



2015 Controls Introduction

Mike Anderson
(robot_maker12@verizon.net)



Herndon High School
FRC Team #116

What We'll Talk About

- Goals
- Why switch controls?
- The RoboRIO Controller
- Peripherals
- CAN bus
- Resources
- Summary

Goals

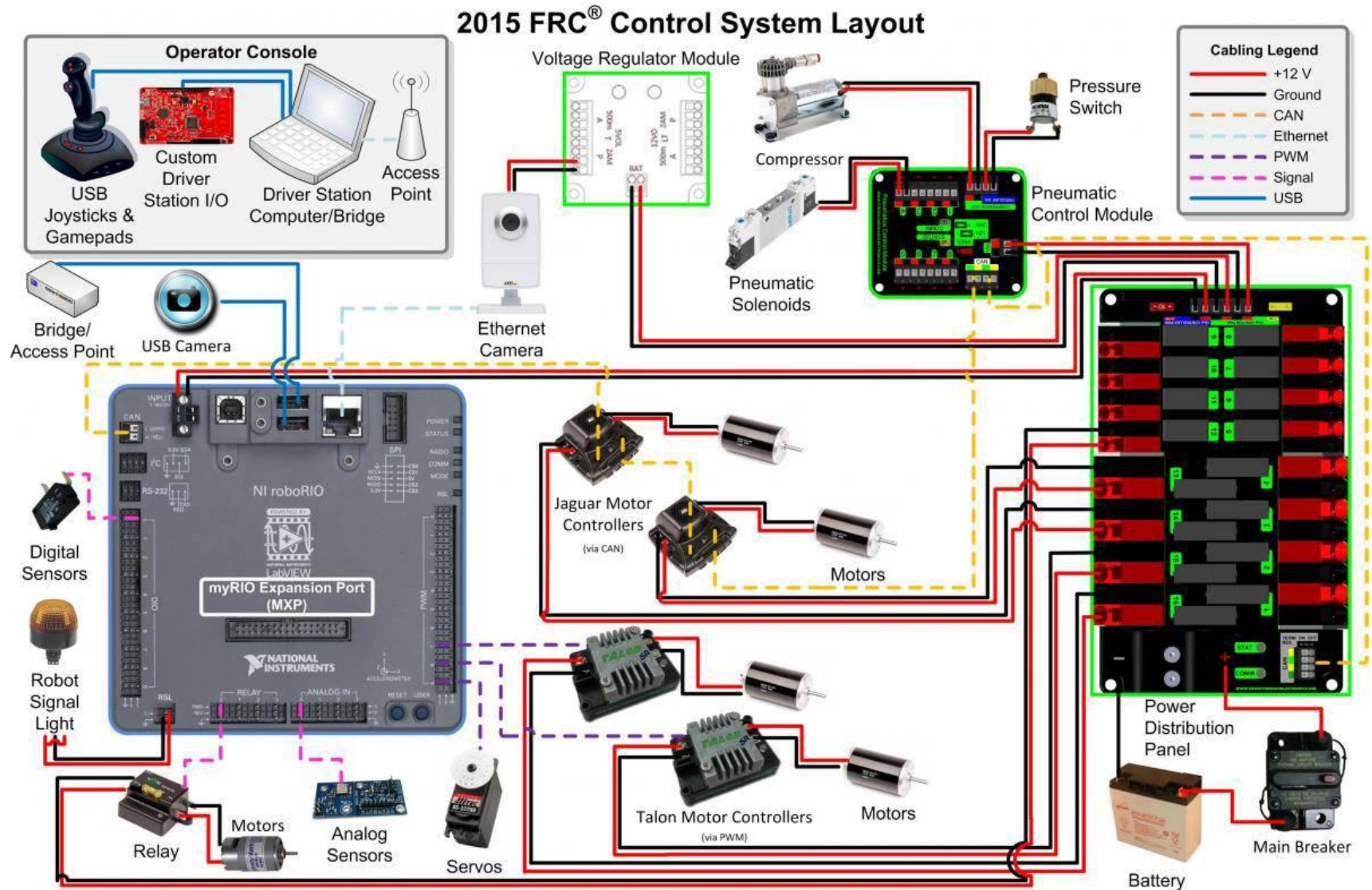
- The goal of this presentation is to acquaint you with the new 2015 control hardware
- We clearly can't explain all of the aspects because we only have 75 minutes
- But, you should leave here with a better understanding of the changes from 2014
- This information is accurate as of the beta units
 - ▶ The software will likely change a bit prior to kickoff

