2015 Beta Testing

TechHounds FRC Team 868 Observations

Presentation Goals

- Review Hardware
- New roboRIO Observations
- Java Programming
- Questions/Demos

Quick Summary

- A lot of changes and enhancements
- No earth shaking changes (easy transition)
- A lot of expansion capabilities
- Expect this to be a more costly year
- Still need Windows
- Some old parts now scrap/teaching aids

Hardware

Power Distribution Panel (PDP)

- Smaller footprint
- CAN + Terminator
- 20 AMP Fuse!
- Read Volt & Temp
- Read Amps/Channel
- Estimate $200
- Class

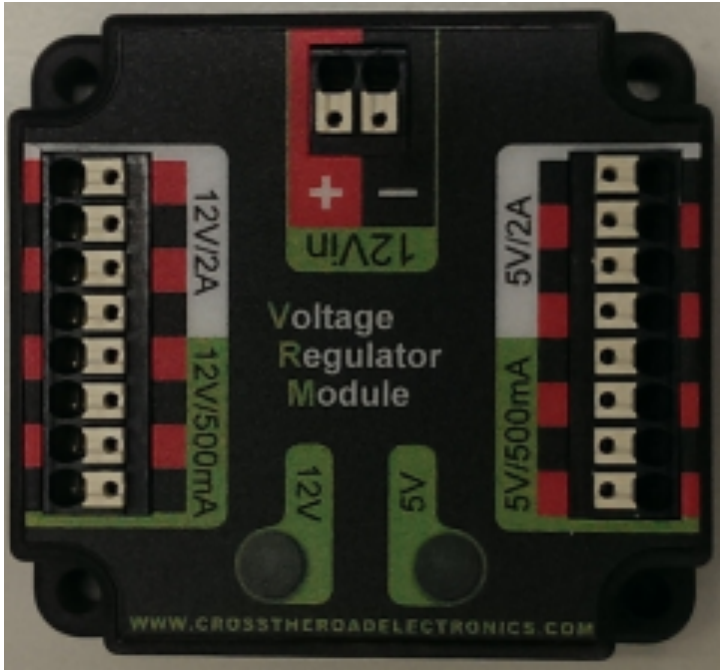
  `PowerDistributionPanel`

Pneumatics Control Module (PCM)

- CAN
- 20 AMP Fuse!
- Powers Compressor
- Estimate $90
- Classes
  - `Compressor`
  - `Solenoid`
  - `DoubleSolenoid`

Voltage Regulator Module (VRM)

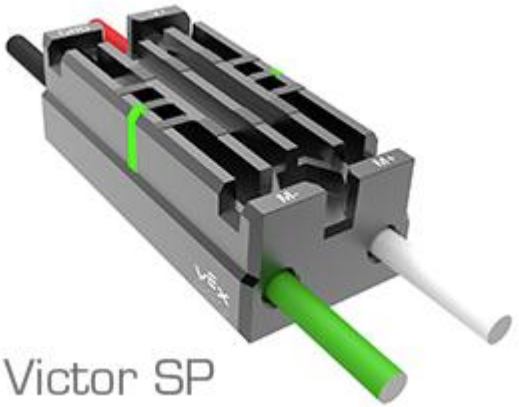
- Simple (dumb)
- Clean power
- Powers bridge/AP
- Low power output
- Is it required?
- Estimate $45



New Motor Controllers

Victor SP (think Talon)

- PWM Controlled
- $60 (Estimated)

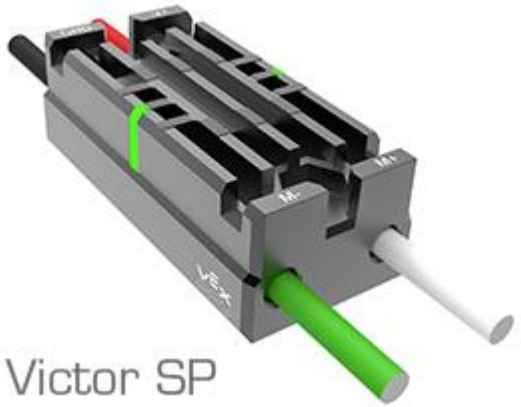


Talon SRX (think Jag)

