

# FRC LabVIEW Quick Start Guide

## FRC 2015 Training Material

Setup

NI LabVIEW  
Basics

Vision, PID,  
and  
Simulation

Advanced  
Programming

# FRC LabVIEW Quick Start Guide

Setup

NI LabVIEW  
Basics

Vision,  
PID, and  
Simulation

Advanced  
Programming

- Software Setup Overview
- FRC Software Orientation
- roboRIO Introduction and Set Up
- Basics of the Robot Project
- Deploying and Driving

# FRC Software Overview

## NI Software for FRC 2015

- LabVIEW 2014
- LabVIEW 2014 Real-Time
- Vision Development Module
- IMAQdx 14.0 (vision driver)
- NI-RIO (roboRIO driver)
- FRC 2015 Simulation



Included in Kit of Parts  
or  
Download at [ni.com/frc](http://ni.com/frc)

## FRC 2015 Update Suite

- Driver Station
- roboRIO Utilities
- WPI Library
- FRC LabVIEW Templates
- FRC LabVIEW Examples

Required for ALL FRC Teams  
Download Only

# 1. Install NI Software

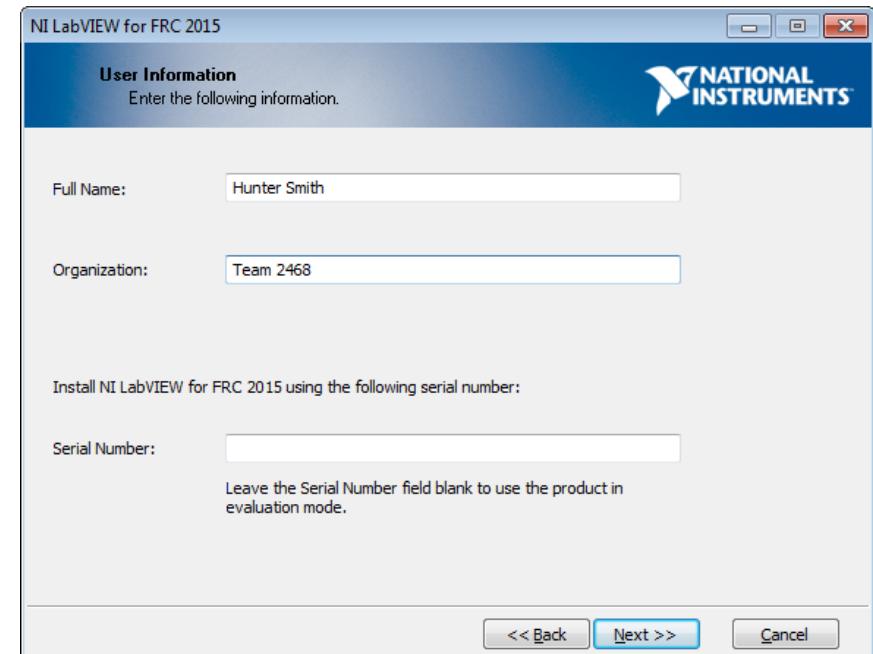
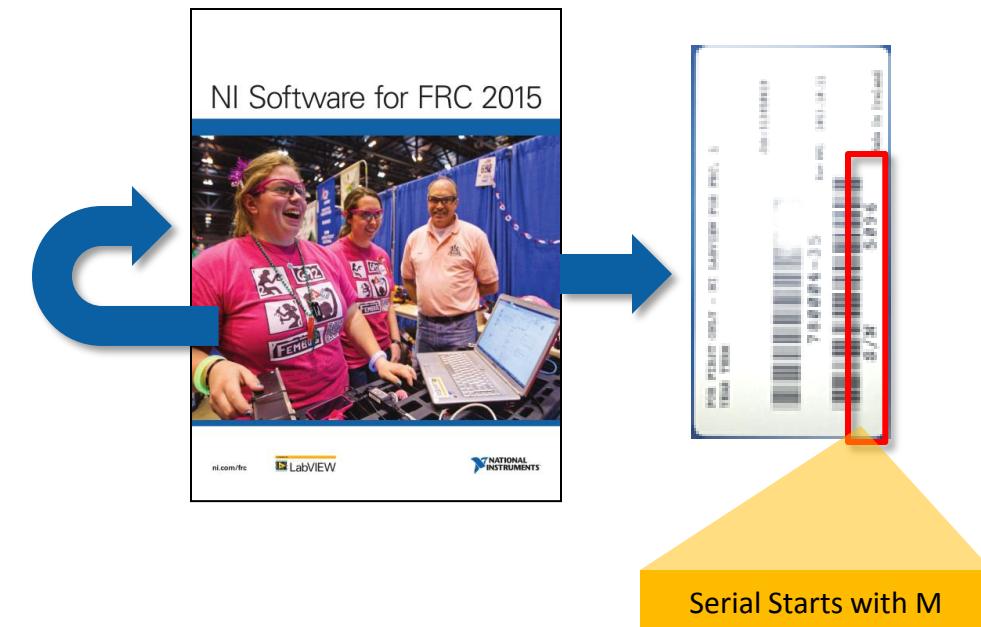
## NI Software for FRC 2015 (DVD or Download)

- If you download the software, you must unzip and run AutoRun.exe.



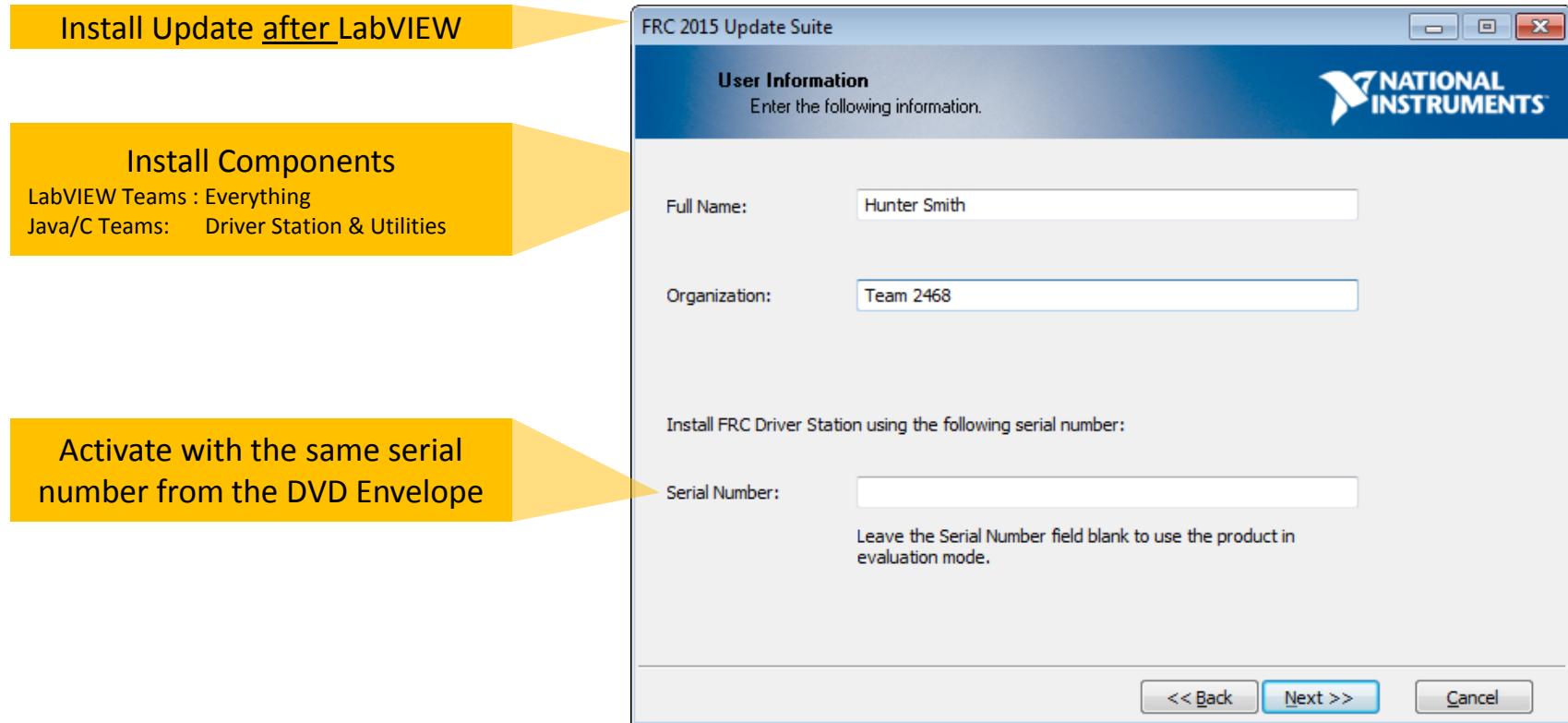
## 2. Activate

- Activate LabVIEW using the serial number on the back of the DVD envelope.



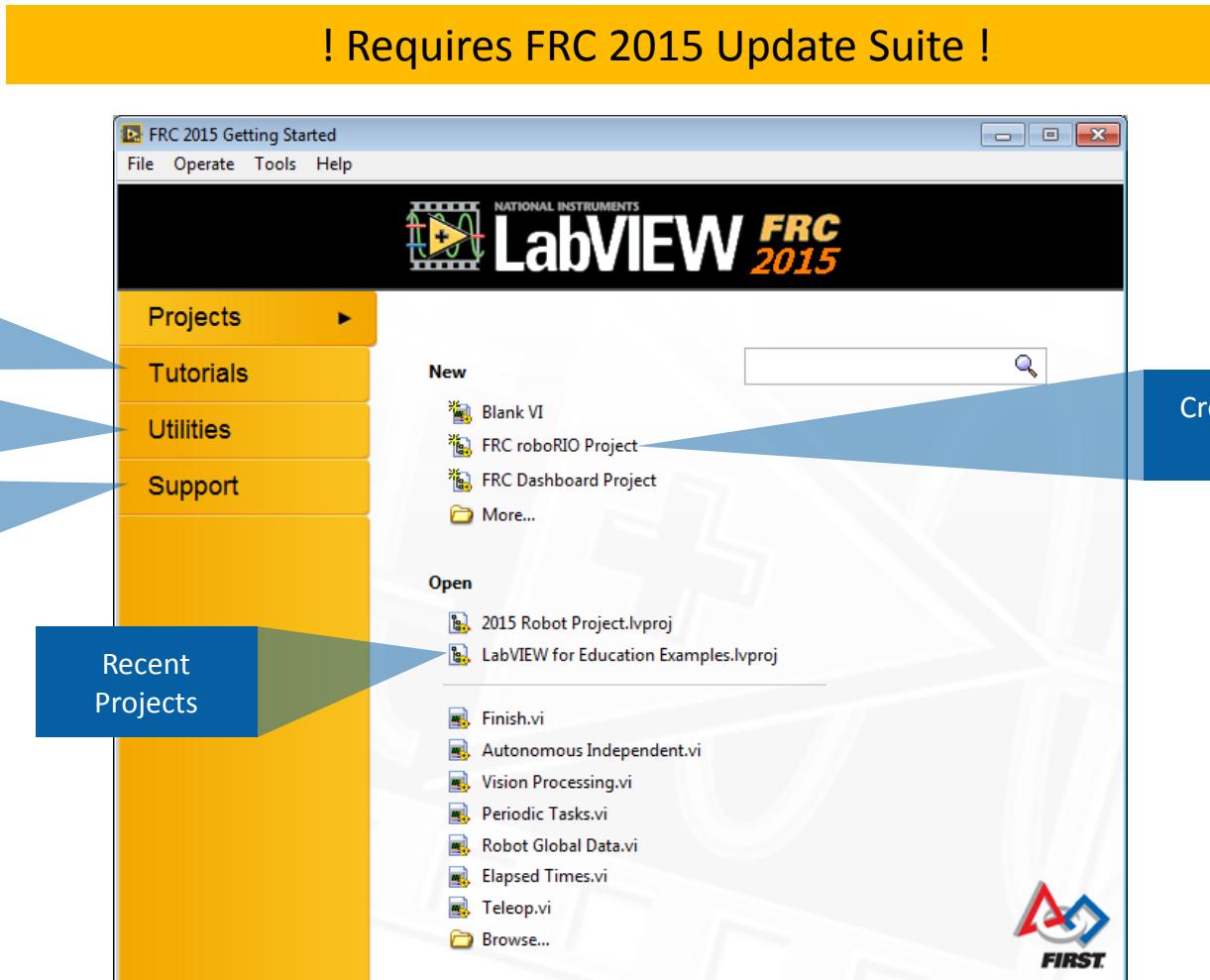
# 3. Install FRC 2015 Update Suite

All teams must install the Update Suite.



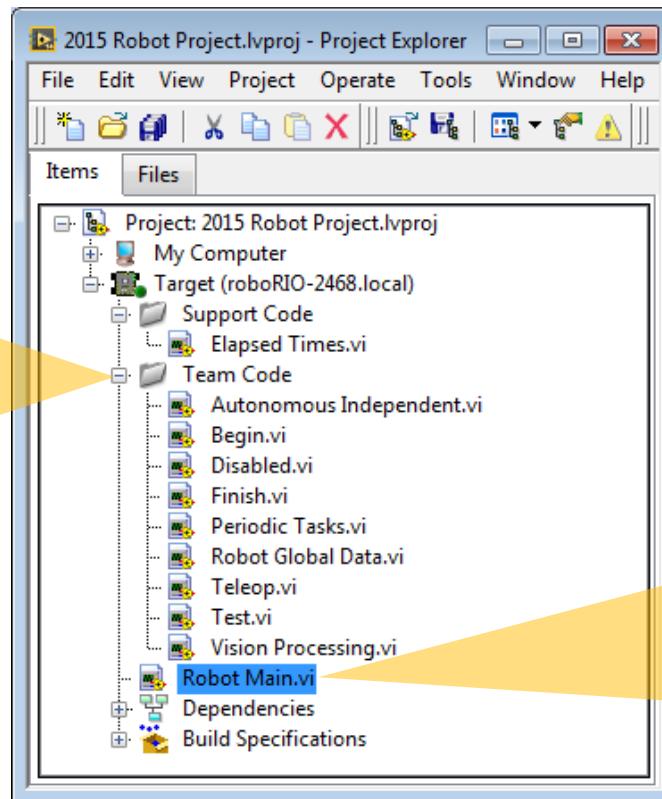
# Software Orientation: Getting Started Window

This window opens when you start FRC LabVIEW.



# Software Orientation: roboRIO Robot Project

- With the LabVIEW project window, you can see all of the code that will run on your robot. The FRC software includes a template project with everything you need to start.

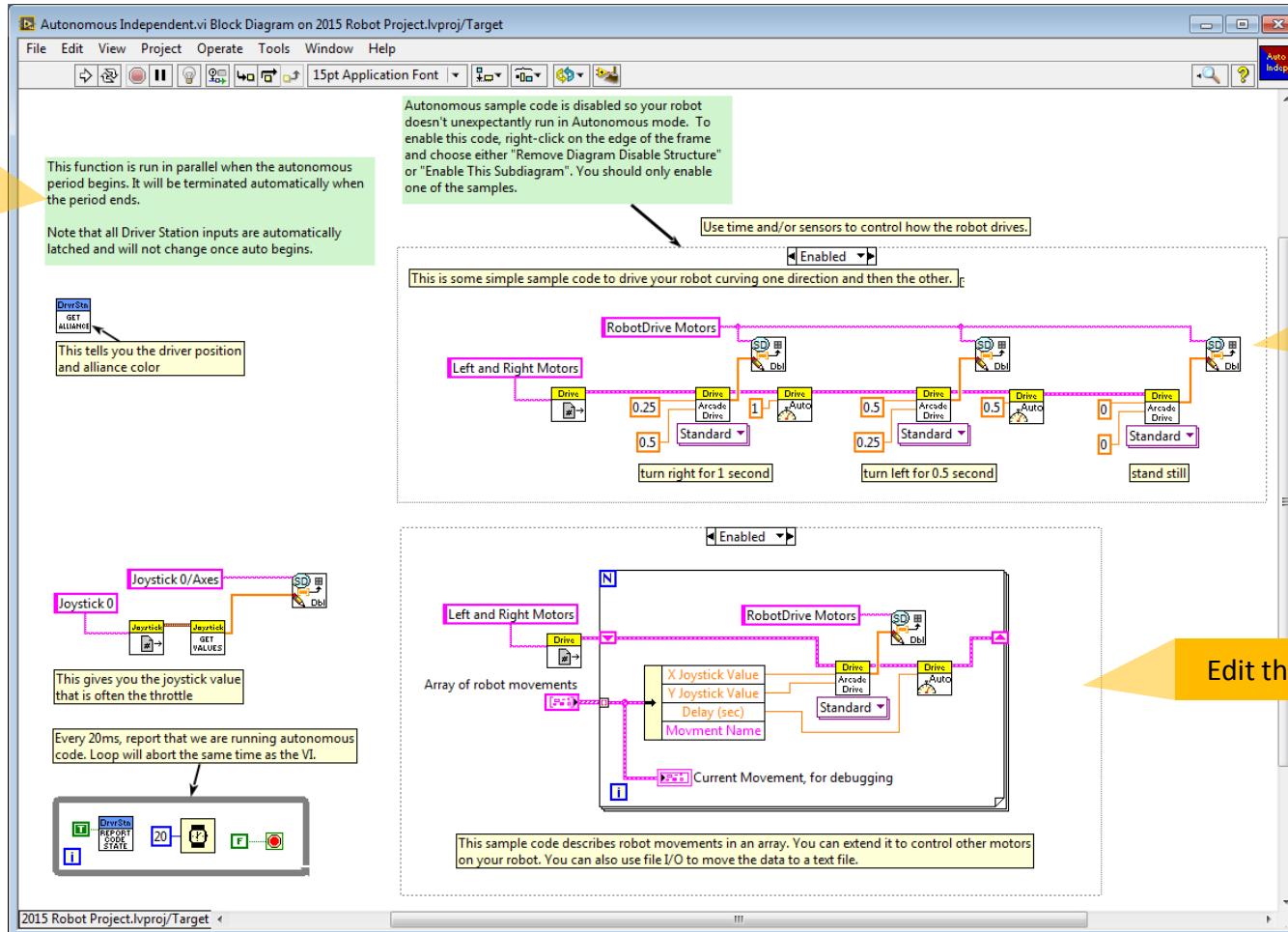


All the code you need to change  
is in the Team Code folder

Robot Main communicates with the field  
control system and runs your team code  
(don't edit)

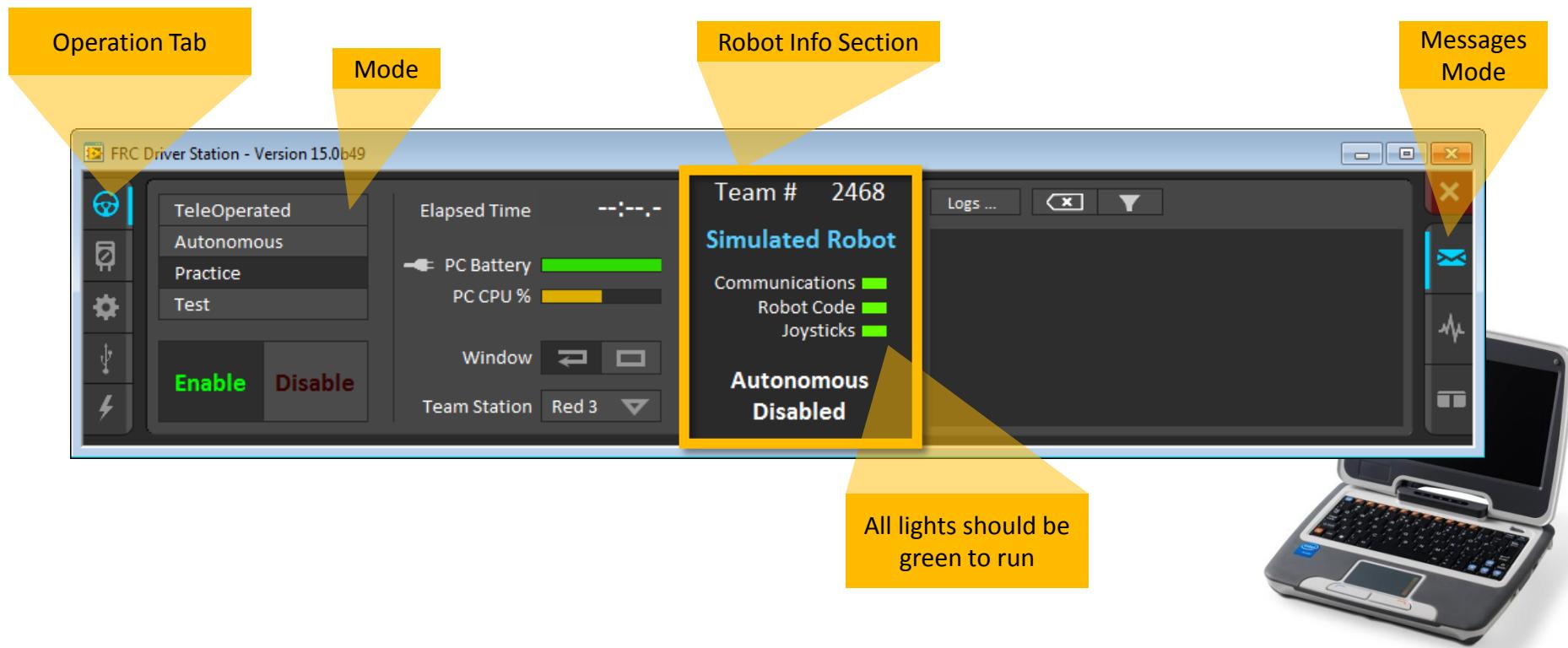
# Software Orientation: Team Code

This is the Autonomous Independent VI for creating your autonomous program. This is one of the VIs you need to edit.