Why Change the Controls?

- The cRIO was getting very long in tooth
 - ▶ 400 MHz PPC running VxWorks™
- Many teams had started using BBBs, Rpi and Arduinos to supplement the sensor and vision processing
- The chassis had become a limitation
 - ▶ The number of slots and bus architecture became a serious bottleneck
 - ▶ Weight was also an issue
- The cRIO is an industrial device that is expensive to build (and buy)
 - ▶ Limits the number that the average team could afford



New 2015 Control System

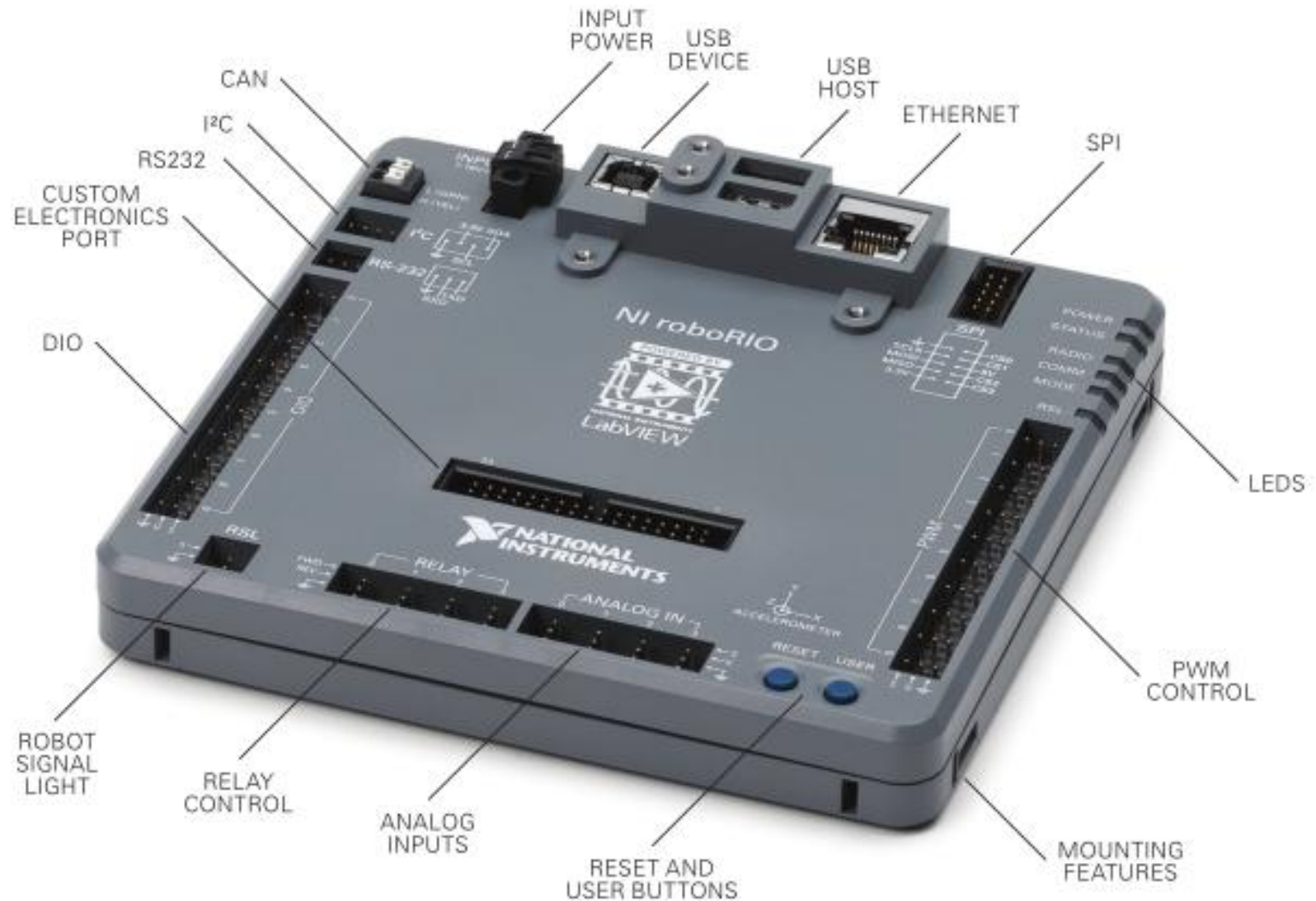


The RoboRIO

- An ARM-based single board computer that increases performance and combines the digital side car into a smaller and lighter platform
 - ▶ Dual-core, 667 MHz ARM Cortex A9 with:
 - 256 MBs RAM (232 MBs usable)
