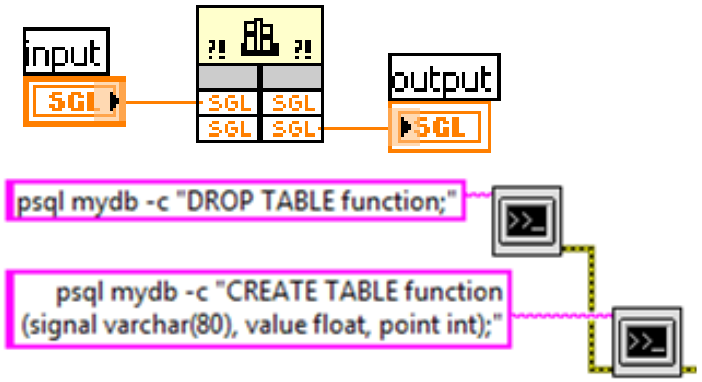
- Update and erase the FPGA bit files on NI Linux Real-Time targets programmatically, from MAX, and the web



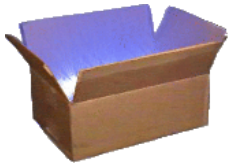
Advanced New Features

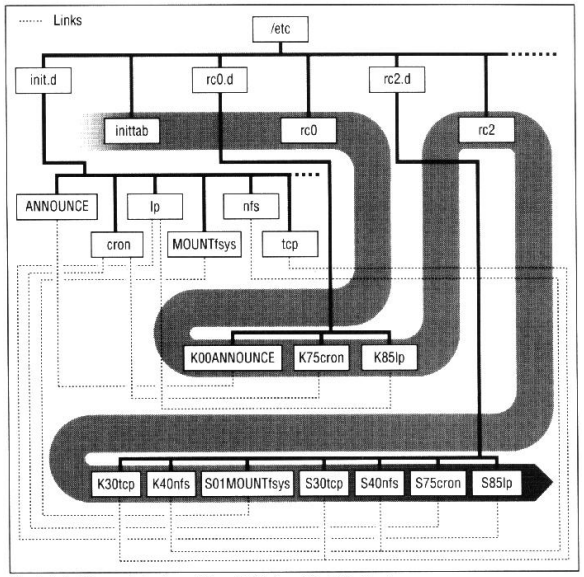
Interacting with Code on NI Linux Real-Time

- To/From LabVIEW
  1. Call Library Function Node
  2. System Execution calls
  3. Localhost communication



- Beyond LabVIEW
  1. Init scripts – for initializing at startup
  2. Cron – for periodic execution
  3. Network enabled SSH programs





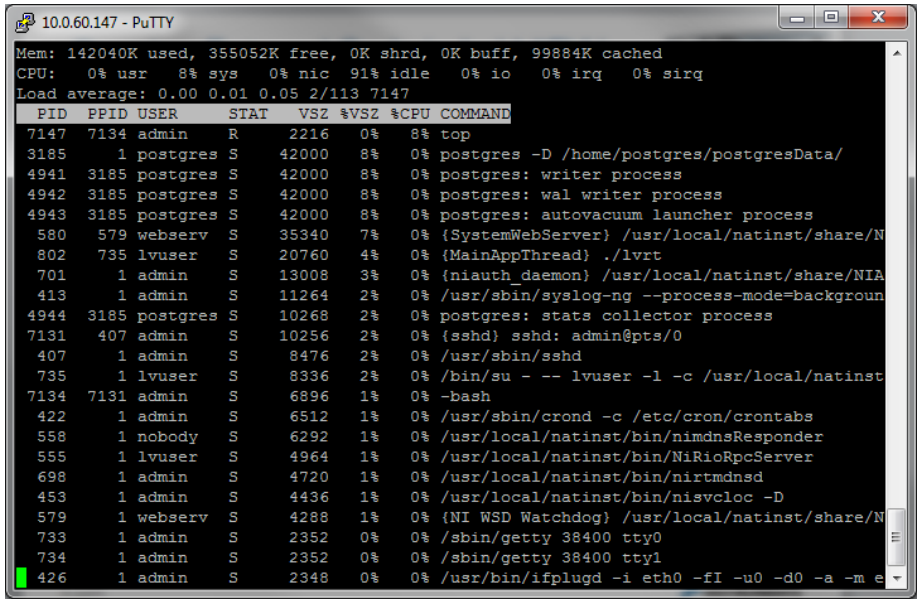
Reprinted with permission from, "Essential System Administration,"
copyright 1991 by Aeleen Frisch O'Reilly & Associates, ISBN # 0-937175-80-3.
For orders & information, call 800-998-9938.