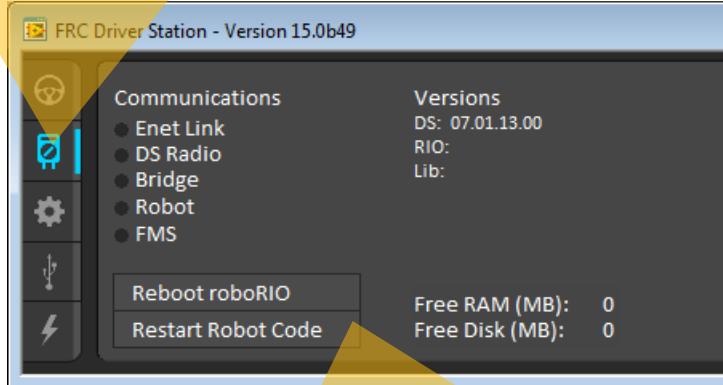
# Software Orientation: Driver Station

The Driver Station is the robot control utility for practice and competition for all FRC teams. Open this to connect to your robot and run your code you wrote.

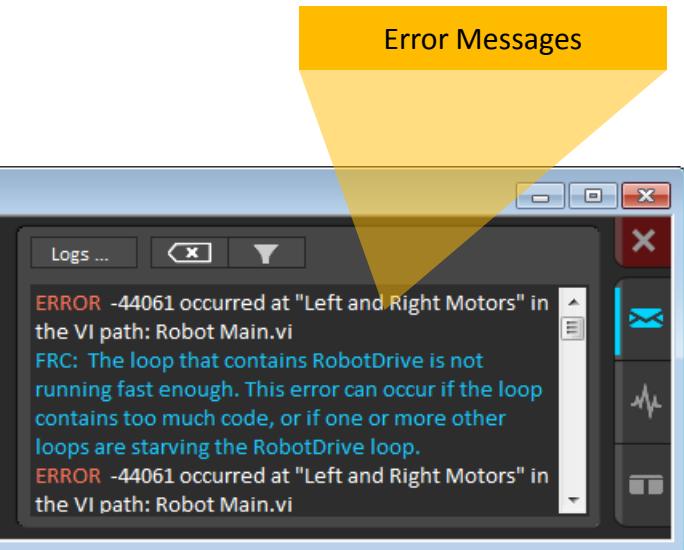


# Software Orientation: Driver Station

Diagnostics Tab



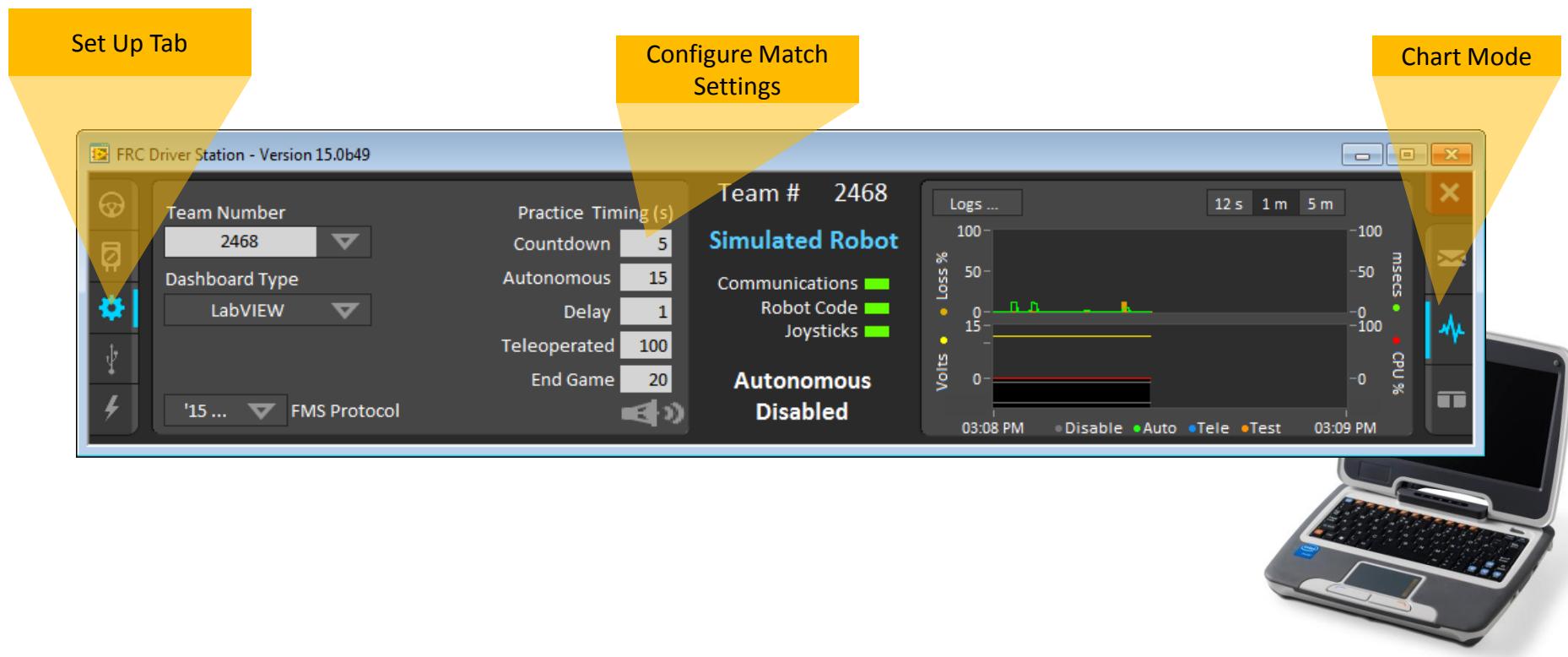
Error Messages



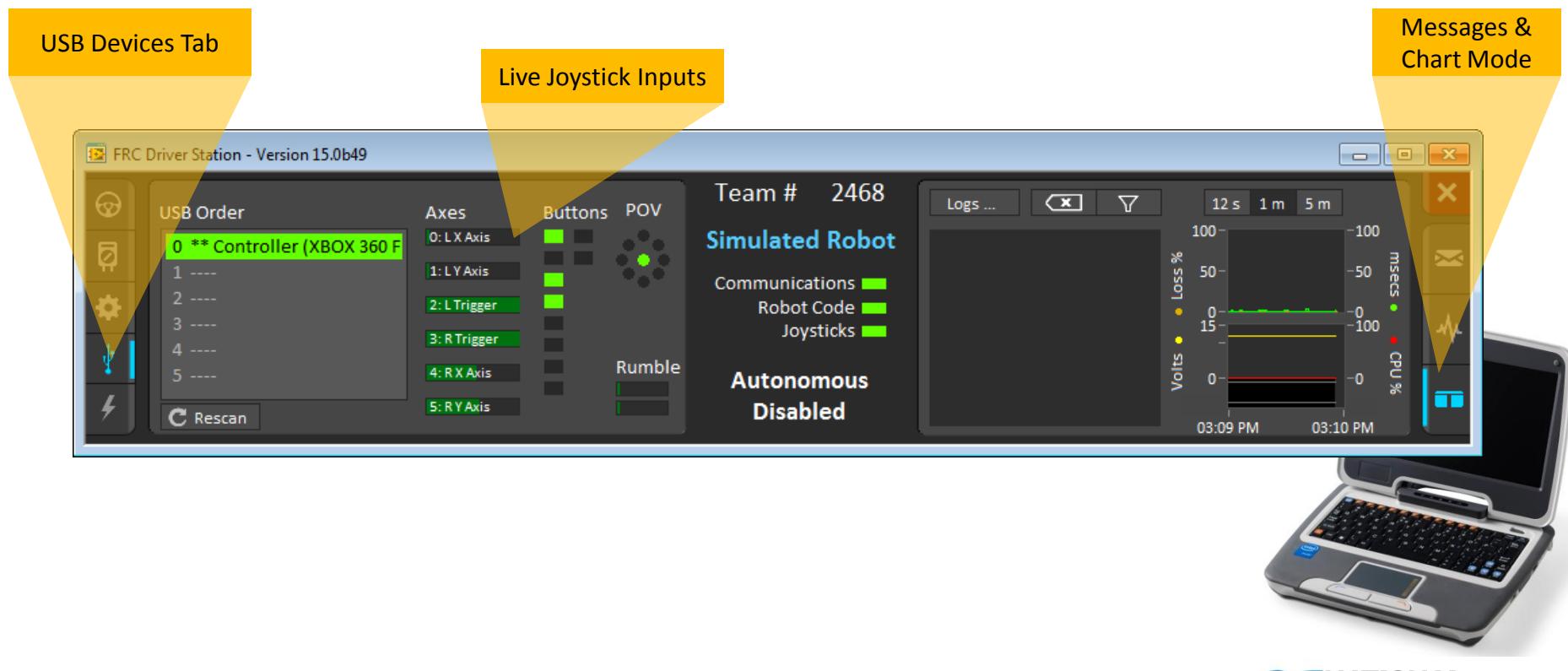
Easily Reboot or Restart from the dashboard



# Software Orientation: Driver Station

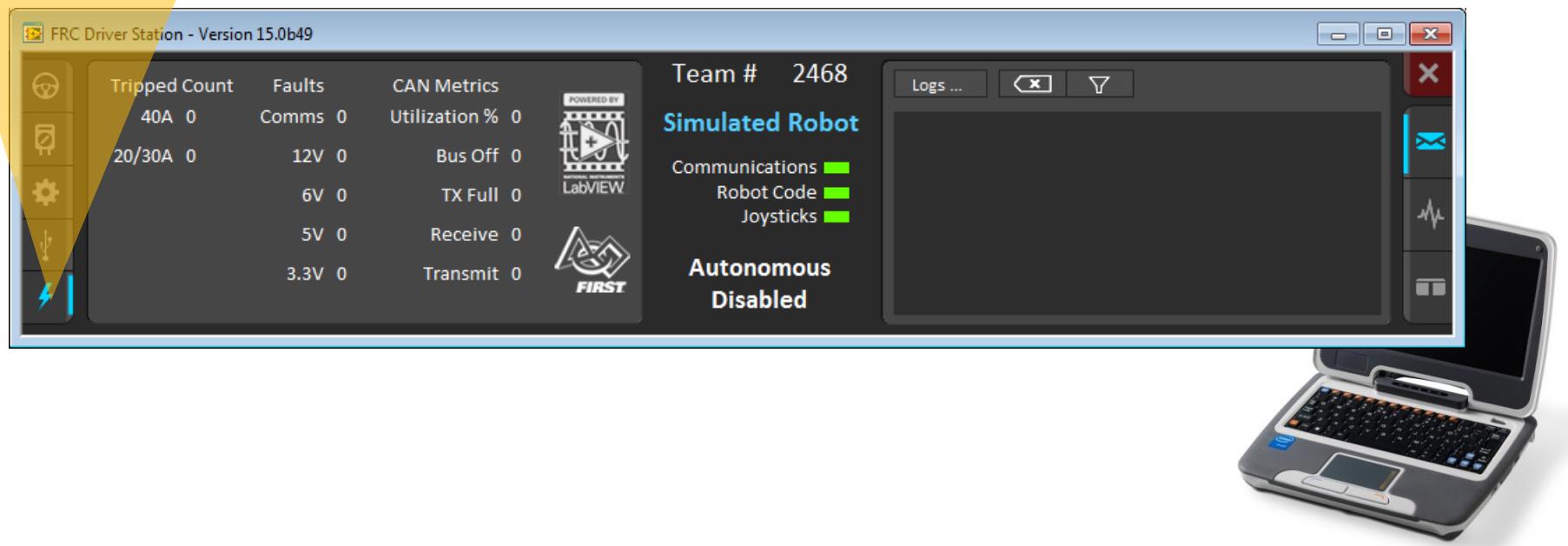


# Software Orientation: Driver Station



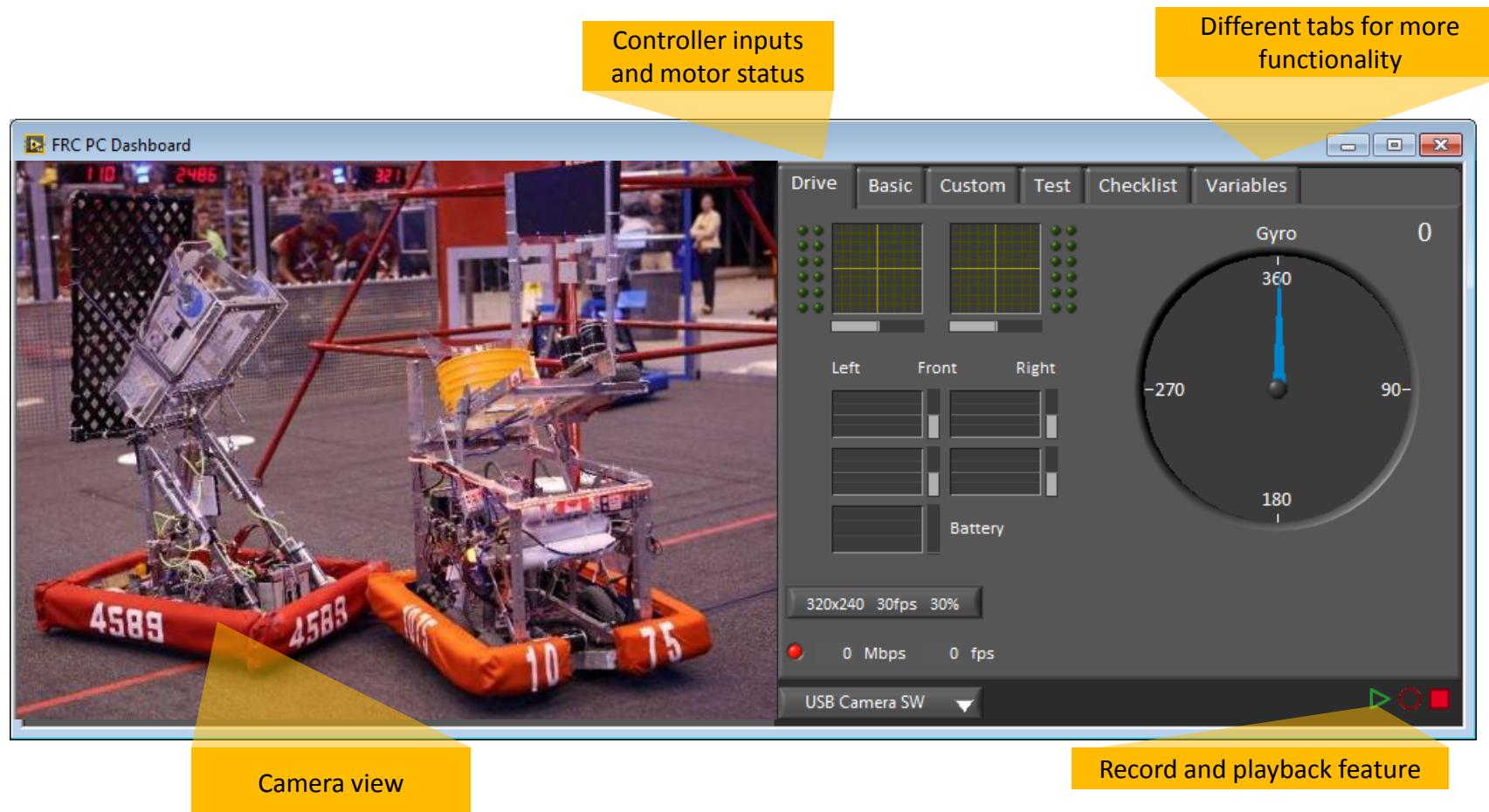
# Software Orientation: Driver Station

## Power & CAN Tab



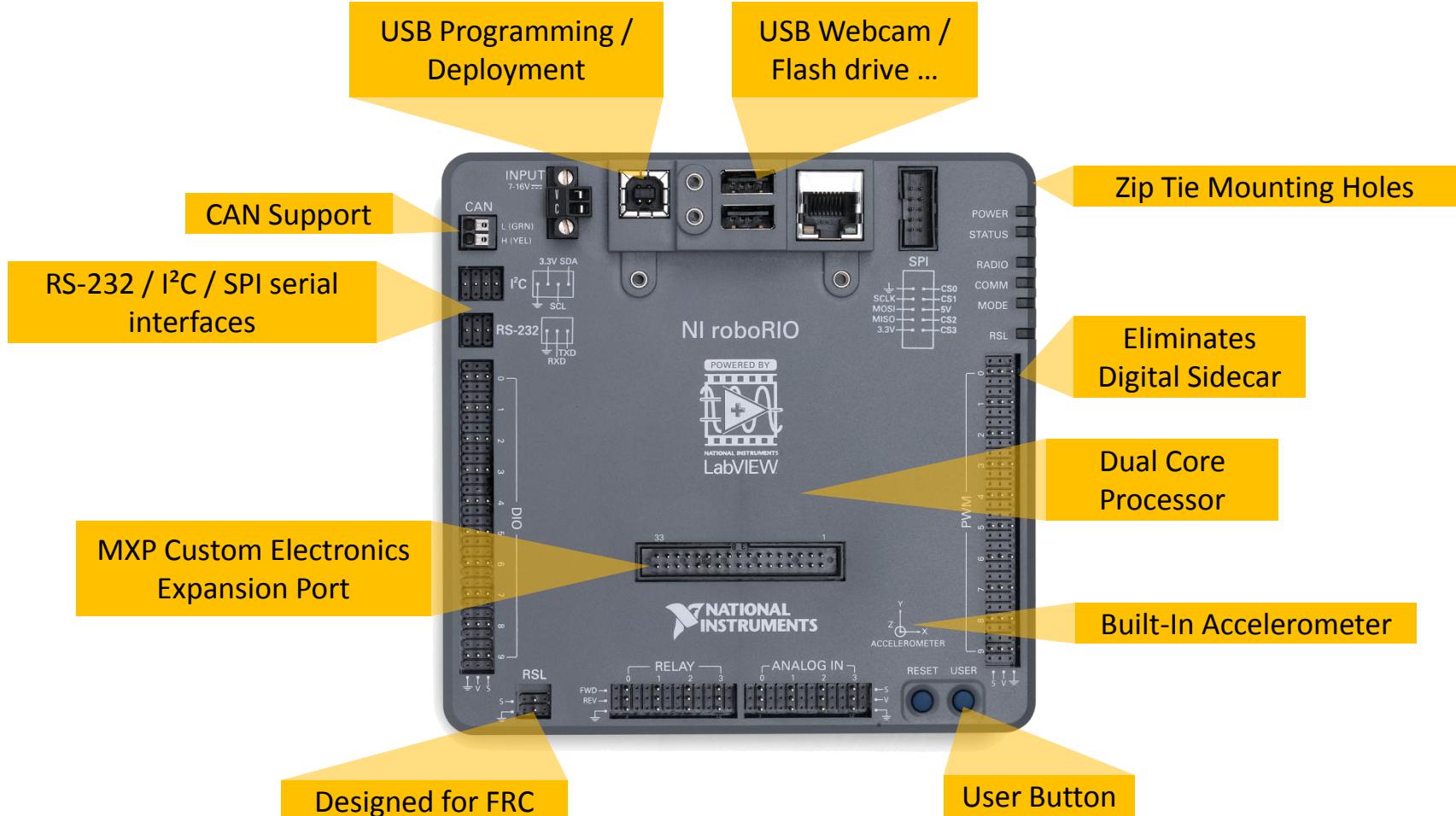
# Software Orientation: Dashboard

The Dashboard is a customizable display and control utility that you can use to read the current status of controllers, sensors, motors, and more.