 - 512 MBs flash (386 MBs usable)
 - Xilinx Zync-7020 FPGA
- Running NI RT-Linux
 - ▶ 3.2.35-rt52 Linux kernel
- File system is derived from Yocto project
 - ▶ Uses the same files as the ARM Angstrom/Poky distribution
 - ▶ Special .ipk format packages that use “opkg” package manager

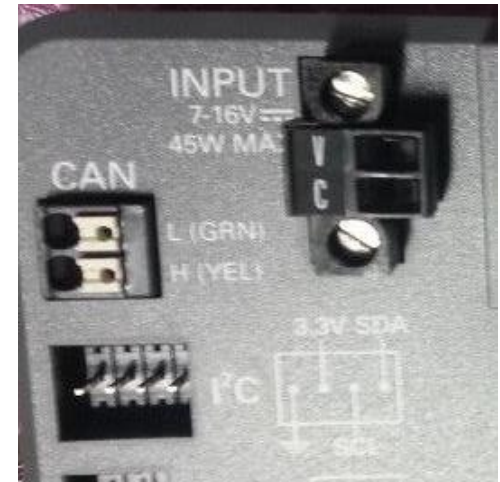


Annotated RoboRIO



Power-Related Info

- The RoboRIO requires 7-16VDC
 - ▶ Max current 45W
 - ▶ Idle current 5W
- Most of the signals are 5V tolerant
- Voltages are:
 - 3.3V (max 1.225A)
 - 5V (max 1A)
 - 6V (max 2.2A)
 - 7-16V (120mA)
- Beware! The UART is 5V EIA RS232
 - ▶ Ready to plug into a PC
 - ▶ Do not use level shifters on the UART or the magic blue smoke will escape!
 - ▶ Do not plug directly into BBB, Rpi or Arduinos either