NATIONAL INSTRUMENTS

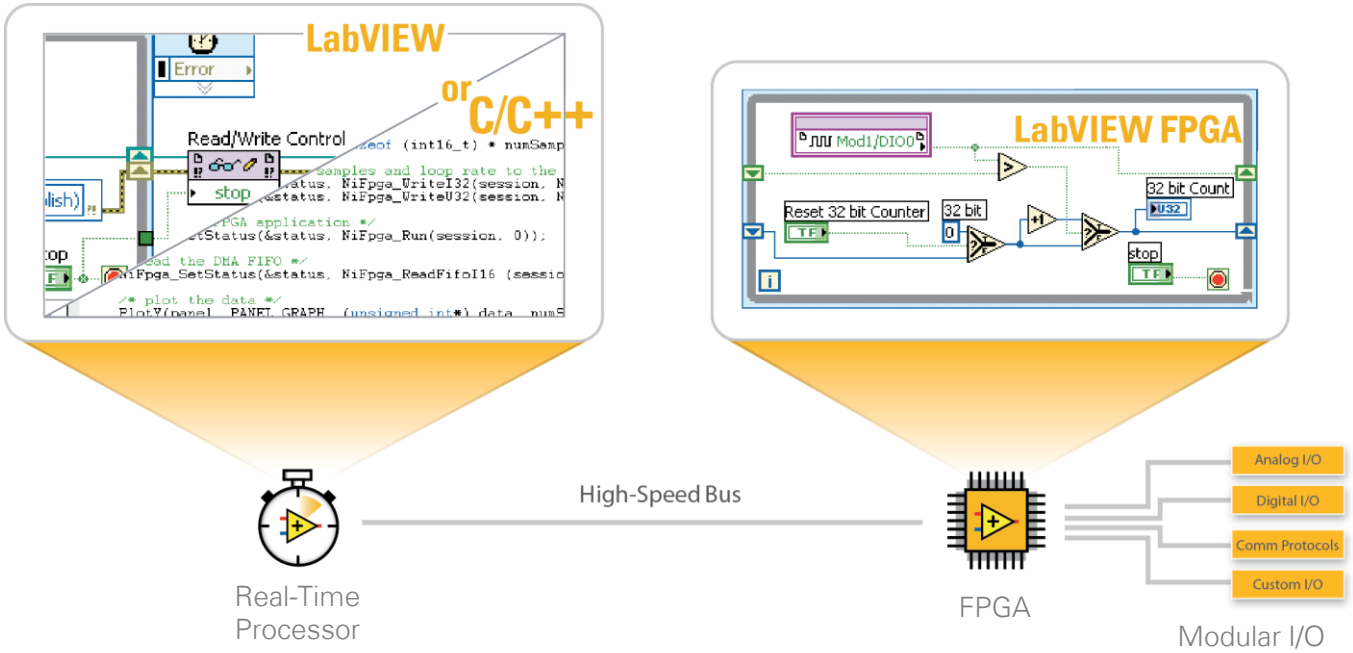
Quality of Life Improvements

- Time Zone changes don't need reboot
- Improved System Logging
  - Increased traceability on Linux – ex: syslog

- Common Linux utilities
  - top, ps, netstat, etc.
- Memory use
  - Don't have to track contiguous memory
  - Virtual (Paged) memory on NI Linux Real-Time



Flexible Software Integration



Code Reuse

- Integrate existing applications and libraries
- Develop, debug and deploy C/C++ code
- Use Eclipse or IDE of choice
- Leverage the Linux ecosystem
- Interoperate with LabVIEW-programmed FPGA

Programmable Hardware

- Offload critical, decision-making code to the FPGA
- Reliable, precision timing for control or processing
- Achieve high-speed, high-accuracy I/O
- Use graphical programming to leverage FPGA technology without HDL expertise

What is Eclipse?