# roboRIO

New controller for FRC 2015-2019



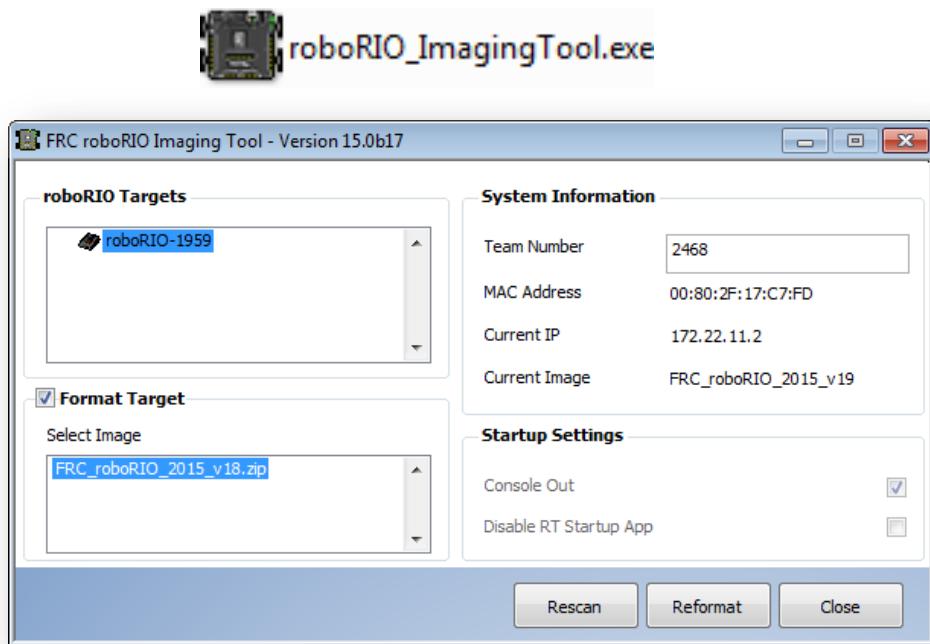
# roboRIO vs cRIO



Controller	cRIO FRC II	roboRIO
Processor	Freescale 5125 400MHz PPC	Xilinx Zynq 7020 667Mhz dual-core ARM Cortex-A9
OS	VxWorks	Linux with Real-Time Extensions
RAM	128MB	256MB
Disc	256MB	512MB
Volume	110 in <sup>3</sup> *	30 in <sup>3</sup>
Mass	60oz *	10oz
Price	\$689 *	\$435

# roboRIO Set Up

1. Install the FRC 2015 Update Suite
2. Connect roboRIO to a power supply
3. Connect roboRIO to your PC over USB
4. Run the roboRIO Imaging Tool



# MXP – myRIO Expansion Port

Create your own expansion board for roboRIO

[bit.ly/roboRIO-MXP](http://bit.ly/roboRIO-MXP)

All FRC Teams get a Free License  
for the NI Circuit Suite

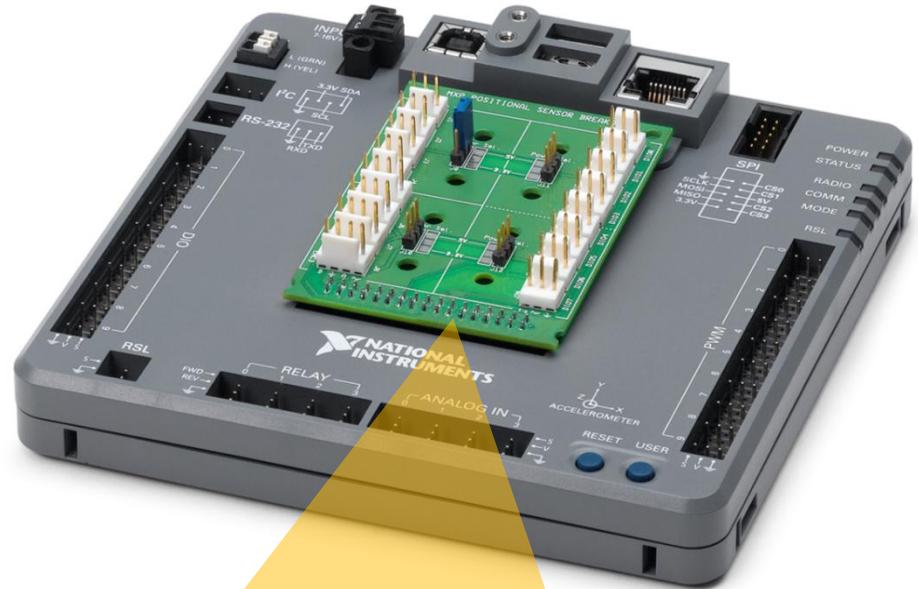


Multisim Circuit  
SPICE & Simulation

Ultiboard PCB  
Design & Layout



Be sure to check the official rules  
from FISRT to make sure your  
accessories are competition legal



MXP I/O

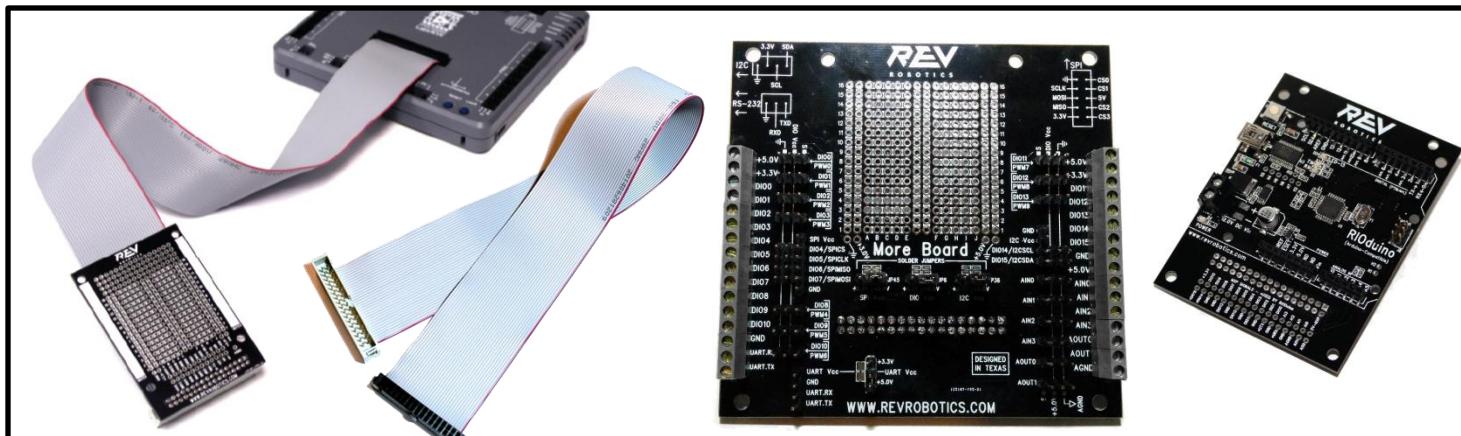
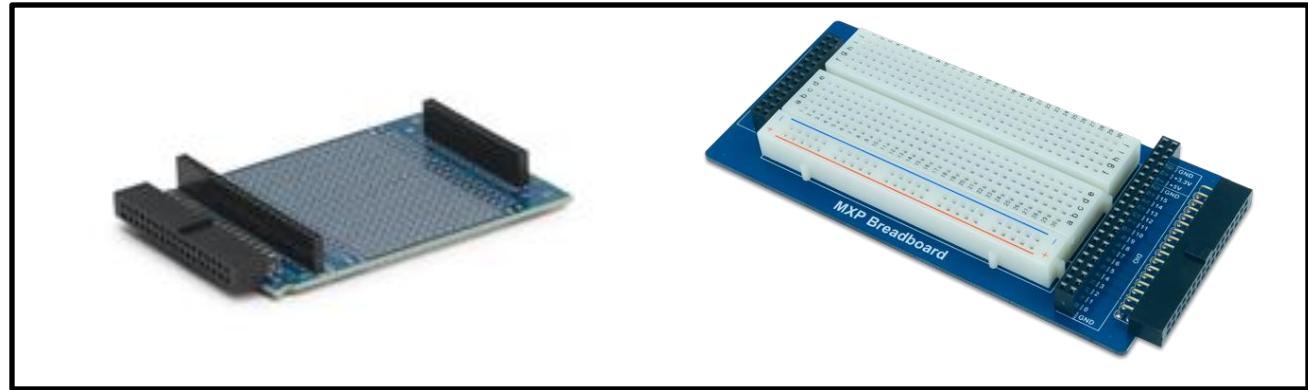
10 PWM / DIO  
I2C / SPI / UART  
3.3V / 5V / GND  
2 Analog Outputs  
4 Analog Inputs

# MXP Accessories

Adaptors and prototyping tools



Available from [ni.com](http://ni.com)



Available from [AndyMark](http://AndyMark.com)

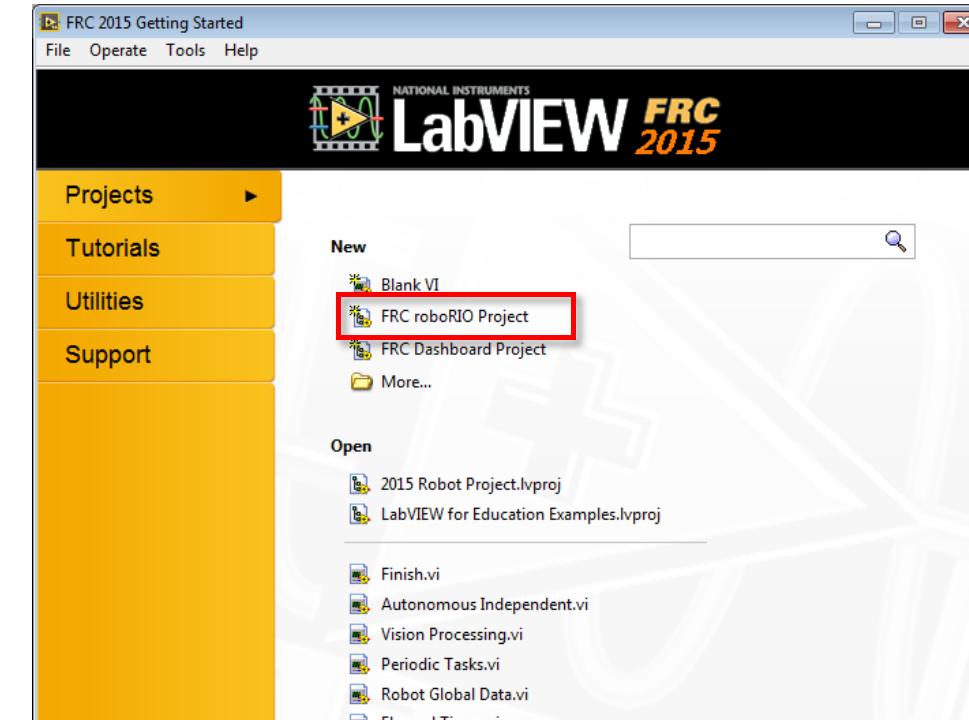
Create a roboRIO accessory? Let us know @NIFIRSTRobotics

# Drive a Robot: Step 1

## Open the Template Project

Setup

**Step 1:** From the LabVIEW Getting Started window, select **New»FRC roboRIO Robot Project** to create a new roboRIO Robot Project. (Step 0: Open LabVIEW for FRC.)



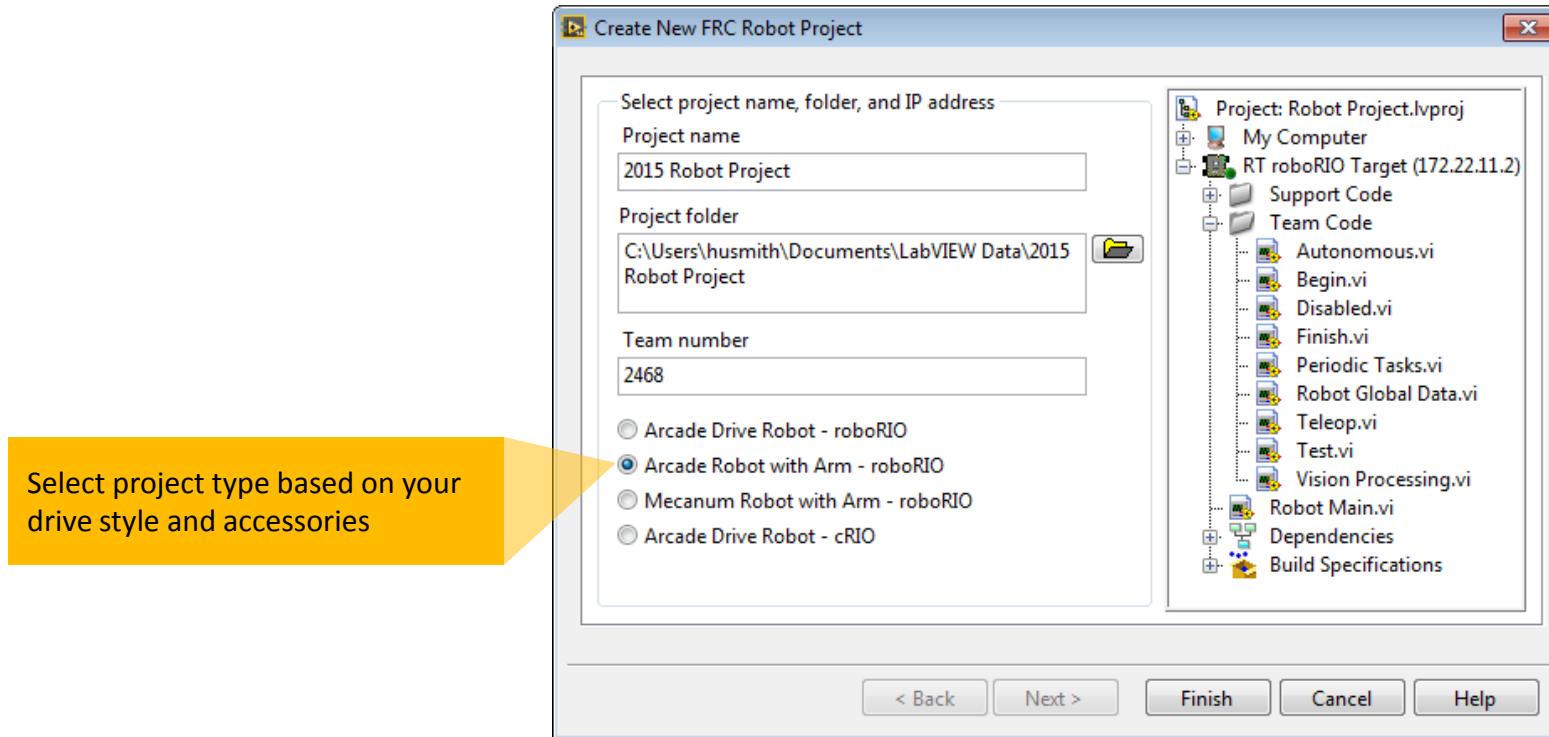
If you plan to use real hardware, you need to properly set up the roboRIO controller and D-Link before attempting the rest. You can run the project in simulation.

# Drive a Robot: Step 2

Setup

## Configure Your roboRIO Robot Project

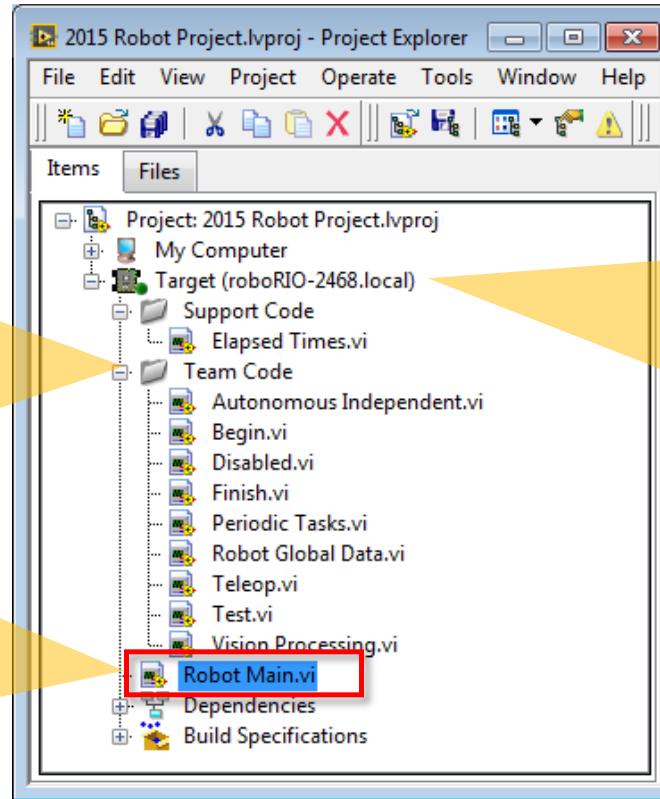
**Step 2:** Fill in the New Project dialog box. Give the project a meaningful name, set your team number and pick the type of project you want to make.



# Drive a Robot: Step 3

## Open the Template Project

Setup



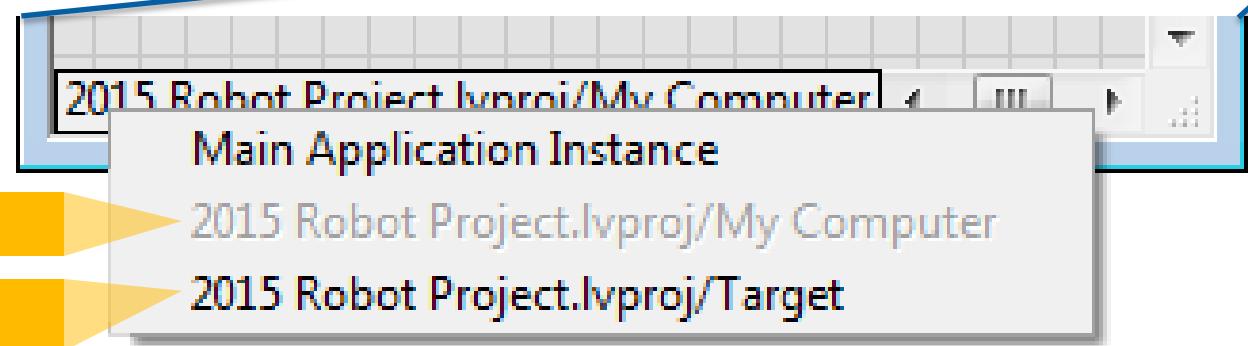
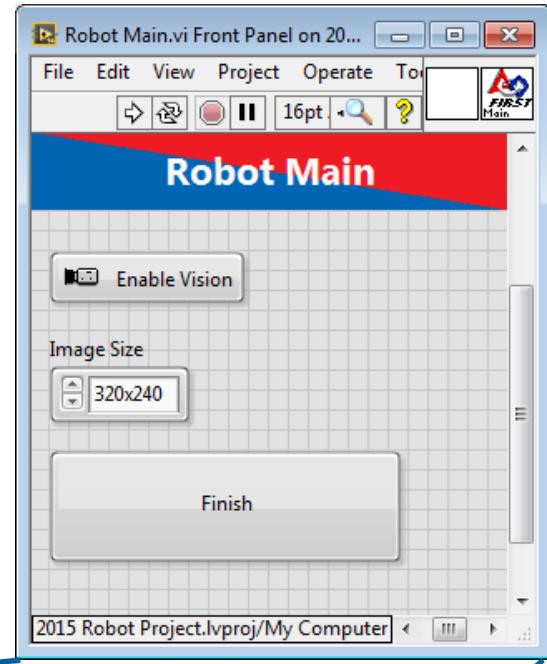
**Step 3:** To open **Robot Main.vi**, double-click on it in the project window. The front panel should launch. Select <Ctrl-E> to open the block diagram and look at the code.

# Drive a Robot: Step 4

## Select a Target

Setup

**Step 4:** If you have a roboRIO controller connected, make sure you select it. If you don't have an actual robot to drive around, you can still simulate one. Select the **...lvproj/My Computer** target to open the simulator.