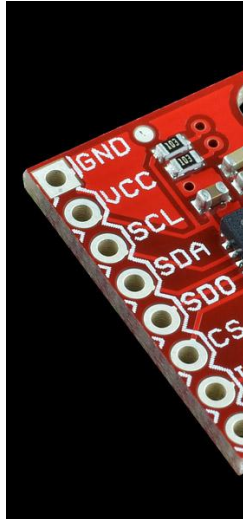


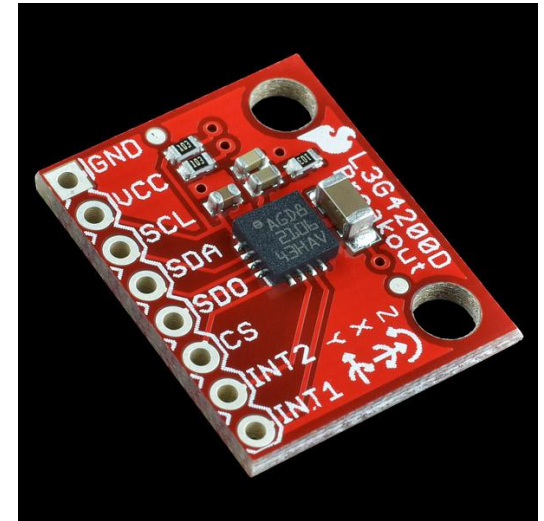
RoboRIO MXP Pin-out

- The *MyRIO* Expansion Port allows for additional I/O opportunities
 - ▶ Gives expansion beyond what we saw with digital side car

DIO 15 / I2C SDA	34	33	+3.3V
DIO 14 / I2C SCL	32	31	DIO 10 / PWM6
DGND	30	29	DIO 9 / PWM5
DGND	28	27	DIO 8 / PWM4
DIO 13 / PWM9	26	25	DIO 7 / SPI MOSI
DGND	24	23	DIO 6 / SPI MISO
DIO 12 / PWM8	22	21	DIO 5 / SPI CLK
DGND	20	19	DIO 4 / SPI CS
DIO 11 / PWM7	18	17	DIO 3 / PWM3
DGND	16	15	DIO 2 / PWM2
UART.TX	14	13	DIO 1 / PWM1
DGND	12	11	DIO 0 / PWM0
UART.RX	10	9	AI3
DGND	8	7	AI2
AGND	6	5	AI1
AO1	4	3	AI0
AO0	2	1	+5V

Digital I/O

- The main RoboRIO has:
 - ▶ 10 DIO lines (each can be programmed as input or output)
 - 20ns minimum pulse width
 - ▶ 1 I2C (1 SDA and 1 CLK)
 - 3.3V
 - 400KHz max frequency
 - ▶ 1 SPI bus (up to 4 devices)
 - 4 MHz max frequency
 - Logic level:
 - ▶ 5V-compatible LVTTL input
 - ▶ 3.3V LVTTL output
 - MXC has
 - ▶ 16 additional DIOs
 - Some pins can be used as aux I2C and SPI
 - ▶ 4 analog inputs
 - ▶ 2 analog outputs
 - ▶ 1 UART
- 



PWM and Relay Lines

- 10 PWM channels
 - ▶ Output only
 - ▶ 15mA max output current
 - ▶ 330 ohm resistor in series
- 4 relay channels
 - ▶ 4 forward, 4 reverse
 - ▶ 5V output
 - ▶ 7.5mA max current
 - ▶ 680 ohm resistor in series
- Max frequency 150 KHz
- Output High Voltage: 4.75V-5.25V max
- Output Low Voltage: 0.0V-0.25V max

Analog I/O

- Analog input:
 - ▶ 500 kS/s @ 12-bit resolution
 - ▶ +/- 16V overvoltage protection
 - ▶ 500k ohm input impedance @ 500 kS/s
- Analog output:
 - ▶ 345 kS/s @ 12-bit resolution
 - ▶ +/- 16V overvoltage protection
 - ▶ 0-5V output range
 - ▶ 50 mV accuracy
 - ▶ 3mA current drive

Onboard 3-axis Accelerometer

- +/- 8G range
- 12-bit resolution
- 800 S/s
- Very little information available during the beta cycle about programming