- CAN Controlled
- $80 (Estimated)

Victor SP

- PWM Controlled
- Speed Control Only
- No Fan
- Sealed/Isolated
- Dumb/Simple
- No connectors



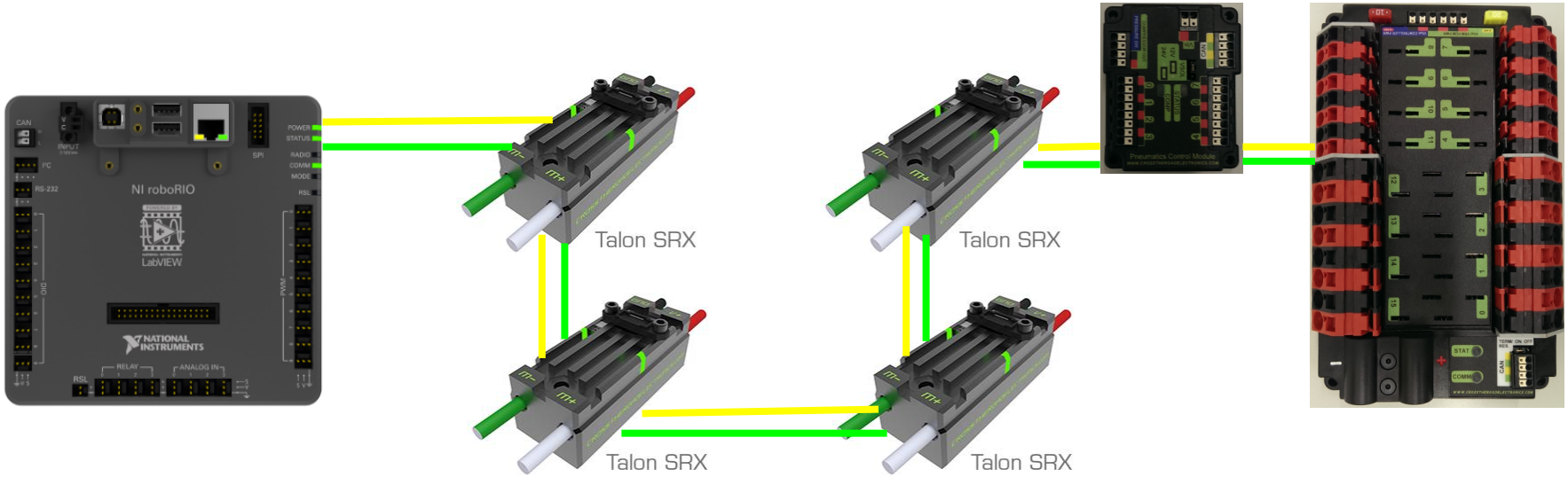
Talon SRX

- CAN Controlled
- Sensors (Limits?)
- Built in PID
- No Fan
- Sealed/Isolated
- Smart/Complex
- Sensor connector

Old Speed Controllers

- Talons still work
- Victors still work
- Jags still work
- All legal for 2015? Not sure.

CAN Overview



CAN (Controller Area Network) is a daisy chained communications bus where the two ends of the chain have a special terminator (see: http://en.wikipedia.org/wiki/CAN_bus).