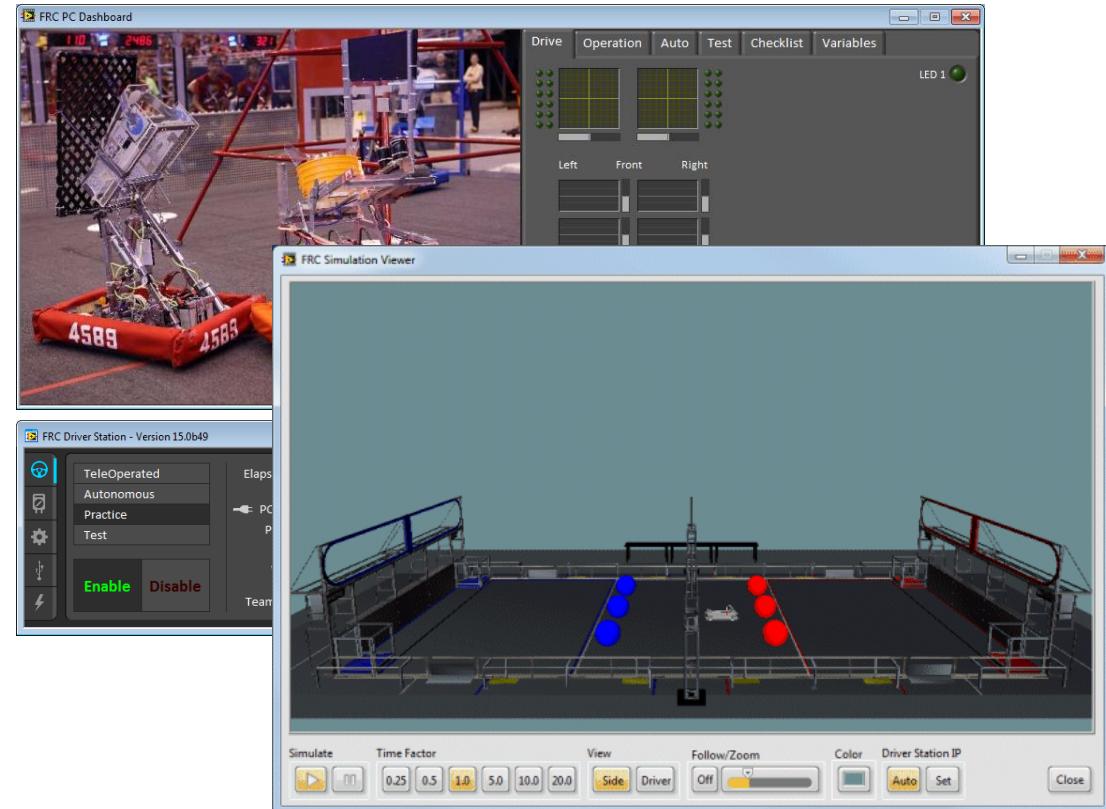
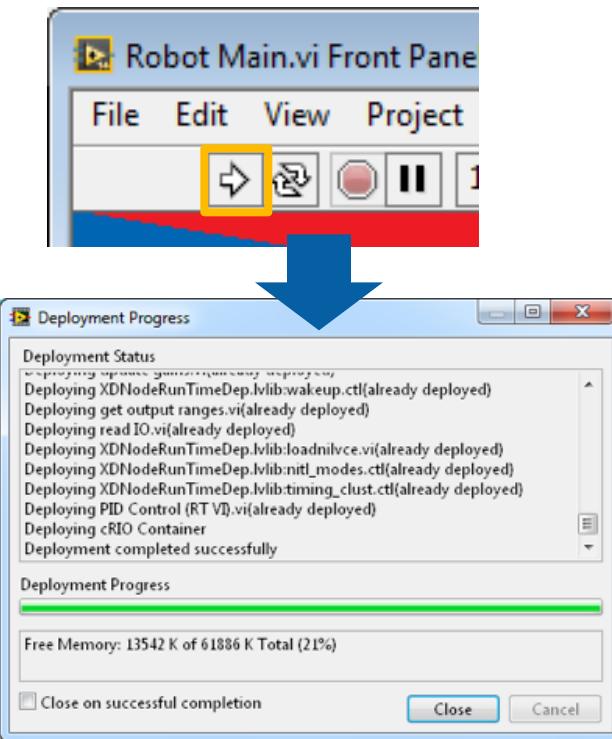


# Driving a Robot: Step 5

## Deploy Code

Setup

**Step 5:** Click on the **white run arrow** at the top left of Robot Main. The robot project is deployed to the roboRIO controller/simulator. Once LabVIEW is finished deploying the code, open the Driver Station. The LEDs should all be green. Click **TeleOp** and then **Start**.



# Setup Resources

- Software Setup Guide
- [LabVIEW Development Suite for FRC 2015](#)
- [Imaging roboRIO and Windows IP Configuration](#)
- [Troubleshooting roboRIO Connections](#)
- [Driver Station Tutorial](#)
- [Custom Dashboard Tutorial](#)
- [Robotics Framework Tutorial](#)
- [Robot Simulation Tutorial](#)
- Forum Support (year round)
  - [ni.com/frc](http://ni.com/frc)
- Phone Support (during build season only)
  - (866) 511-6285 from 1:00 to 7:00 p.m. (Central Time)

All links available at  
ni.com/frc or  
click this button



# FRC LabVIEW Quick Start Guide

Setup

NI LabVIEW  
Basics

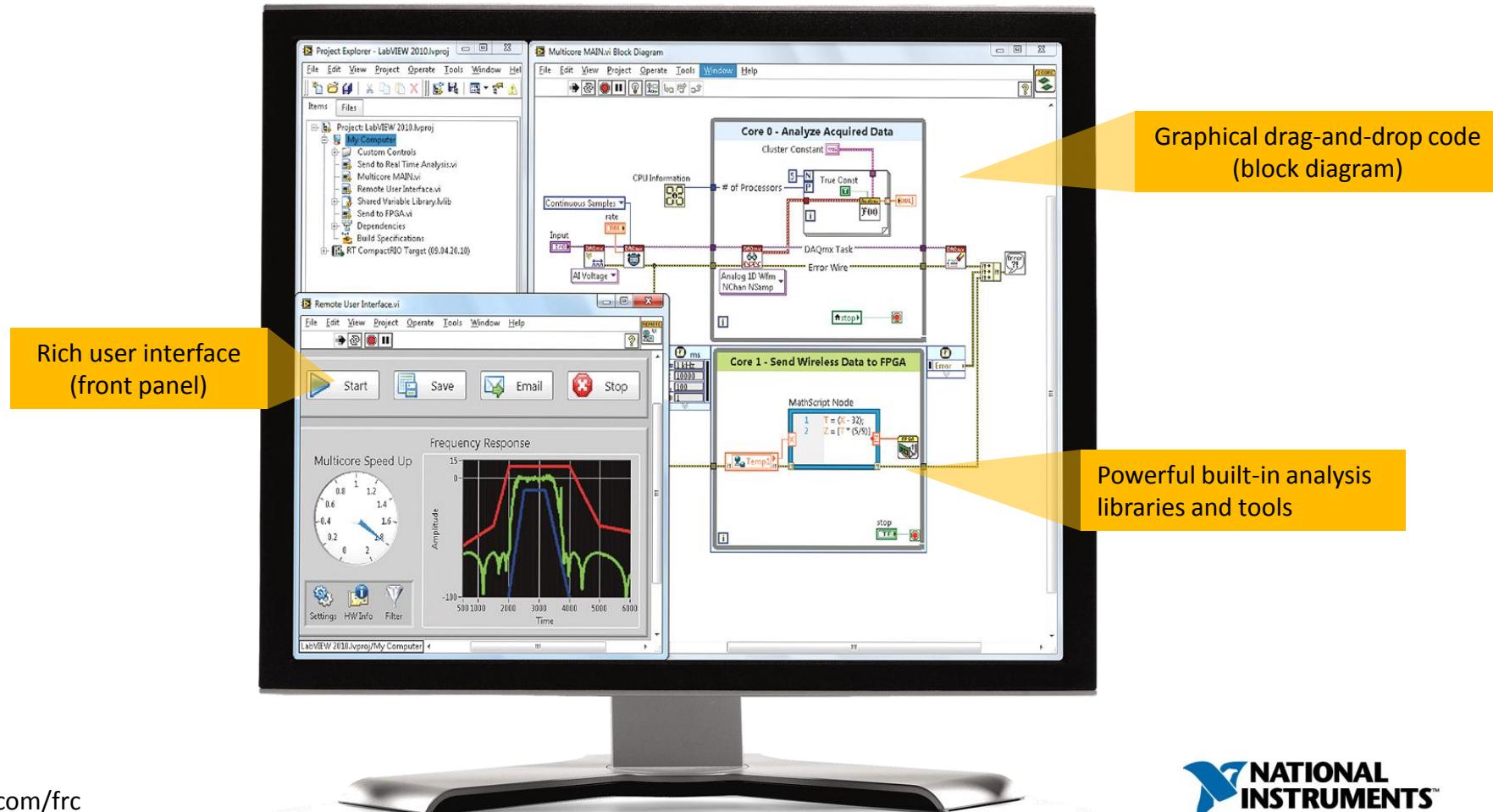
Vision,  
PID, and  
Simulation

Advanced  
Programming

- What Is LabVIEW?
- LabVIEW Development Environment
- LabVIEW Programming Fundamentals
- Tools and Troubleshooting
- Teleop and Autonomous Code

# What Is LabVIEW?

- A graphical programming environment used to develop sophisticated measurement, test, and control systems