Built-In Accelerometer

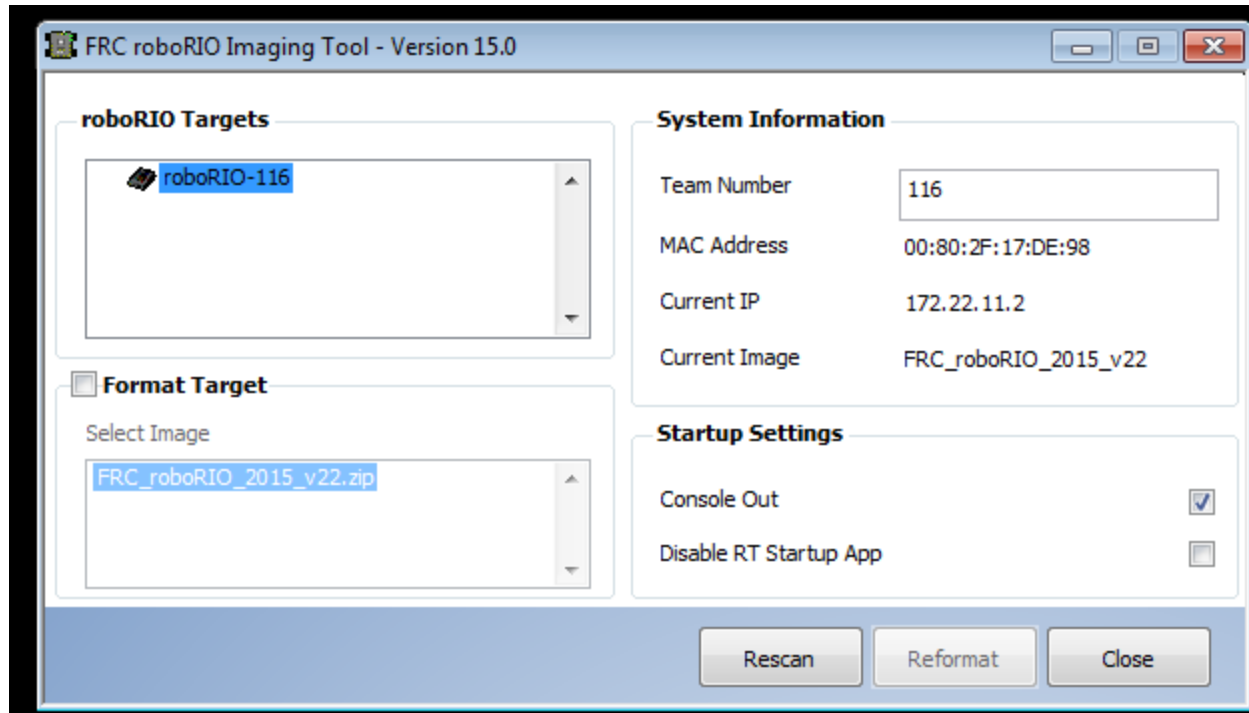
Information about the Built-in accelerometer and class should go here

Accelerometer interface

Information about using the generic Accelerometer interface should go here.

Imaging of RoboRIO

- There is a new imaging tool
- Same image for C/C++, Java and LabVIEW
 - ▶ LabVIEW enable option in web interface