Estimate $1045 for parts shown

CAN Notes Part 1 of 2

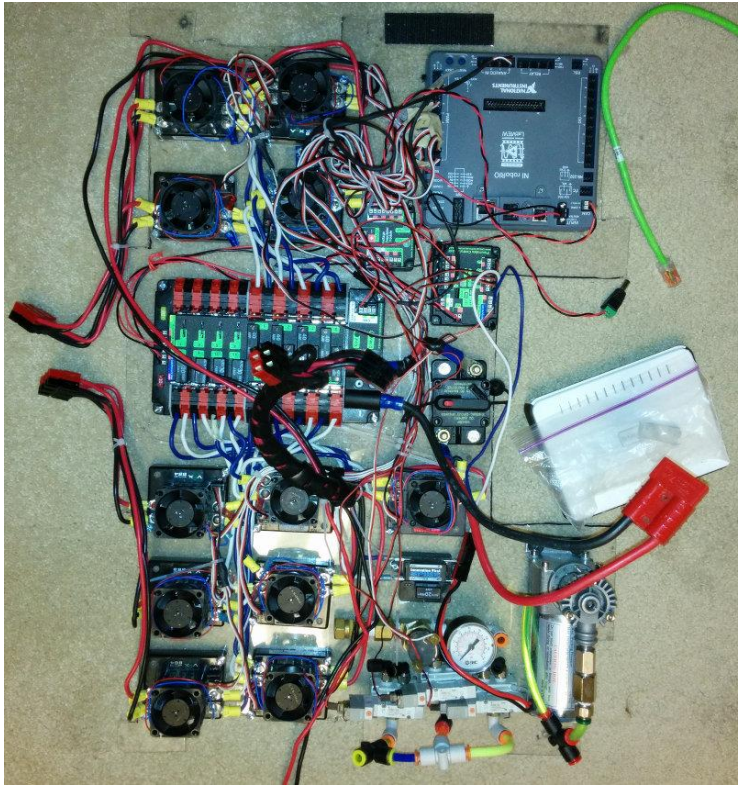
- Use green and yellow wire (16-24AWG)
- roboRIO MUST be at one end of bus
- PCM should be in middle of bus
- Recommend Talon SRX in middle of bus
- Recommend Jaguar in middle of bus
- Recommend PDP at other end of bus

CAN Notes Part 2 of 2

- Talon SRX should plug in directly
- Uncertain as to Talon SRX firmware updates
- Need RJ12 to 2 wire adapters for Jags (you just need two - one at each end of chain).
- Still need RJ12 to serial for Jag firmware

Wiring Notes

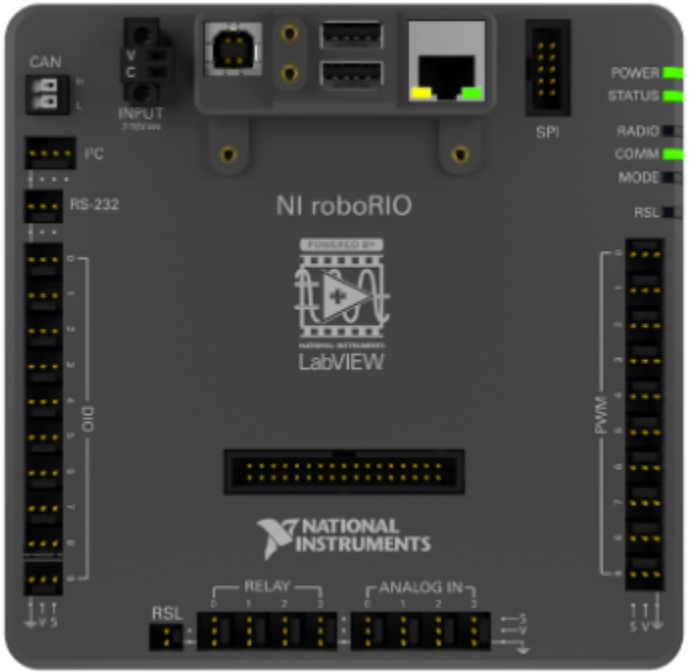
- Nicer/Quicker
- Fewer dongles
- Fewer connectors for new speed controllers
- 18AWG to power PCM
- Compressor power doesn’t feel right



roboRIO

roboRIO = cRIO + Digital Sidecar

- PWM x10
- DIO x10
- Analog x4
- Relay x4
- SPI, RS-232, CAN, I2C
- USB x2
- Ethernet + USB Comm
- Expansion Port
- Estimated Cost $435