- Free IDE Framework, originally for Java development
- De facto Standard IDE for Embedded Development
- Updated with tools for C/C++ development and debugging
- Eclipse C/C++ Development Toolkit (CDT)
- Target Management/Remote System Explorer

- Note: NI Provided Installer is for Windows
- It is not necessary to use Eclipse

Known Limitations of the FPGA Interface C API

- Generic RIO API
  - Module specific methods are not supported (not many of these)
- Limited data type support, no support for
  - Floating point
  - Fixed point
  - Clusters
  - Arrays of any data type other than supported scalar types
- To use a different FPGA VI, user must regenerate the header and recompile the application
  - Challenging to create application that works with multiple bitfiles

Eclipse Experience for C/C++ on NI Linux Real-Time

LabVIEW FPGA Development
(Windows)

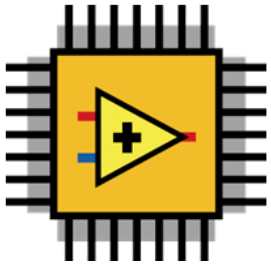
Remote Development
And Debug
Deploy

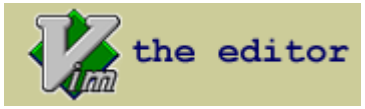
CompactRIO/myRIO

1. Develop LabVIEW FPGA VI, compile bitfile & generate C API.
2. Develop and build C/C++ application with generated C API & debug.
3. Deploy built application and bitfile and run.

Elements to the C/C++ Development Option

- LabVIEW FPGA
  - FPGA Interface C API





- C/C++ IDE
  - Develop, debug, deploy
- Cross-compiler for cRIO-9068
  - armv7a compatible compiler
  - NI Provides: Mentor Graphics Sourcery G++ Lite 2010.09-50 for ARM GNU/Linux (GCC 4.4.1)

Demo

ECLIPSE ‘LED’ DEMO

Accessing the Linux Community: Package Manager

- What is a package manager?
  - Set of software tools for installing, updating, configuring and removing applications
  - Think of the App Store, VIPM, etc.
- CompactRIO-9068 shipping with a package manger pre-installed called “opkg”
  - Does not take the place of MAX
- Access to hundreds of applications in the Linux community

Demo

OPKG DEMO

Leveraging the Linux Community

- Enable OPKG access to repositories

- Download and configure applications as necessary

- Rely on the System Execution VI, Call Library Function Node, or localhost communication to interact

Databases	Security	Code Reuse	Connectivity
Raima	IPTables	C/C++	NTP
MySQL	OpenVPN	Shell Scripts	SNMP
PostgreSQL	fail2ban	Python	IPv6

"By leveraging the open Linux-based real-time operating system on a commercial off-the-shelf controller, we could port our existing Linux software components in a very short time. That saved us at least four months of development time."

- Wolfram Koerver, executive director of S.E.A.