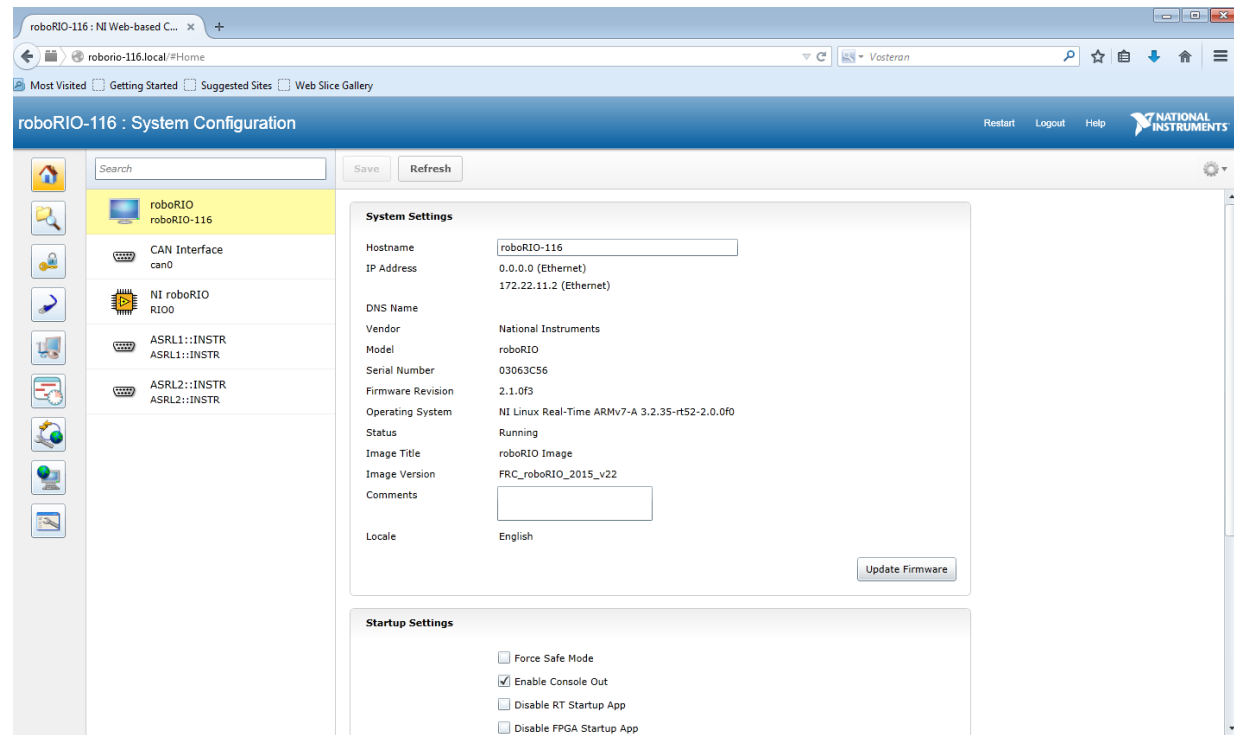


Imaging RoboRIO #2

- Imaging should use the USB device (type B) interface
 - ▶ This sets up a USB IP point-to-point network with 172.22.11.x network link
 - ▶ Do not use Ethernet to image the RoboRIO
- mDNS addressing will work at this point
- Ethernet interface will DHCP an address from the wireless interface

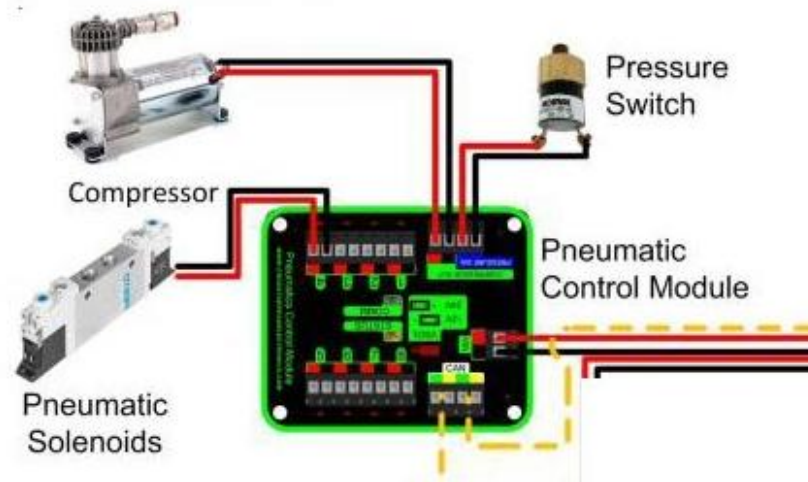
New RoboRIO Web Server

- New interface for RoboRIO
 - ▶ Used to load new firmware
- Requires Microsoft Silverlight
- Addressing is now done via mDNS
 - ▶ roborio-<team #>.local
- Option for enabling ssh server



Pneumatics Control Module (PCM)

- CAN-controlled
- Supports more than 1 PCM
- Closed-loop operation
- Jumper selectable 12V or 24V solenoid operation