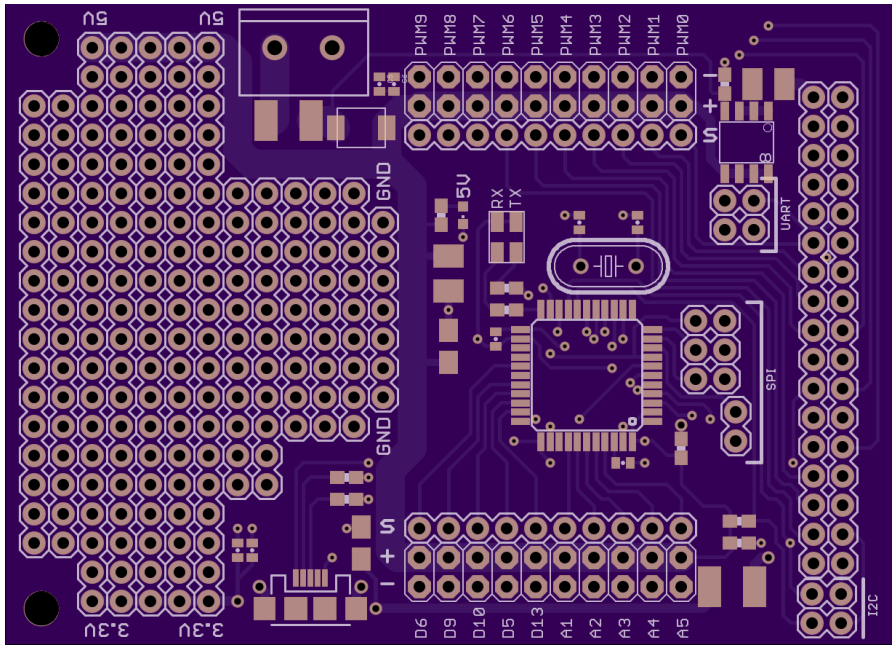


Wiring Comments

- Still has dongle to connect to power
- Will not accept keyed PWM cables
- Team excited about breakout pins (working on breakout board with Arduino addition)
- Careful with USB (ground touching 12V is bad)
- Exposed opening to collect metal shavings

Breakout Board

- Custom Electronics Port (MXP)



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

Firmware Upgrades

- Windows Firmware update utility
- Use USB cable when updating firmware

OS Notes

- Linux based
- Busybox version of many commands
- Can run other code (Full JRE 8)
- Some USB devices supported
- No video output
- admin instead of root
- No password by default on admin account

Windows Toolset

- Command Prompt (ping, ipconfig, tracert)
- PuTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/)
- WinSCP (http://sourceforge.net/projects/winscp/)
- Notepad++ (http://sourceforge.net/projects/notepad-plus/)
- Nmap (http://nmap.org/)
- Wireshark (https://www.wireshark.org/download.html)

Remote Access to roboRIO

- Nice web interface on port 80
- You can ssh (PuTTY) to admin account
- Automatic name resolving regardless of Ethernet or USB (roborio-868.local)

```
ping roborio-868.local
ssh admin@roborio-868.local
http://roborio-868.local/
```