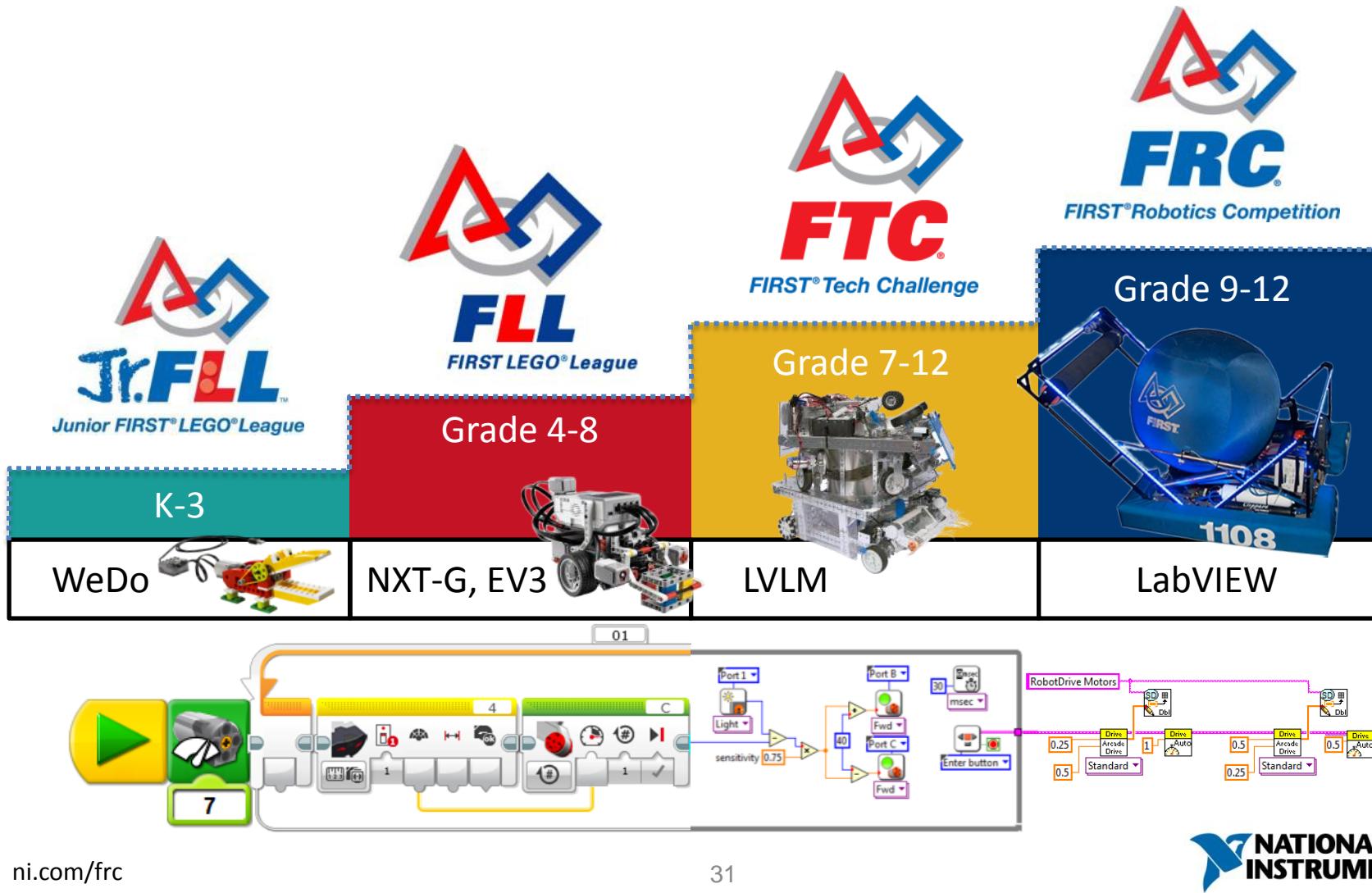


# LabVIEW Usage in Universities

110  
Countries  
7000+  
Universities



# The LabVIEW Continuum



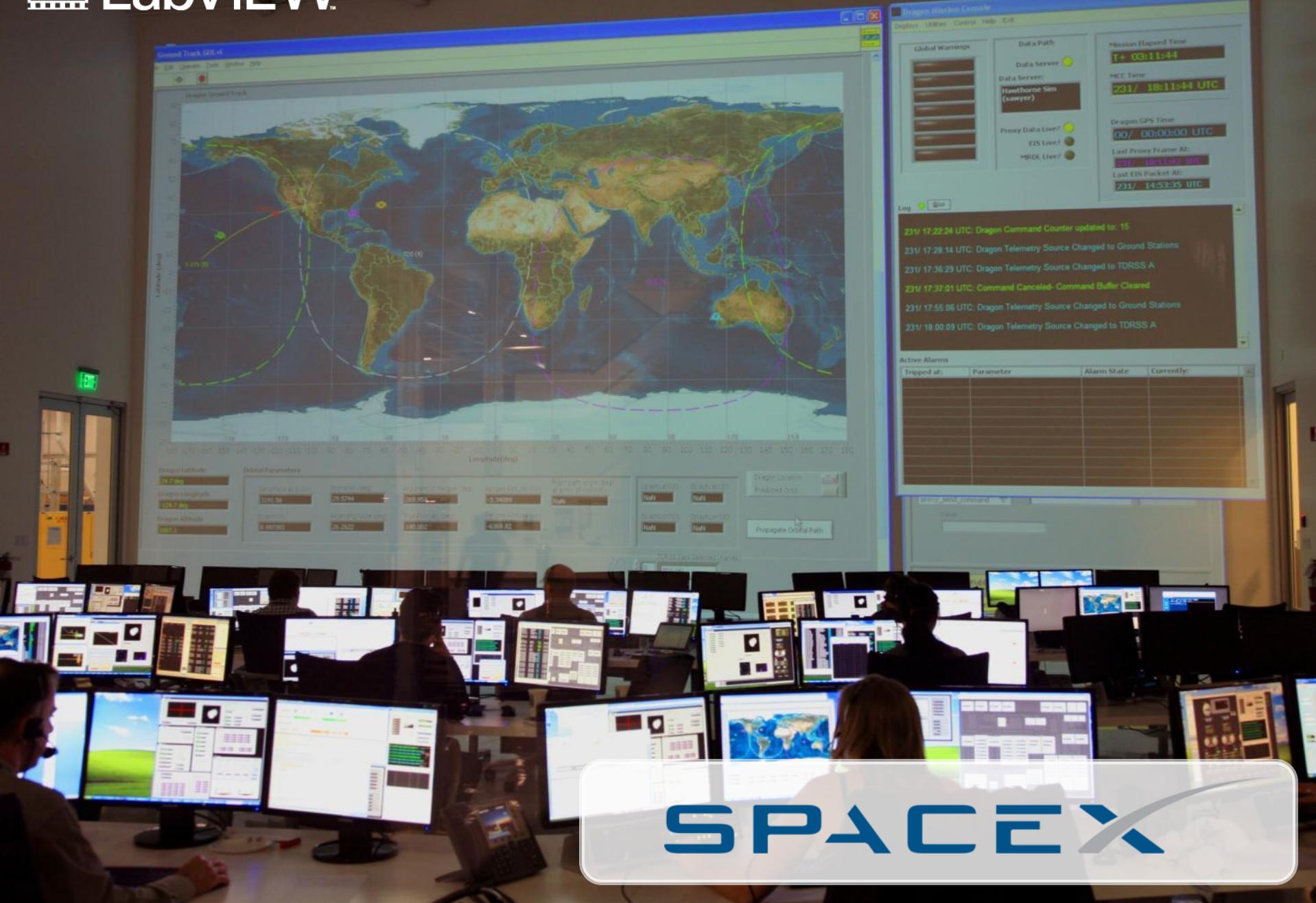
# Companies that use LabVIEW



More than 35,000 companies use NI Tools

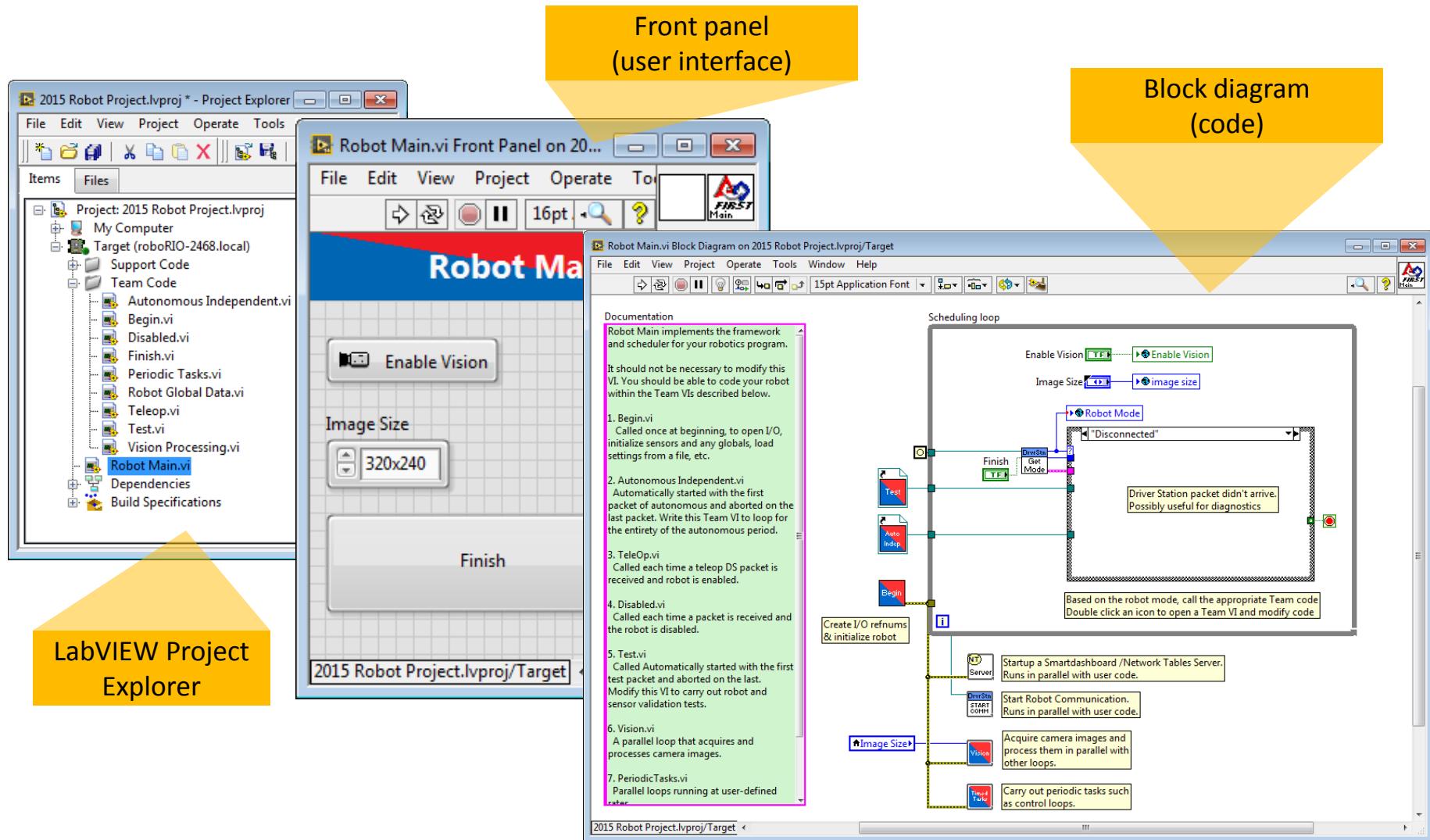


NATIONAL INSTRUMENTS  
LabVIEW™



# The LabVIEW Environment

# Parts of Robot Main.vi



# LabVIEW Project Explorer

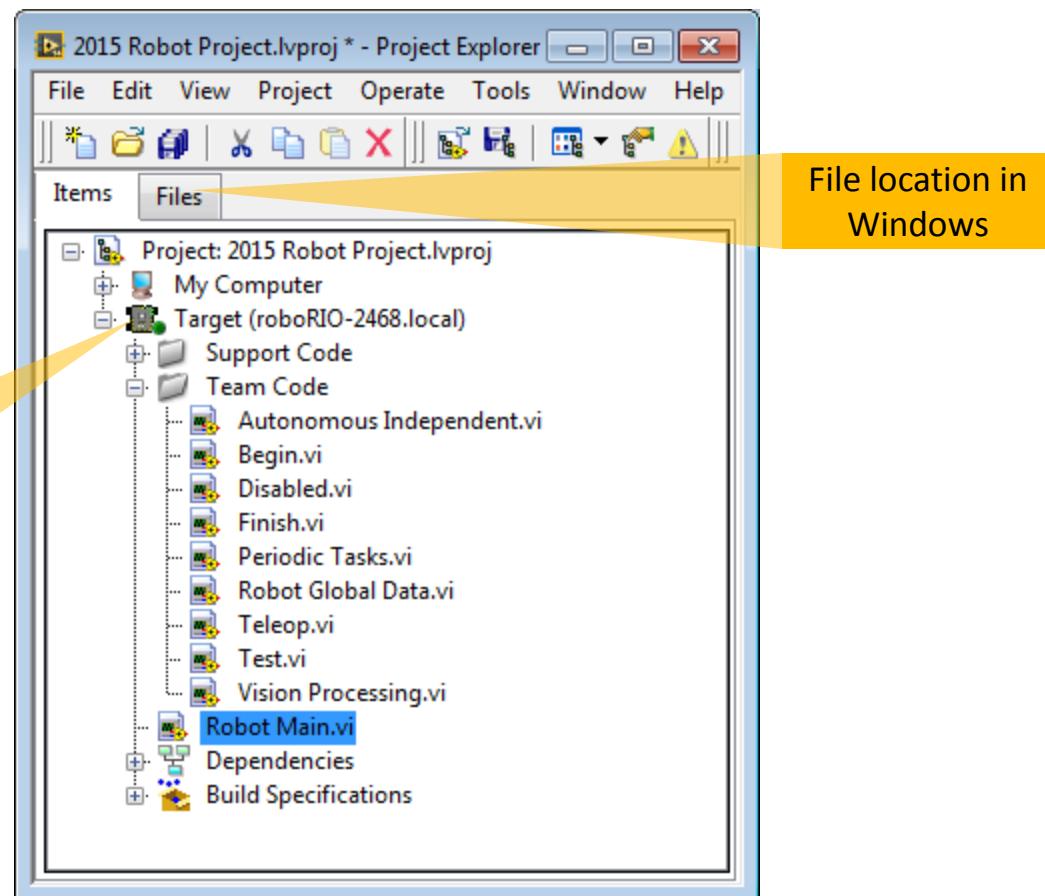
## LabVIEW File Extensions

LabVIEW project—.lvproj

Virtual instrument (VI)—.vi

Custom control—.ctl

Right-click to connect to  
roboRIO target



File location in  
Windows

# Parts of a VI: Front Panel

The front panel is the VI's graphical user interface with inputs and outputs.

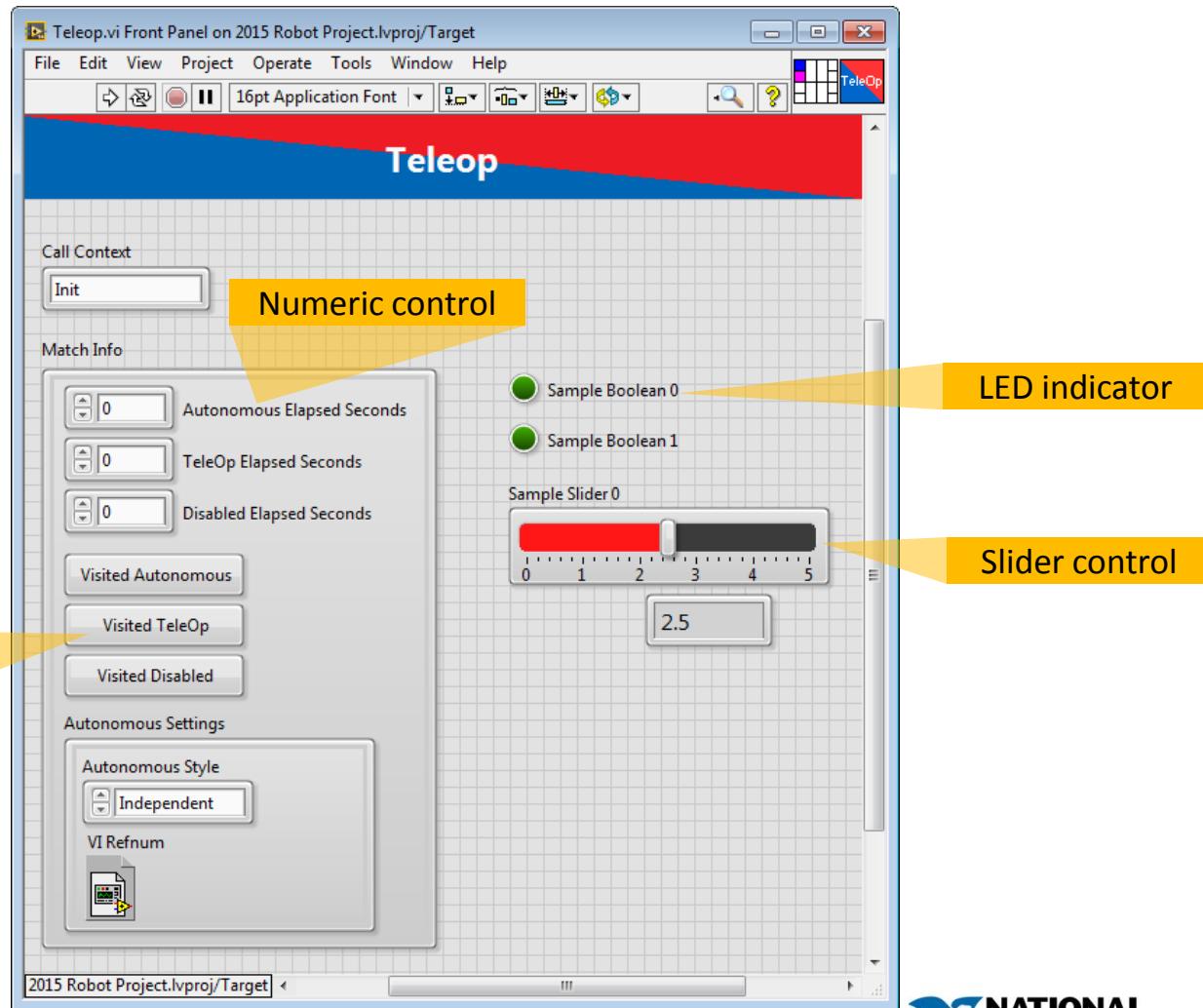
Controls (inputs)

buttons,  
knobs, slides...

Indicators (outputs)

numeric display,  
graphs, LEDs...

Boolean (on/off)  
control



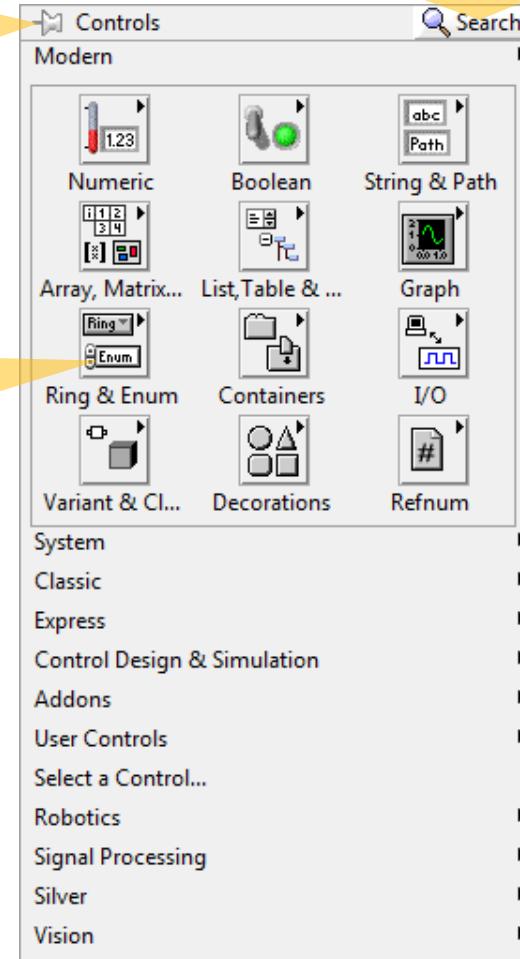
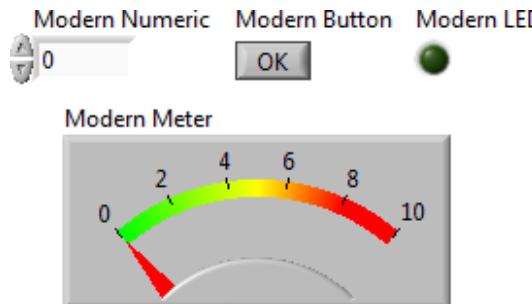
# Controls Palette

Right-click on the front panel (not the block diagram) to open the Controls Palette

Pin to keep open

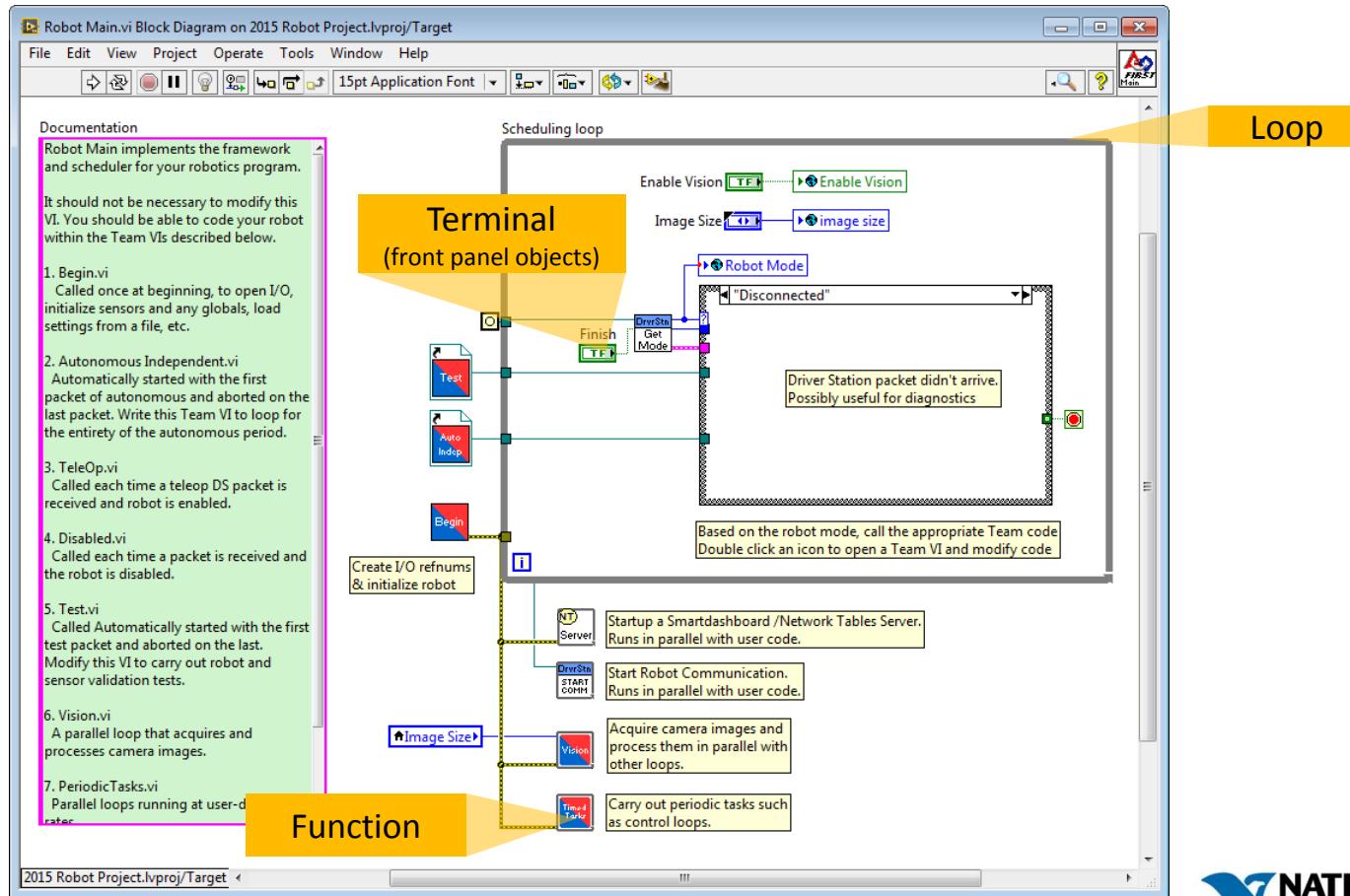
Search for controls

Browse subpalettes

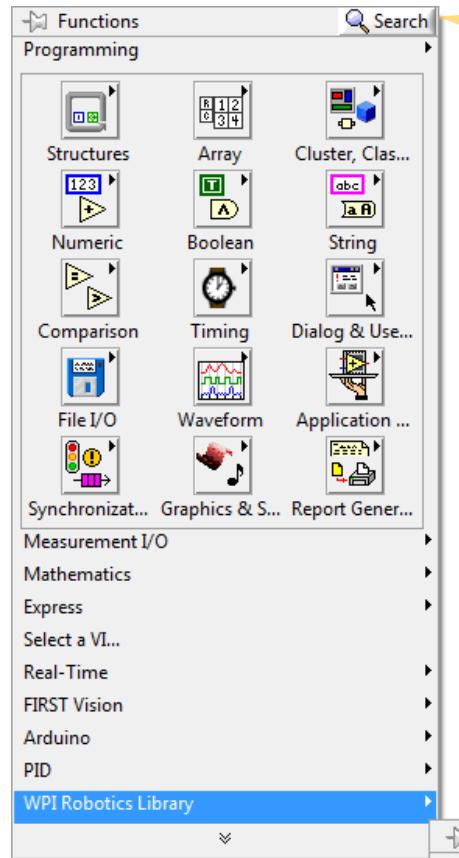


# Parts of a VI: Block Diagram

- The block diagram is the code for each VI. In LabVIEW, graphical blocks are connected with wires to control the execution.

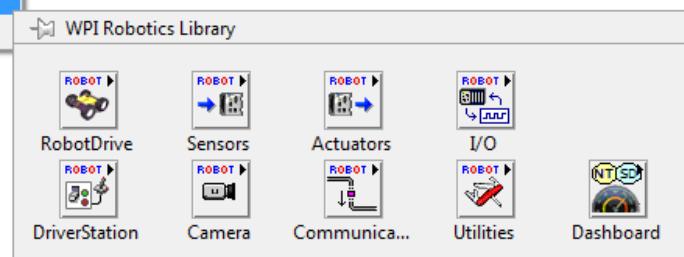


# Functions Palette



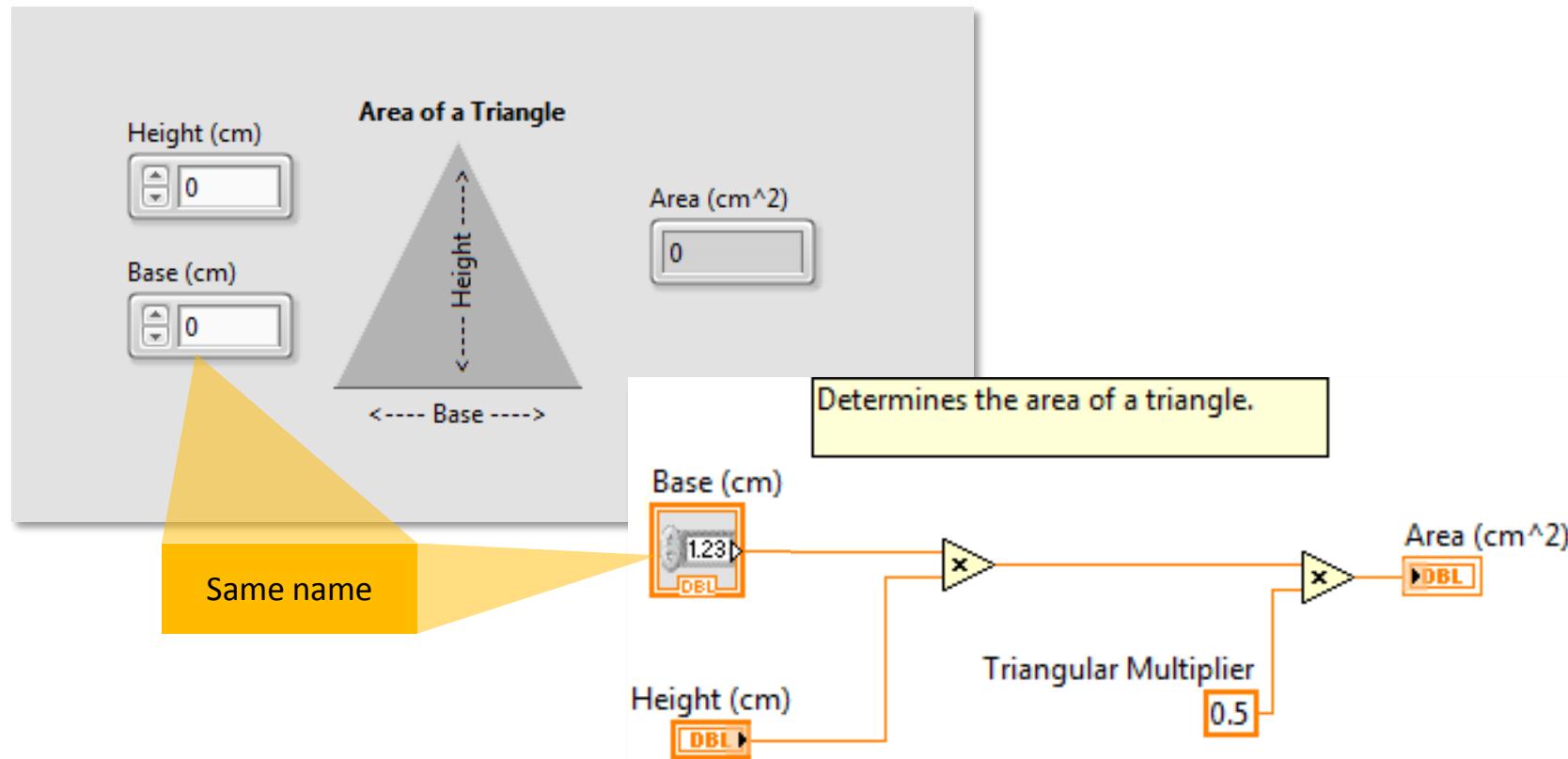
Search for functions by name

- Right-click on the **block diagram** (not the front panel) to open the Functions Palette
- Contains the VIs, functions, and constants you use to create the block diagram
- FRC-specific palettes include the WPI Robotics Library, FIRST Vision, and PID



# Terminals

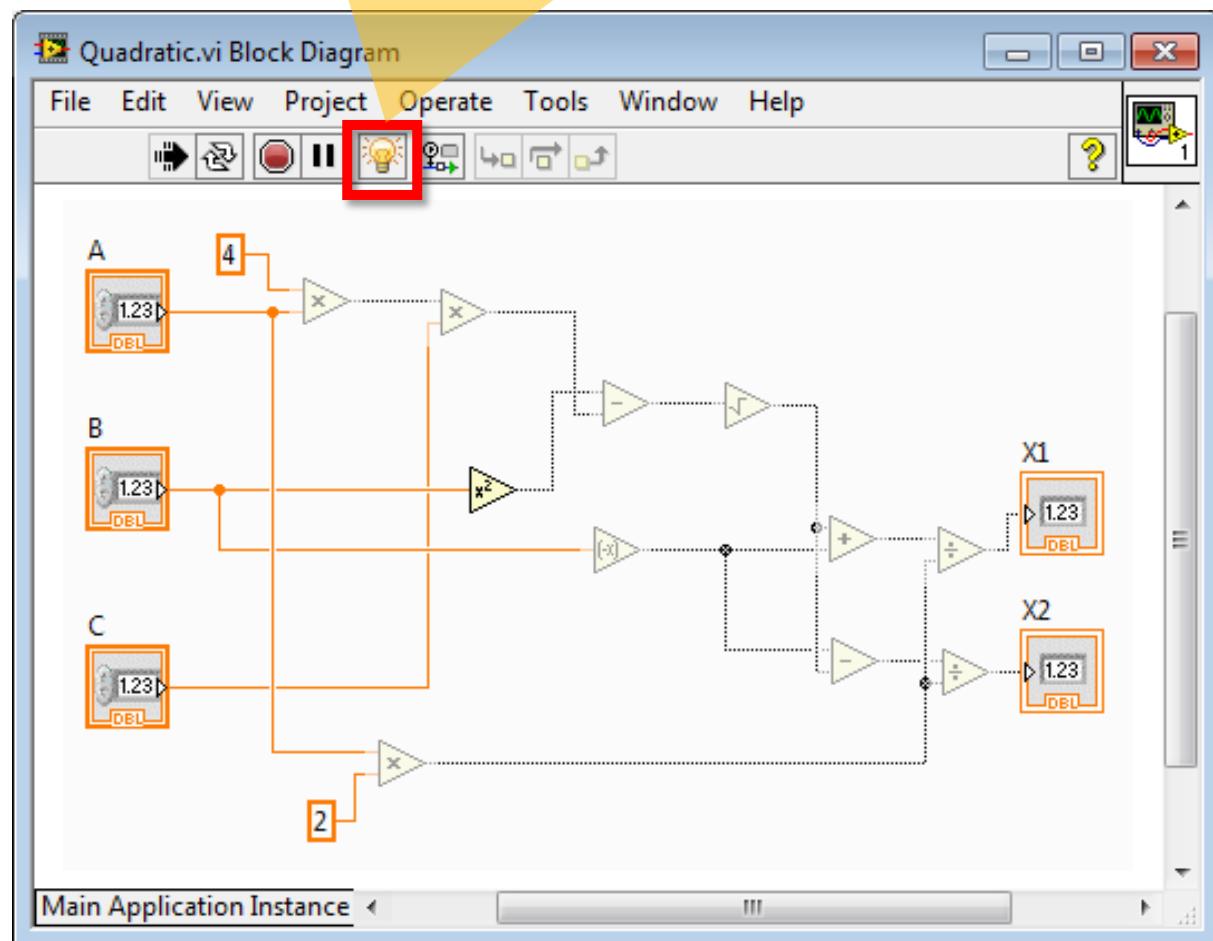
- With terminals, you can read and write to front panel controls and indicators.