Structural Monitoring

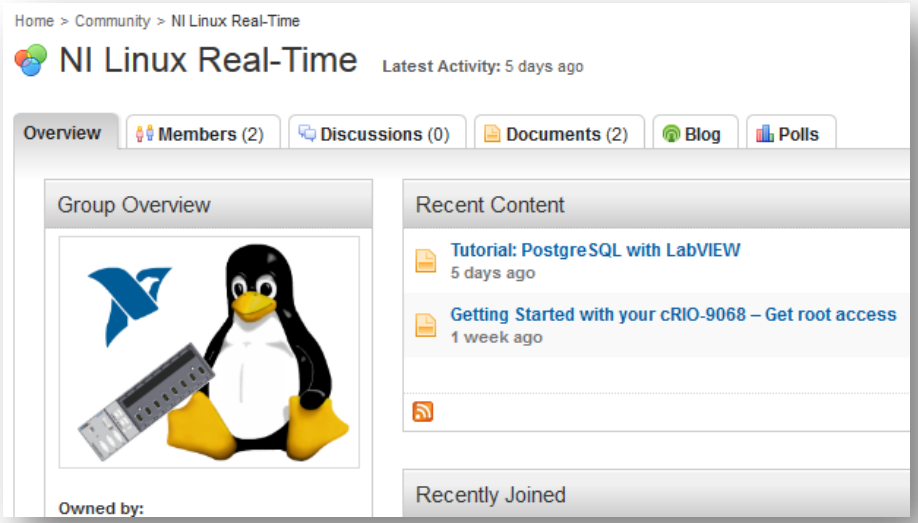
S.E.A

Support Policy

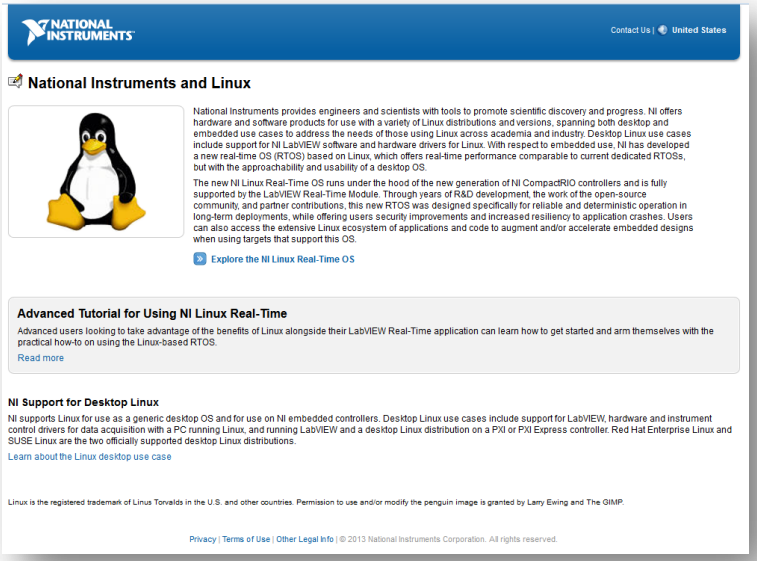
- Limited user mode support
  - Equivalent to 3rd party C code with Call Library Nodes on Desktop
- No support for kernel mode changes
  - You can get the NI Linux Real-Time open source kernel by emailing licensing@ni.com
- Feel free to innovate and explore
  - Can restore to factory default state without having to RMA

Key Resources

- Linux Real-Time Community
  - ni.com/linuxrtforum
  - Tutorials
  - Documentation
  - Forum for discussions



- ni.com/linux
  - Links to whitepapers
  - Embedded and Desktop uses



Questions?

How would you rate 'NI Linux Real-Time: Revolutionizing Embedded System Design'?

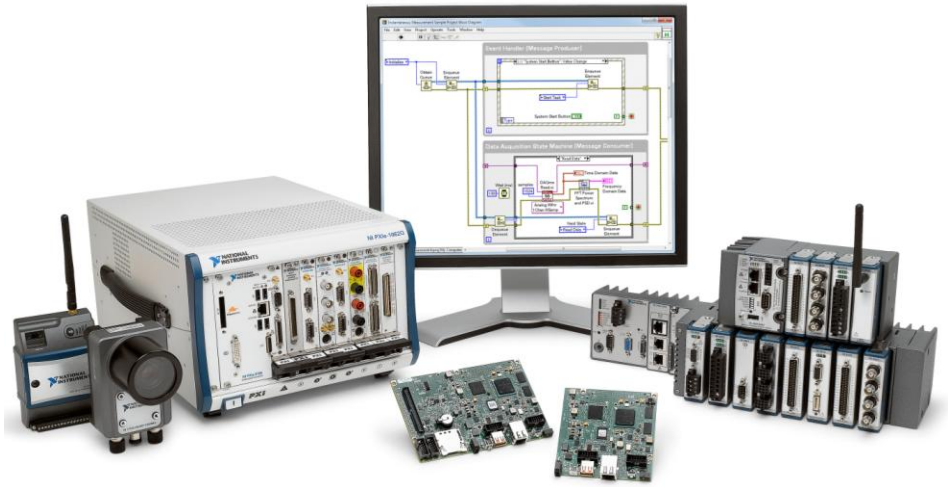
Text a **CODE** to 22333

Excellent	**790065**
Good	**790066**
Okay	**790068**
Poor	**790069**

Get your Embedded Certification!

National Instruments is now offers the Certified LabVIEW Embedded Developer Exam

Learn more **ni.com/cled**



Email certification@ni.com
To Schedule Your Exam