Java Programming

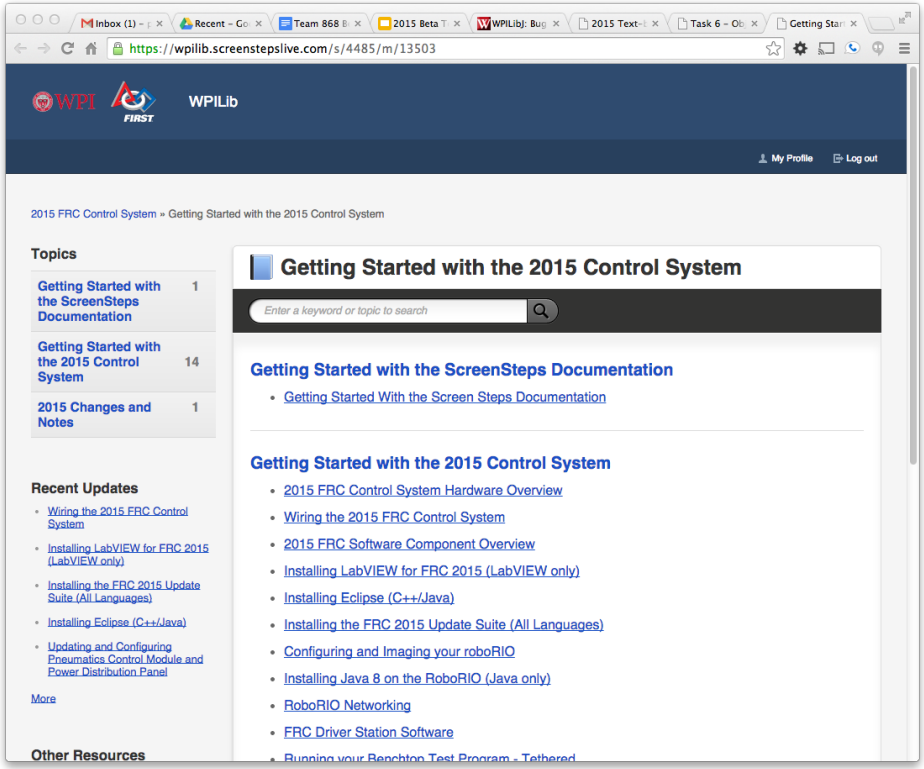
Overview

- For the most part, Java programming will look the same.
- Programming is now done in Eclipse--code is deployed via plugins.
  - Netbeans is no longer supported.
- Windows is still required for Driver Station support.
- Ant build scripts to compile/deploy.

Setup First Time

- Long Process
- Excellent Instructions
- Use Windows 7 or 8 First
- Use 32 bit JDK/Eclipse
- Follow Instructions

Don’t be a Click Monkey!



Similarities

- Command-base is still available.
  - Eclipse provides project templates for Command Base, Sample Robot and Iterative Robot.
- For the most part, previously-existing classes have not changed.

Differences

- Classes added for new hardware components:
  - `PowerDistributionPanel`
  - `Compressor`
  - `Solenoid`
  - `DoubleSolenoid`
- Deploy is now done through SSH:
  - roboRIO runs Linux, terminal can be accessed via SSH.
  - Deploy can be password-protected; specify password in build.properties.
  - Deploy now takes ~10 seconds.

Removals

- Kinect sensor
- Cypress board
- DriverStationLCD class removed
- No more start/stop methods for Gyro, Counter, Encoder
- Team ID no longer available

Additions

- Interrupts (digital or analog trigger)
- Fully automatic compressor control
- Information from PDP
- Built-in accelerometer
- Analog outputs on MXP

Performance Issues

- Eclipse (ant deploy) puts a load on system when robot code is running.
- Run task manager (top)
- If you get multiple instances going, load gets worse
- Restarting Eclipse seems to clear issue
- Fix for “tail” load Eclipse/roboRIO coming

Using Breakpoints

- Be paranoid (put the robot on blocks)!
- Debug run is not difficult
- Multiple debug runs were a bit cumbersome
  - Stop current debug session
  - Return to Java perspective
  - Start new debug session
- Fix to debug is coming

If DNS Doesn’t Work

- Development machine unable to resolve roborio-868.local
- Development machine able to reach by IP
- Added following to the build.properties file:

```
target=10.8.68.2
```