# Dataflow Programming

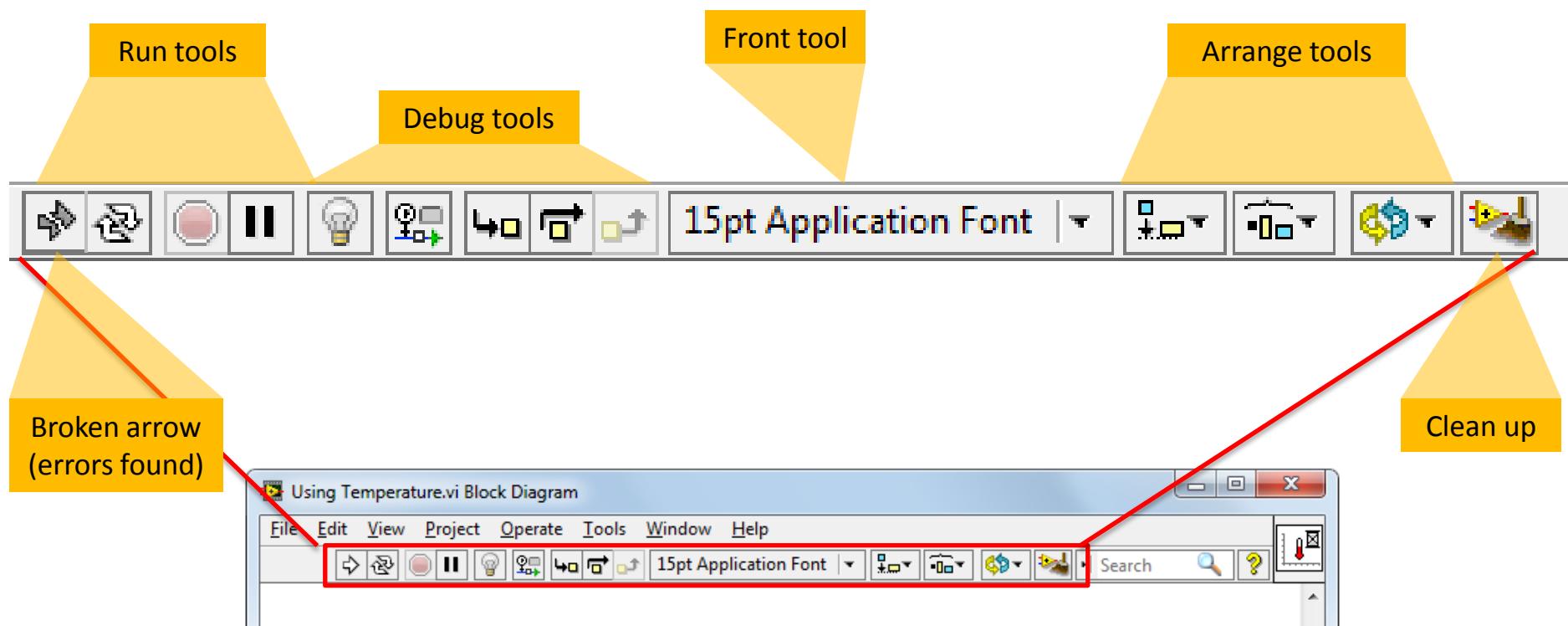
Turn on  
highlight  
execution

- Order of execution is controlled by how the wires are connected
- Each function executes when all of the inputs are ready
- Parallel code executes at the same time



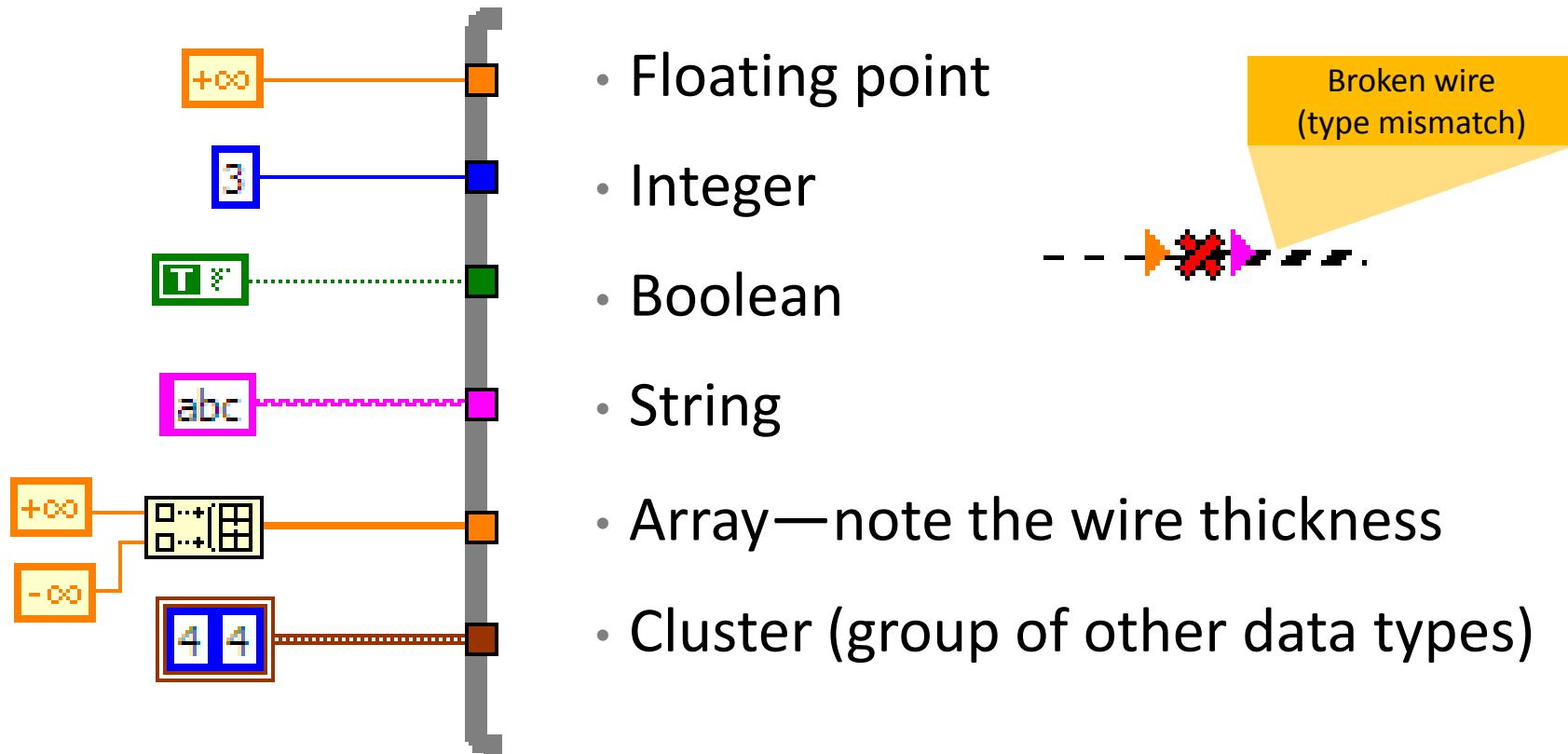
# LabVIEW Toolbar

- The tools you need to run, debug, clean up, and edit your code



# Wires (Data Types)

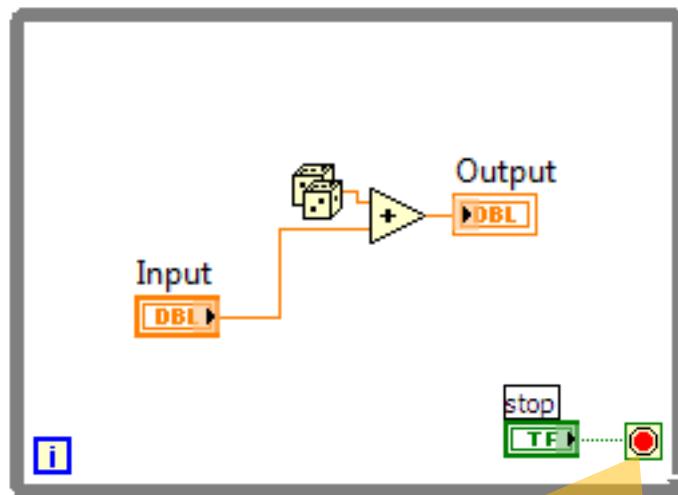
- Other examples of wires in LabVIEW



# Loops

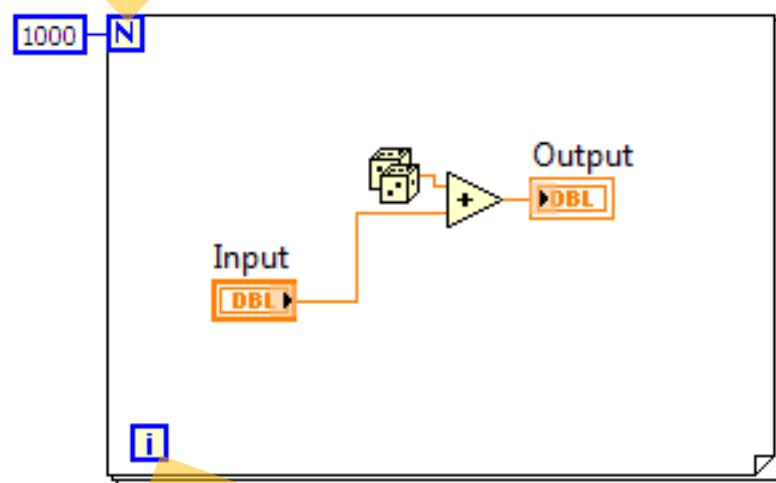
- Allow the same piece of code to run multiple times
- Exit conditions are different for each

## While Loop



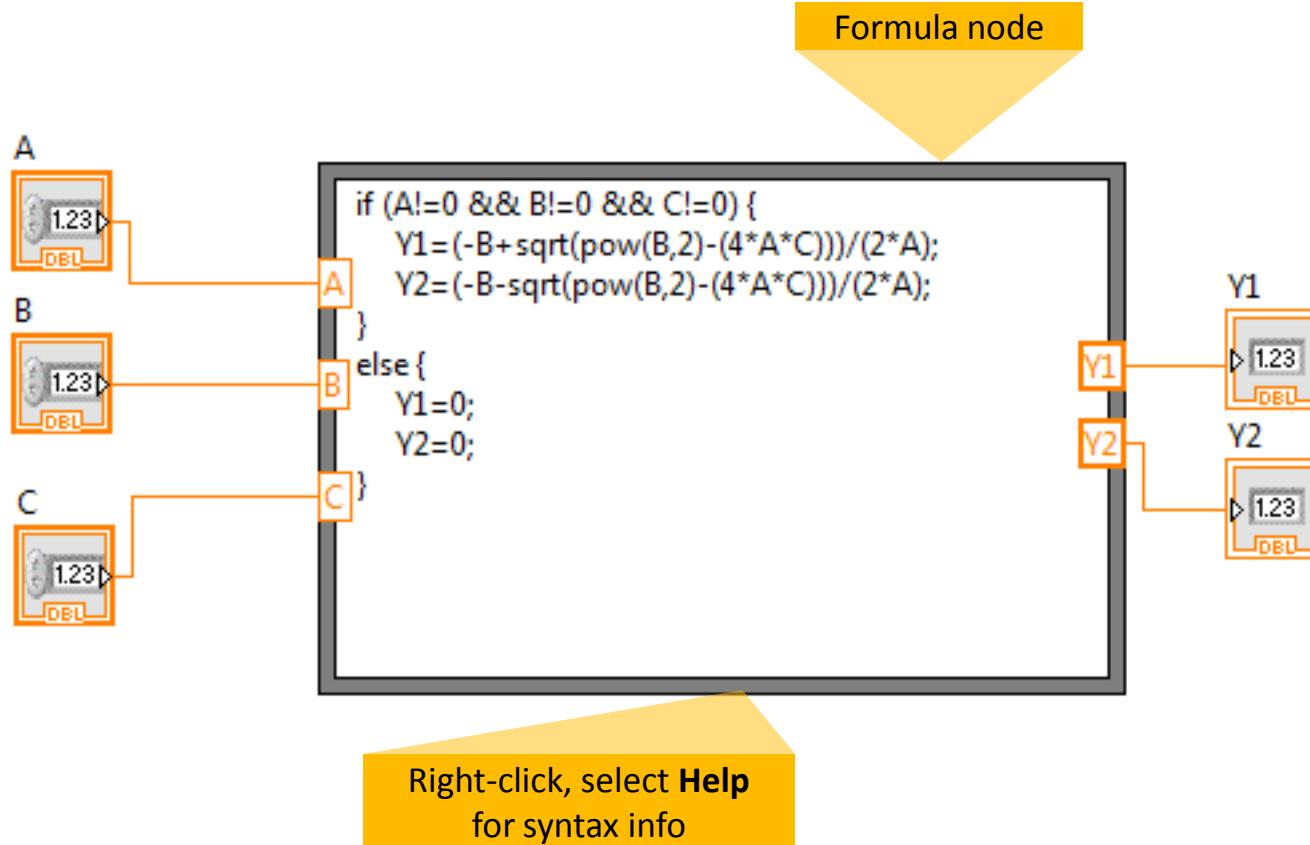
Run until stop  
condition met

## For Loop



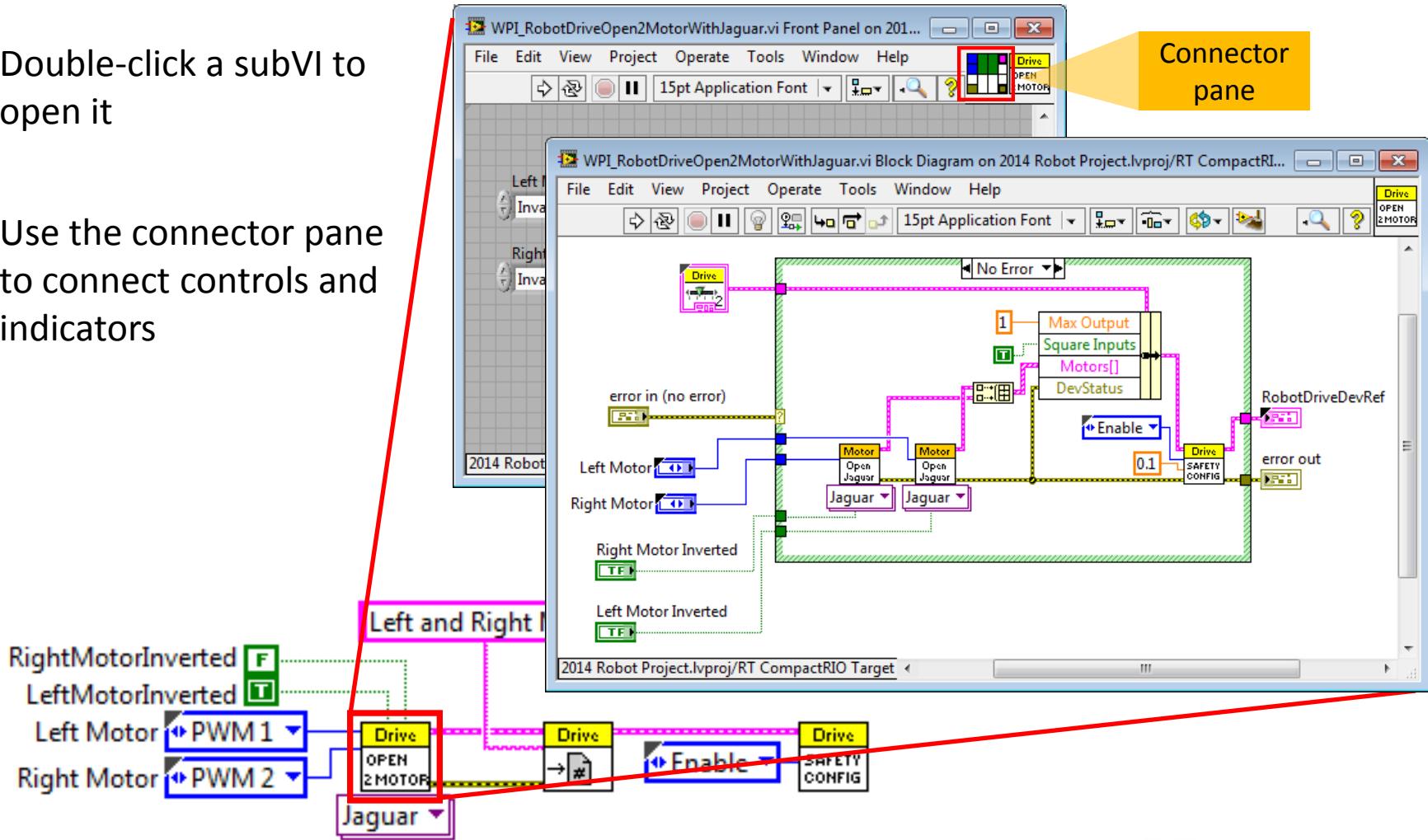
Iteration counter

# Text-Based Math in LabVIEW



# SubVIs

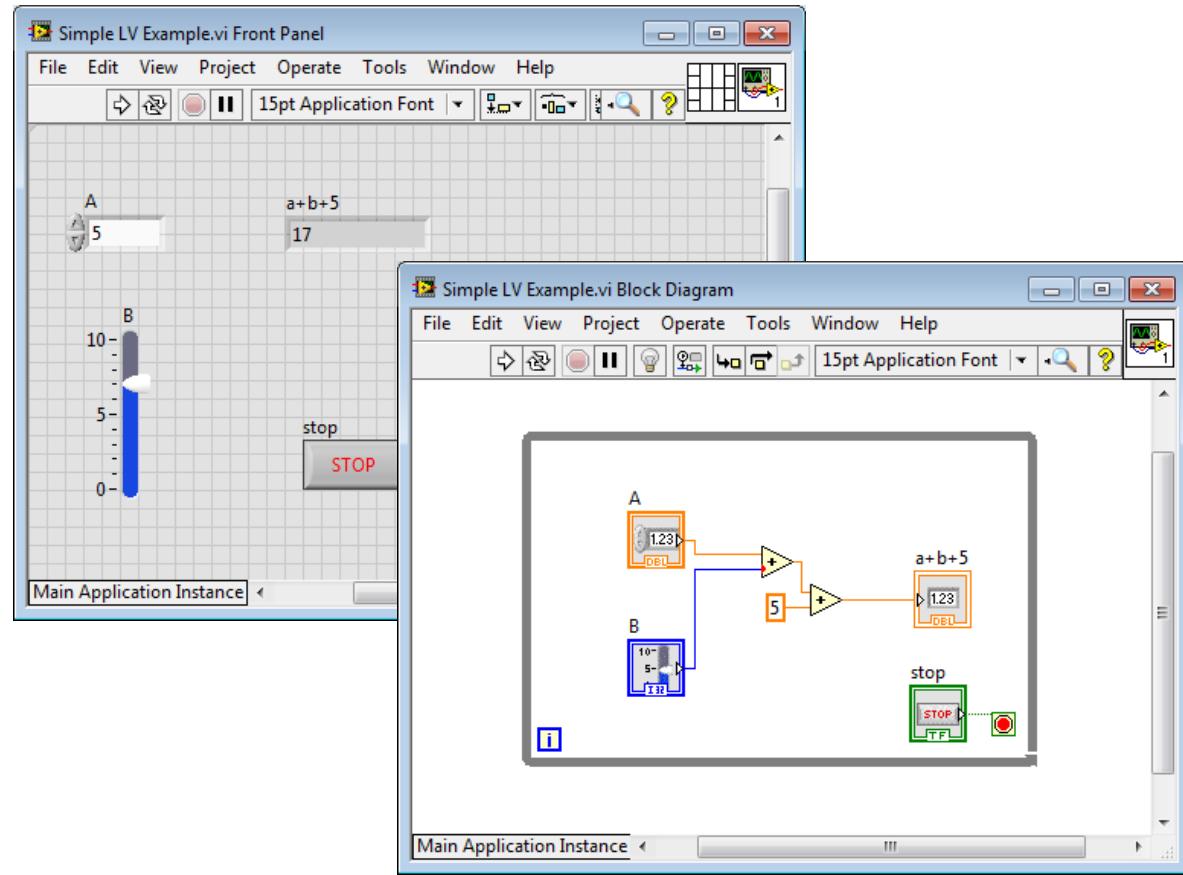
- Double-click a subVI to open it
- Use the connector pane to connect controls and indicators