Voltage Regulator Module

- Regulated 5V and 12V
 - ▶ Both 500mA and 2A
- Great for powering Wi-Fi access point
- Good brown-out capability



Power Distribution Panel

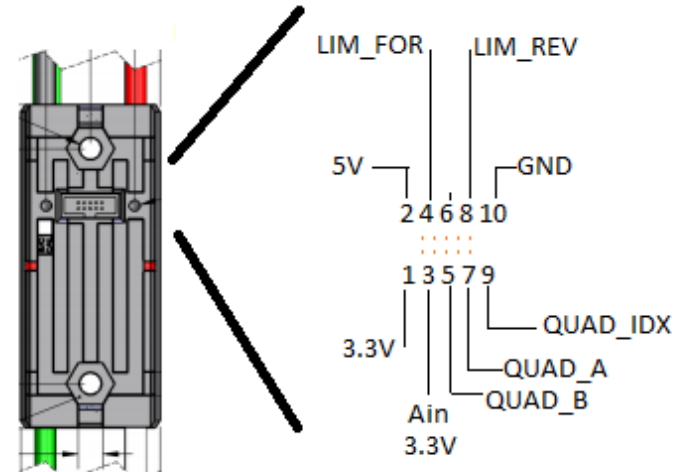
- PDP is smaller than 2014 unit
- Dedicated outputs for PCM and VRM
 - ▶ Separate fuses
- Power input is now shielded
 - ▶ Requires 2.5mm metric hex drive
- CAN bus interface
 - ▶ Allows measurement of current draw from slots
 - ▶ Has option for CAN bus termination



New Motor Controllers (Tentative)

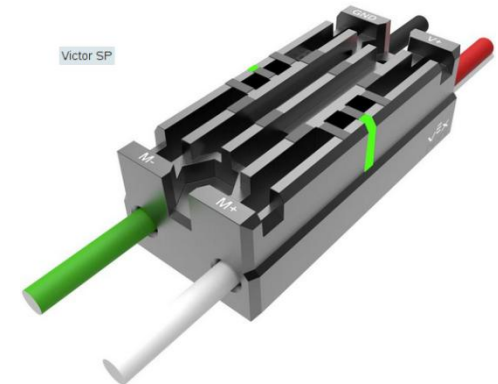
■ Talon SRX

- ▶ CAN-based equivalent to Jaguar
- ▶ Quadrature encoder input
- ▶ Forward and reverse limit switch inputs



■ VexPRO Victor SP

- ▶ Essentially, PWM-based Talon SRX
- ▶ No additional inputs or capability



CAN Bus

- Controller Area Network
 - ▶ If you've got a car made since 1968, you've got CAN bus
 - ▶ CAN is very reliable
- CAN bus got a bad rep from the early Jaguar motor controllers
 - ▶ Finicky connectors
 - ▶ Tricky termination issues
 - ▶ Slow update speeds
 - ▶ Thin traces would melt if the motor stalled for excessive time
- If you want to use Jaguars, they must be wired separately
 - ▶ Their CAN packet format is different than the rest of the CAN control system
 - ▶ Suggest using CTRE 2CAN to speed Jaguar CAN updates

CAN Bus #2

- New PCM, PDP, Talon SRX and RoboRIO all have CAN bus support
 - ▶ Two-wire daisy chain with fail-through capability
 - Failed component doesn't kill the bus
 - ▶ Much faster than serial CAN from earlier seasons
- RoboRIO has CAN termination
 - ▶ PDP has a jumper to select termination option
- CAN bus is *required* for PCM and PDP (if you want current-related data)
 - ▶ You can have more than one PCM on the robot if you need more solenoids

Summary

- The new control system is working pretty well at this point
 - ▶ New software loads are coming every few days up to kick-off
- Requirement for CAN bus if you want pneumatics is new
 - ▶ Closed-loop PCM makes working with pneumatics much easier
- New motor controllers are smaller and easier to work with than previous versions
 - ▶ Assuming they get approved by FIRST for competition