If admin Password

- Development machine unable to log into roboRIO
- Deploys will fail
- Unsure how to remove admin password
- Add password to the build.properties file:

```
password=ROBORIO_PASSWORD
```

InterruptHandlerFunction<?>

- New Feature
- Hardware Triggered
- Trickier than Threads
- Worry About
  - Synchronization
  - Deadlock

```
// To make this thread safe, you
// could use synchronized, but how
// do you make it interrupt safe?

public void addObject(Object objToAdd) {

  if (myCntr < myBuffer.length) {
    myBuffer[myCntr] = objToAdd;
    myCntr++;
  }

}
```

Hardware to Class Map

roboRIO Classes

CAN
- CANJaguar
- PowerDistributionPanel
- Compressor
- Solenoid
- DoubleSolenoid

Built In
- BuiltInAccelerometer
- Utility.getUserButton
- DriverStation.getBatteryVoltage

SPI
- SPI
- ADXL345_SPI

I2C
- I2C
- ADXL345_I2C

PWM x 10
- PWM
- Servo
- Jaguar
- Victor
- Talon

RS-232 x 1
- SerialPort

DIO x 10
- DigitalInput
- DigitalOutput
- Counter
- GearTooth
- Encoder
- Ultrasonic

Relay x 4
- Relay

Analog In x 4
- AnalogInput
- AnalogTrigger
- Gyro
- AnalogAccelerometer
- AnalogPotentiometer

roboRIO MXP Port

DIO x 16 (10 - 25)
- DigitalInput
- DigitalOutput
- Counter
- GearTooth
- Encoder
- Ultrasonic

Analog In x 4 (4 - 7)
- AnalogInput
- AnalogTrigger
- Gyro
- AnalogAccelerometer
- AnalogPotentiometer

Analog Out x 2 (0 - 1)
- AnalogOutput

I2C x 1
- I2C

SPI x 1
- SPI

RS-232 x 1
- SerialPort

PWM x 10 (10 - 19)
- PWM
- Servo
- Jaguar
- Victor
- Talon

NOTE: DIO pins are shared with above

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

PCM Classes

Compressor Out
Pressure SW
- Compressor

Pneumatics x 4
- Solenoid
- DoubleSolenoid

Pneumatics x 4
- Solenoid
- DoubleSolenoid

PCM Classes

Compressor Out
Pressure SW
- Compressor

Pneumatics x 4
- Solenoid
- DoubleSolenoid

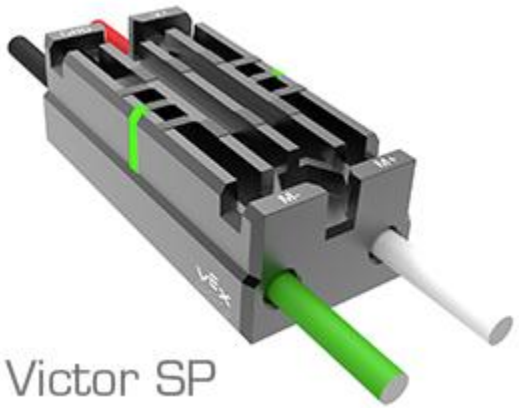
Pneumatics x 4
- Solenoid
- DoubleSolenoid

NOTE: You do not need to program/use Compressor class unless you want more control.

New SpeedController Classes

PWM
- Victor?

CAN
- CANJaguar?



PDP Classes

PowerDistributionPanel

- getVoltage() // Volts
- getTemperature() // Celsius
- getCurrent(0-15) // Amps

Activitity

- Class to track min/max/avg
- Robot class to post to network table
- Stand-alone Java application to monitor
- Options to show min/max/avg of each channel and sum of all channels (how close are we to 120 amps)

Goal

- Learn comm between robot program and external program
- Tool for checking load and/or max load
- Something to do while waiting on build