# LabVIEW Programming Demonstration

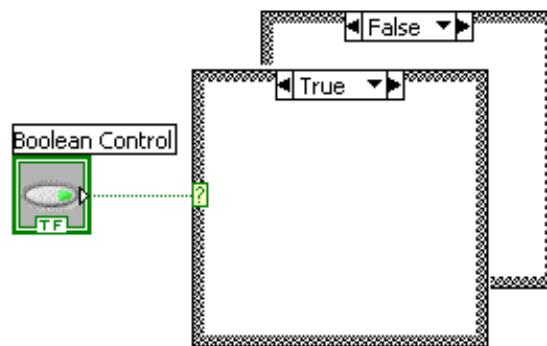
## Programming Demonstration

- Front panel/block diagram
- Tools/functions palettes
- Controls/terminals
- Wiring
- Constants
- Dropping functions
- Run/run continuously
- Loops
- Data types
- Highlight execution

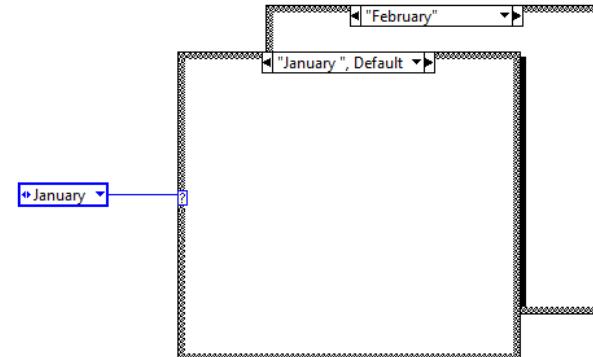


# How Do I Make Decisions in LabVIEW?

## 1. Case Structures

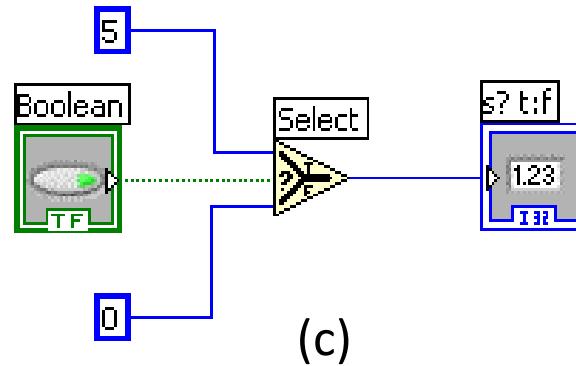


(a)



(b)

## 2. Select



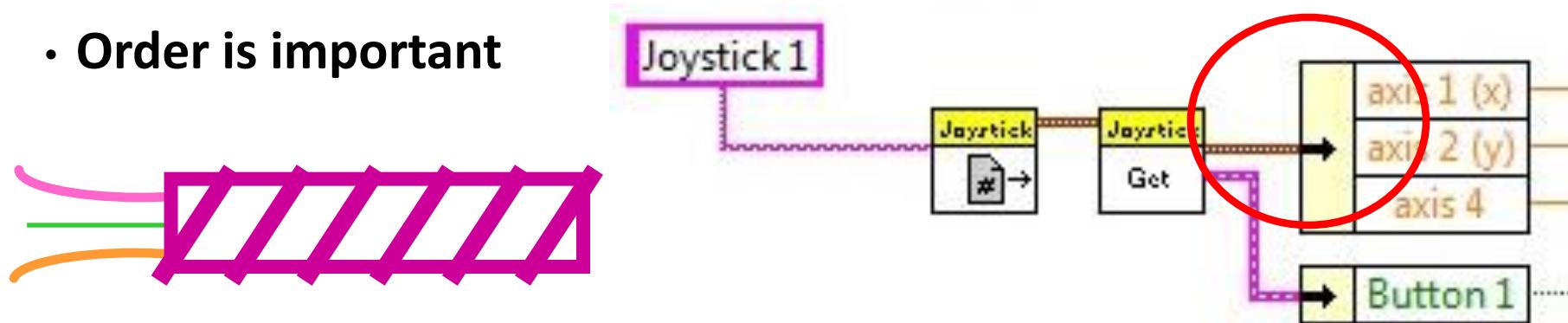
(c)

# Introduction to Clusters

- Data structure that groups data together
- Data may be of different types
- Elements must be either all controls or all indicators
- Thought of as wires bundled into a cable

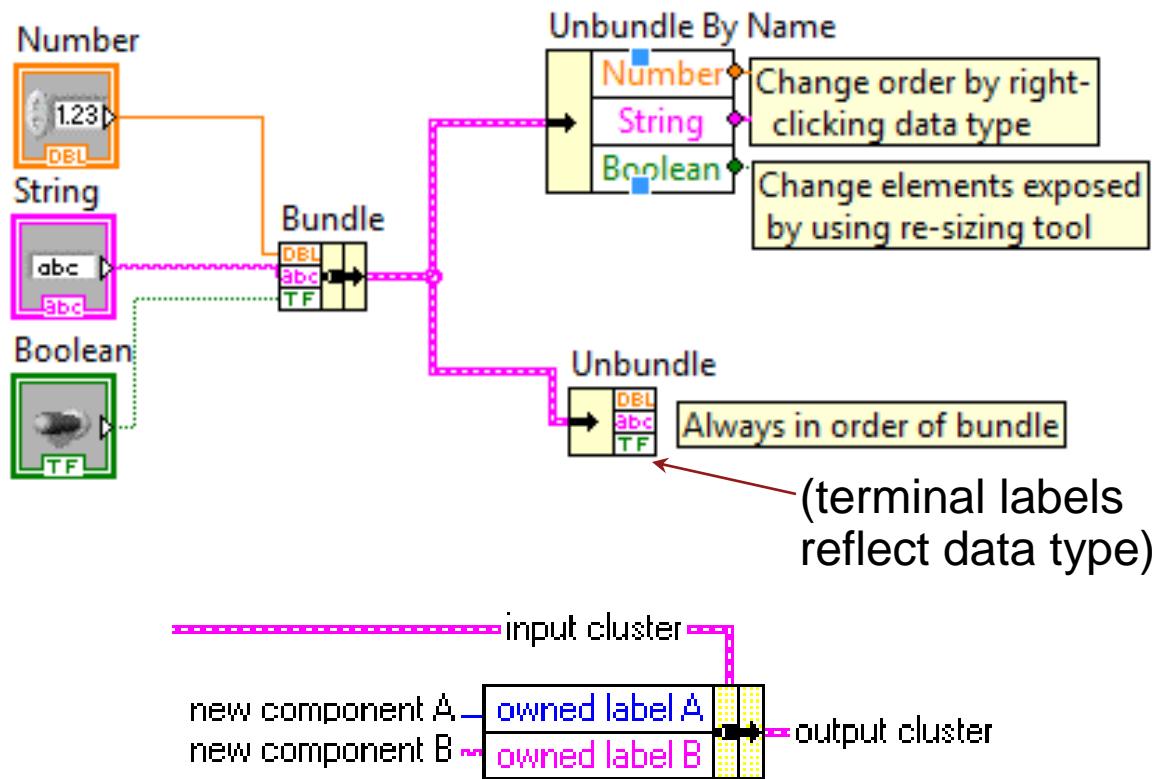
Easier to bundle wires together and pass around as a group than wire each individually

- **Order is important**



# Cluster Functions

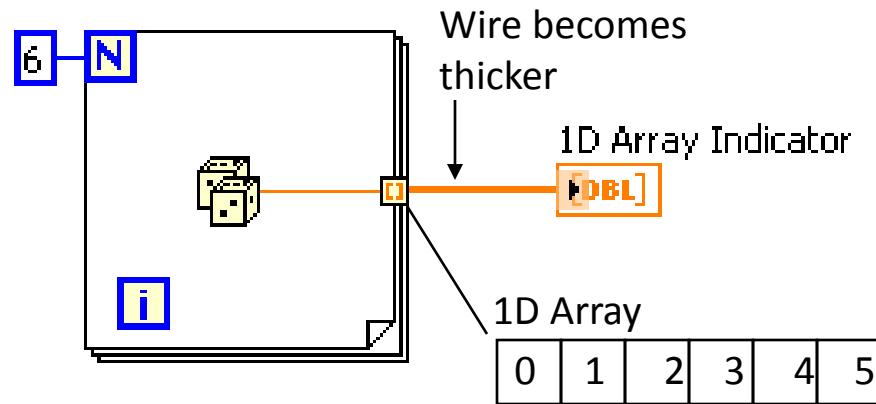
- In the Cluster & Variant subpalette of the Programming functions palette
- Can also be accessed by right-clicking the cluster terminal



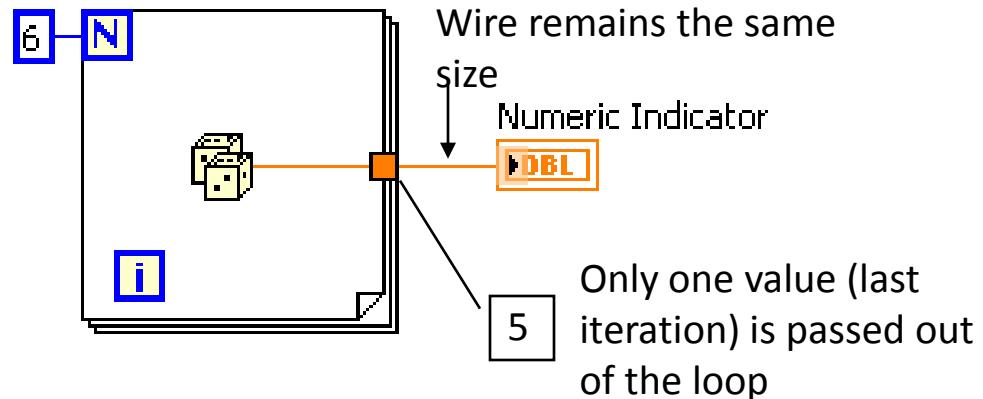
# Building Arrays With Loops

- Loops can accumulate arrays at their boundaries with auto-indexing
- For Loops auto-index by default
- While Loops output only the final value by default
- Right-click tunnel and enable/disable auto-indexing

Auto-Indexing Enabled

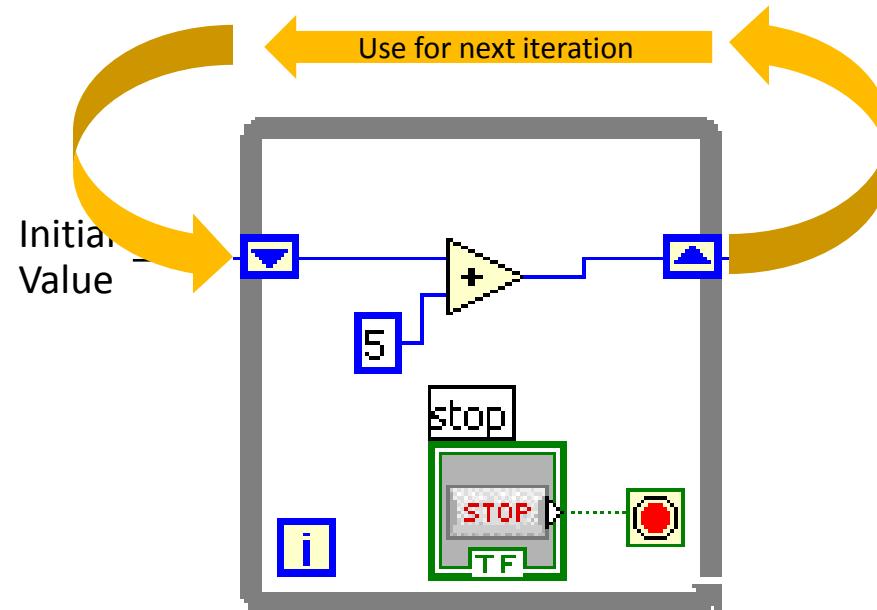


Auto-Indexing Disabled



# Shift Register: Access Previous Loop Data

- Available at left or right border of loop structures
- Right-click the border and select Add Shift Register
- Right terminal stores data on completion of iteration
- Left terminal provides stored data at beginning of next iteration

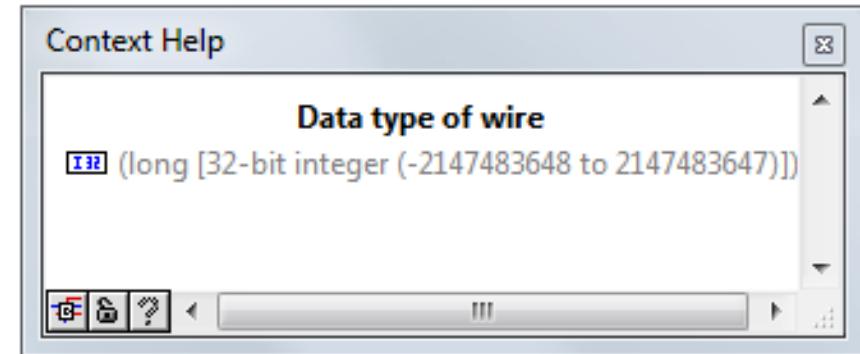


# Context Help

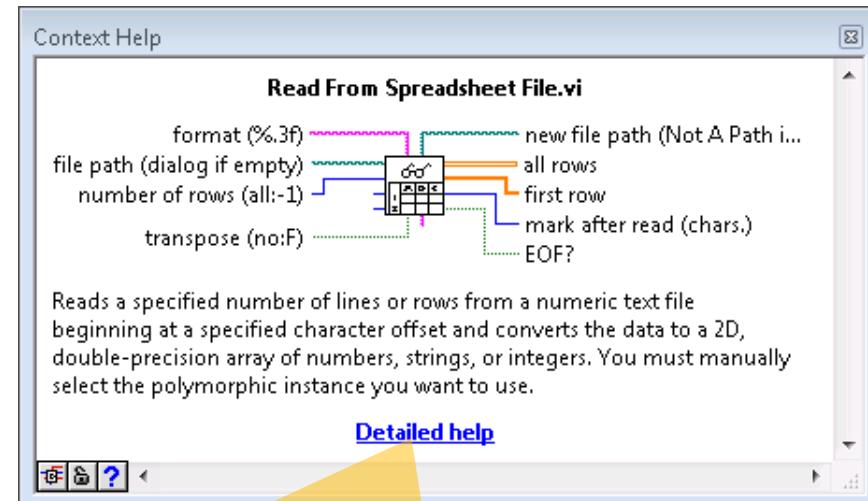
Click ? to open context help



- Displays basic information about wires and nodes when you hover over with your mouse



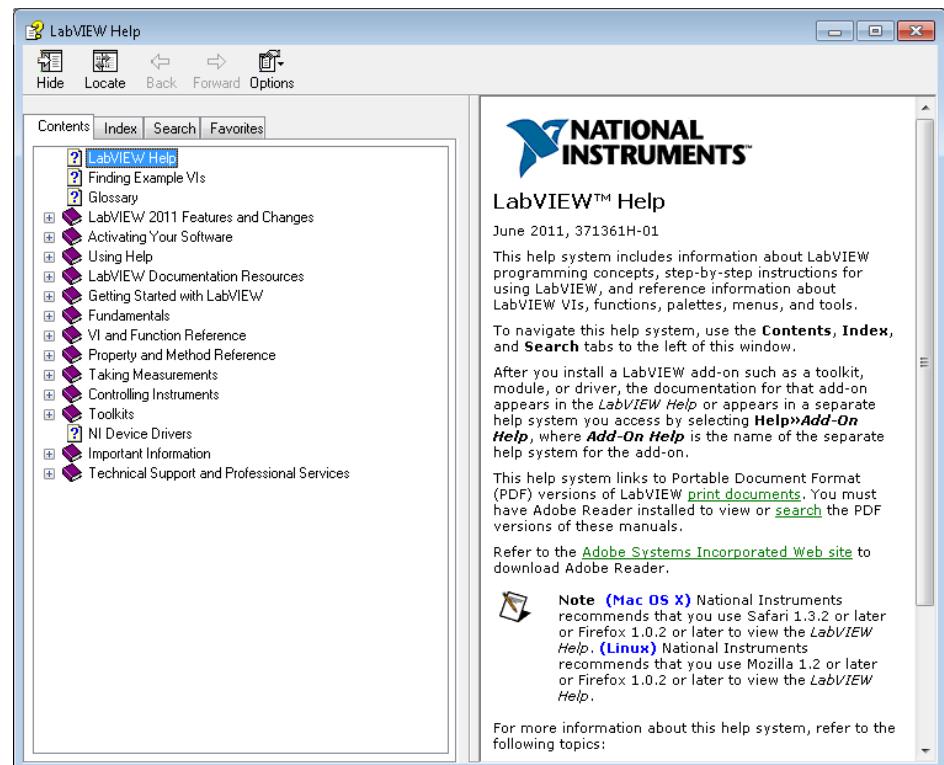
- Turn context help on/off
  - Click the yellow ? on the top right corner of your VI
  - Press <Ctrl-H>
  - Select **Help»Show Context Help** from the LabVIEW menu



Click for more info

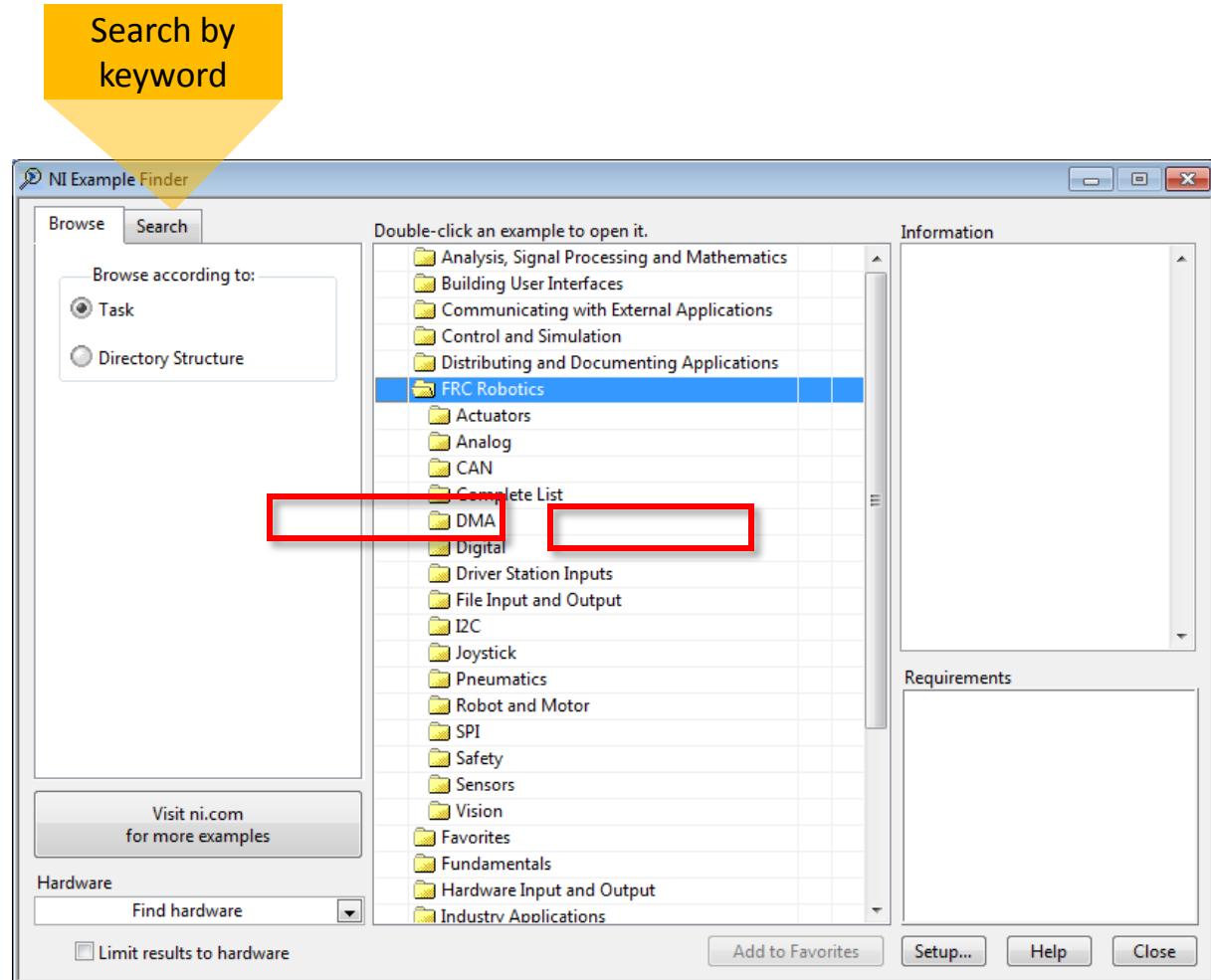
# LabVIEW Help

- Contains detailed descriptions and instructions for most palettes, menus, tools, VIs, and functions.
- Open LabVIEW Help by
  - Selecting **Help»LabVIEW Help** from the menu
  - Clicking the **Detailed help** link in the **Context Help** window
  - Right-clicking an object and selecting **Help** from the shortcut menu
  - Pressing **F1**



# Find Examples

- Find FRC templates in the Support Tab»Find FRC Examples...
- Or Help»Find Examples and browse to the FRC Robotics Folder

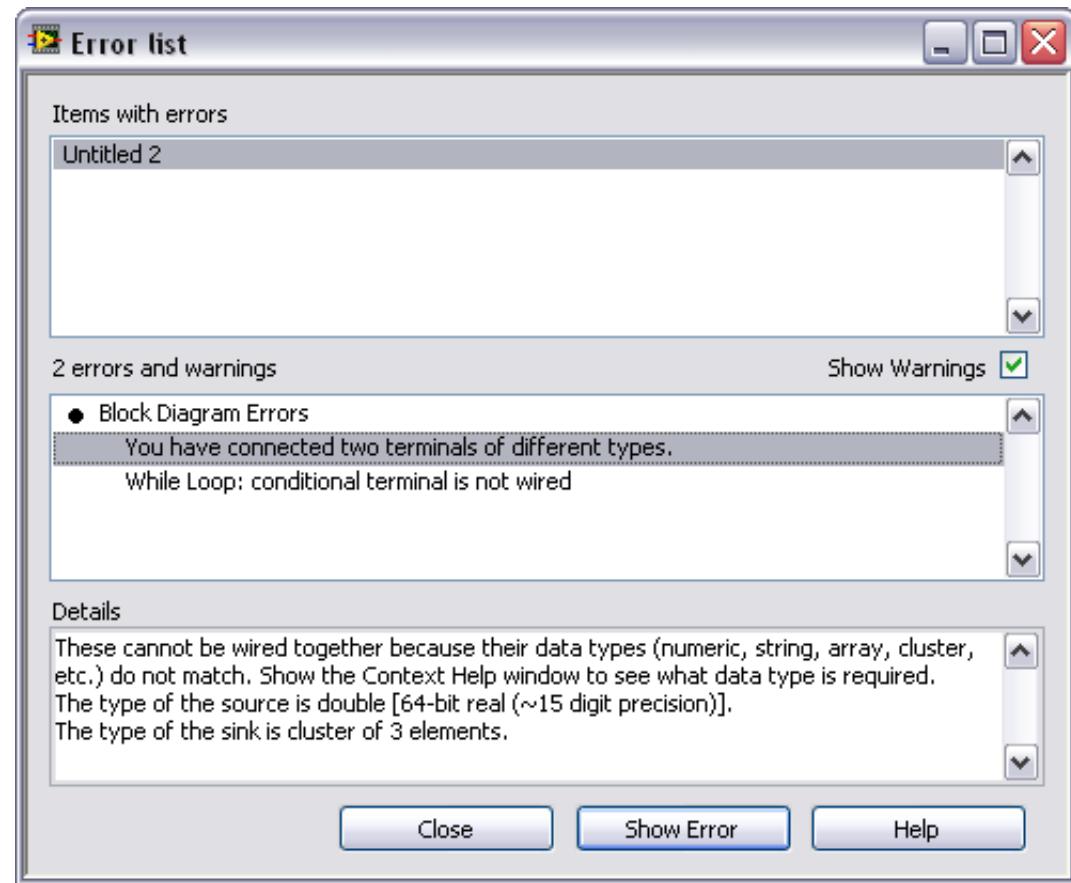


# Debugging: Correcting Broken VIs

Broken Run arrow»VI cannot be compiled»VI cannot be executed

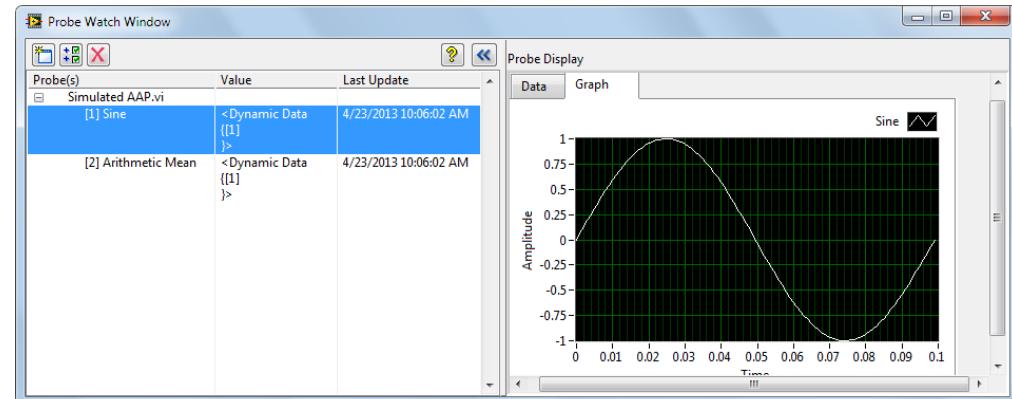
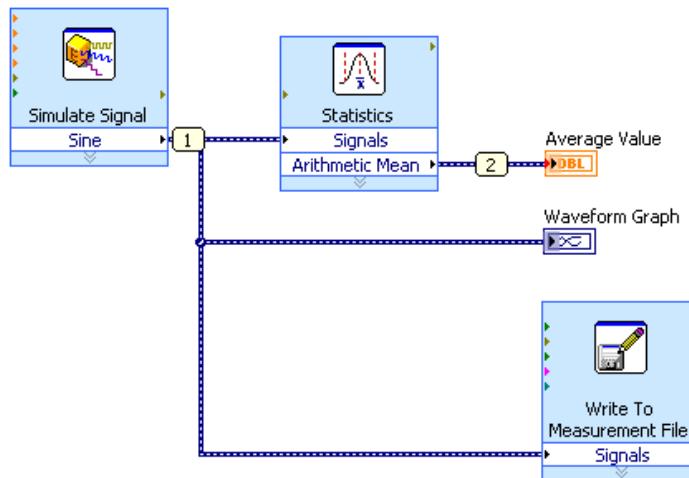
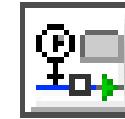
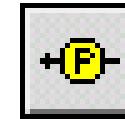


Click to open  
Error List



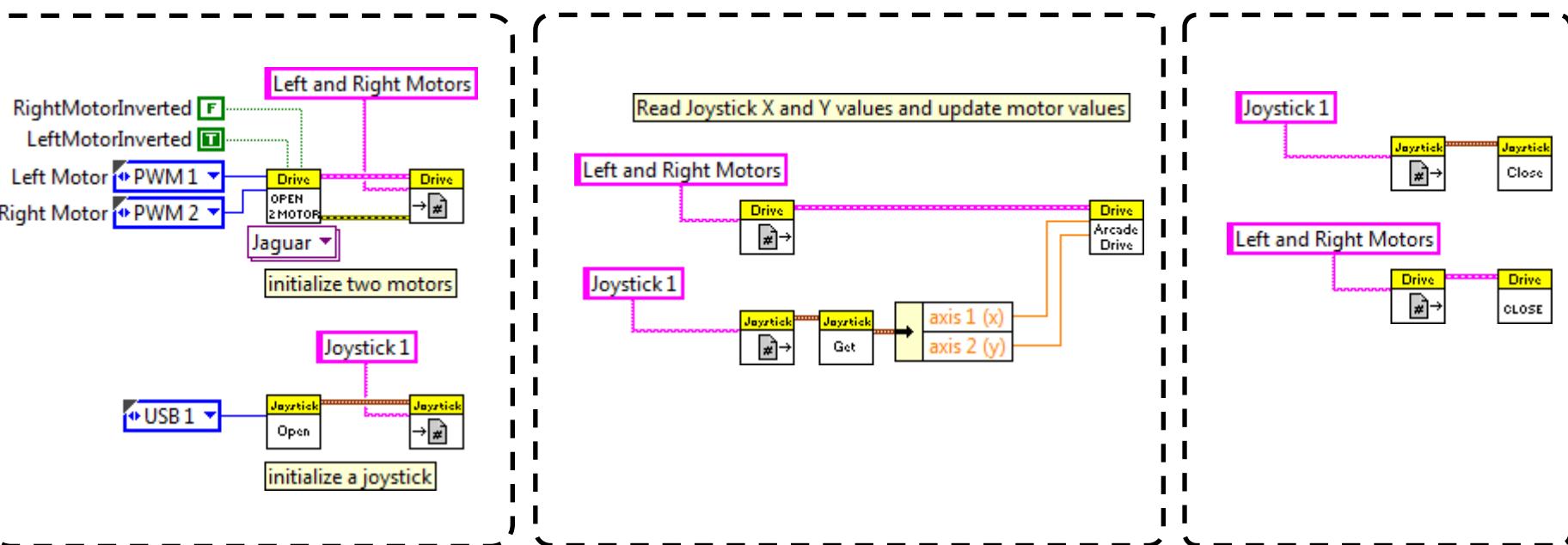
# Debugging: Probes

- Use the Probe tool to observe intermediate data values and check the error output of VIs and functions, especially those performing I/O.
- Specify to retain the values in the wires so that you can probe wires for data after execution.



# Teleop Code

- When modifying the FRC Robot Project template, do **not** modify Robot Main.vi.
- Modify the code within the Team Code folder within the project. This code is called on in Robot Main.vi



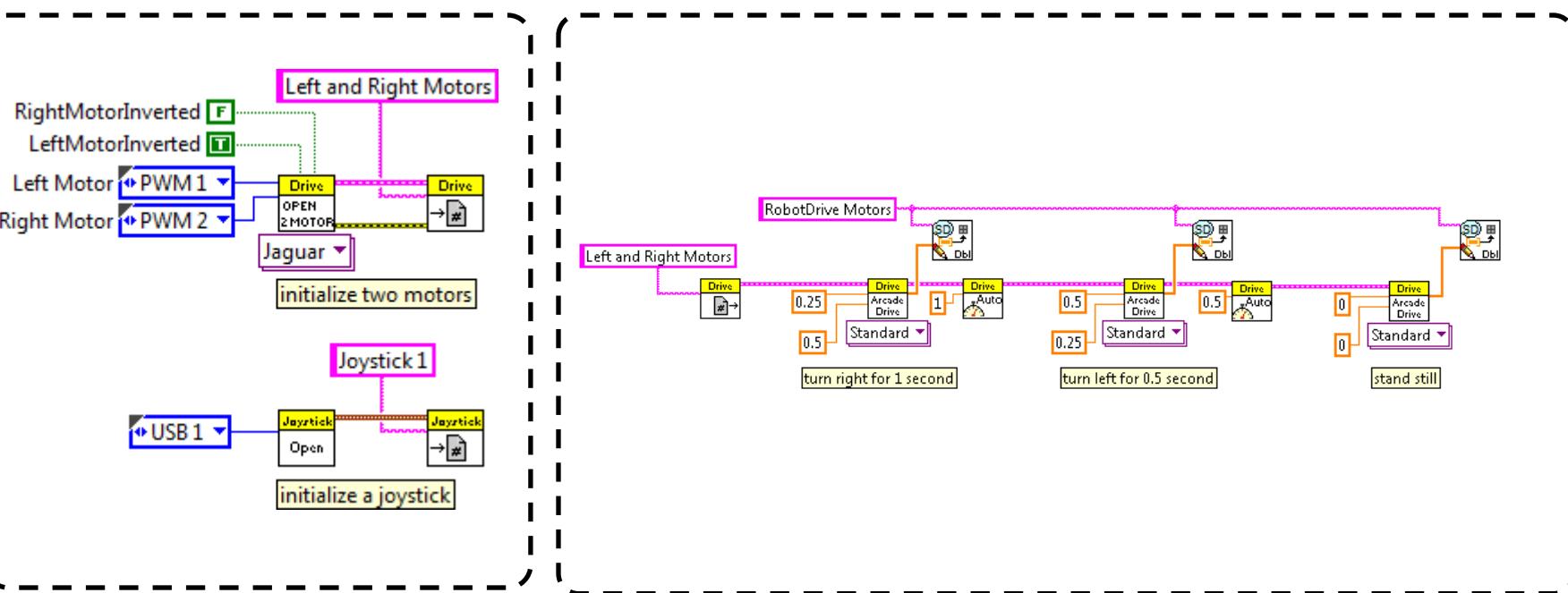
Begin.vi

Teleop.vi

Finish.vi

# Autonomous Code

- Similar to Teleop, Autonomous executes code by references.



Begin.vi

Autonomous Independent.vi

# Learn Your Hotkeys



NATIONAL INSTRUMENTS

# LabVIEW™ Quick Reference Guide

Keyboard Shortcuts			Tools Palette					
File	Ctrl-X	Cut object	Right Click	Display controls/functions palette	Tool	Icon	Description	
Ctrl-N	Create new VI	Ctrl-Z	Undo last action	Shift-Right Click	Display tools palette	Automatic Tool Selection		Automatically choose the appropriate tool
Ctrl-S	Save VI	Ctrl-Shift-Z	Redo last action	Ctrl-T	Tile block diagram and front panel windows	Operating Tool		Change the value of a control or select the text within a control
Ctrl-P	Print	Operate		Help	Display context help	Positioning Tool		Position, resize, and select objects
Ctrl-V	Paste object	Ctrl-R	Run VI	Ctrl-H	Display context help	Labeling Tool		Edit text and create free labels
Ctrl-U	Clean up diagram	Ctrl-.	Abort VI			Wiring Tool		Wire objects together on a block diagram
Ctrl-Space	Activate quick drop	Window						
Ctrl-B	Remove broken wires	Ctrl-E	Display block diagram/front panel			Scroll Tool		Scroll the window without using the scroll bars
Ctrl-C	Copy an object					Breakpoint Tool (used for debugging)		Set breakpoints on VIs, functions, wires, loops, sequences, and cases

Editing Tools			Debugging Tools		
Tool	Icon	Description	Tool	Icon	Description
Show Context Help		Display the context help window	Run		Execute the VI
Text Settings		Change the font setting for the VI, including size, style, and color	List Errors		List errors that prevent the VI from running
15pt Application Font					
Align Objects		Align selected objects	Run Continuously		Execute the VI continuously until abort or pause is pressed
Distribute Objects		Space objects evenly	Stop		Stop VI execution immediately
Resize Objects		Resize multiple front panel objects to the same size	Execution Highlighting		Animate data movement on the block diagram wires
Reorder		Reorder the layers of the objects	Pause		Temporarily stop execution to debug a portion of the VI
Clean Up Diagram		Rearrange wires and objects on the block diagram	Step Into		Single-step into a subVI or structure to debug it
Enter		Appears when a new value is available to replace an old value	Step Over		Execute a subVI or structure and resume single-stepping in next main function
			Step Out		Execute a subVI or structure and resume single-stepping in calling VI or structure

# Setup Resources

- [Learn LabVIEW—Video Training](#)
- [LabVIEW Environment Overview](#)
- [Creating Custom Controls](#)
- [LabVIEW Dataflow](#)
- [LabVIEW Tools](#)
- [LabVIEW Data Structures](#)
- [LabVIEW Debugging](#)
- [WPI Library Overview](#)

All links available at  
[ni.com/frc](http://ni.com/frc) or  
click this button



- Forum Support (year round)
  - [ni.com/frc](http://ni.com/frc)
- Phone Support (during build season only)
  - (866) 511-6285 from 1:00 to 7:00 pm (Central Time)

# FRC LabVIEW Quick Start Guide

Setup

NI LabVIEW  
Basics

Vision,  
PID, and  
Simulation

Advanced  
Programming

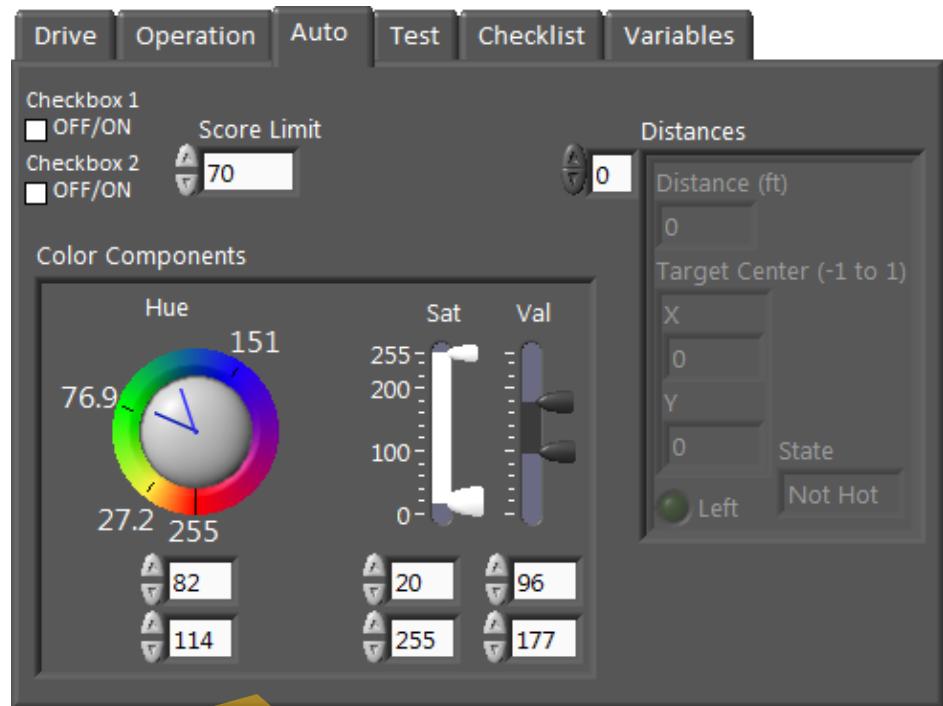
- Vision
  - Code Overview
  - NI Vision Assistant
- PID Control
  - What Is It?
  - LabVIEW PID
- Robot Simulation

# Vision

NI provides vision tools to make your robot smarter.

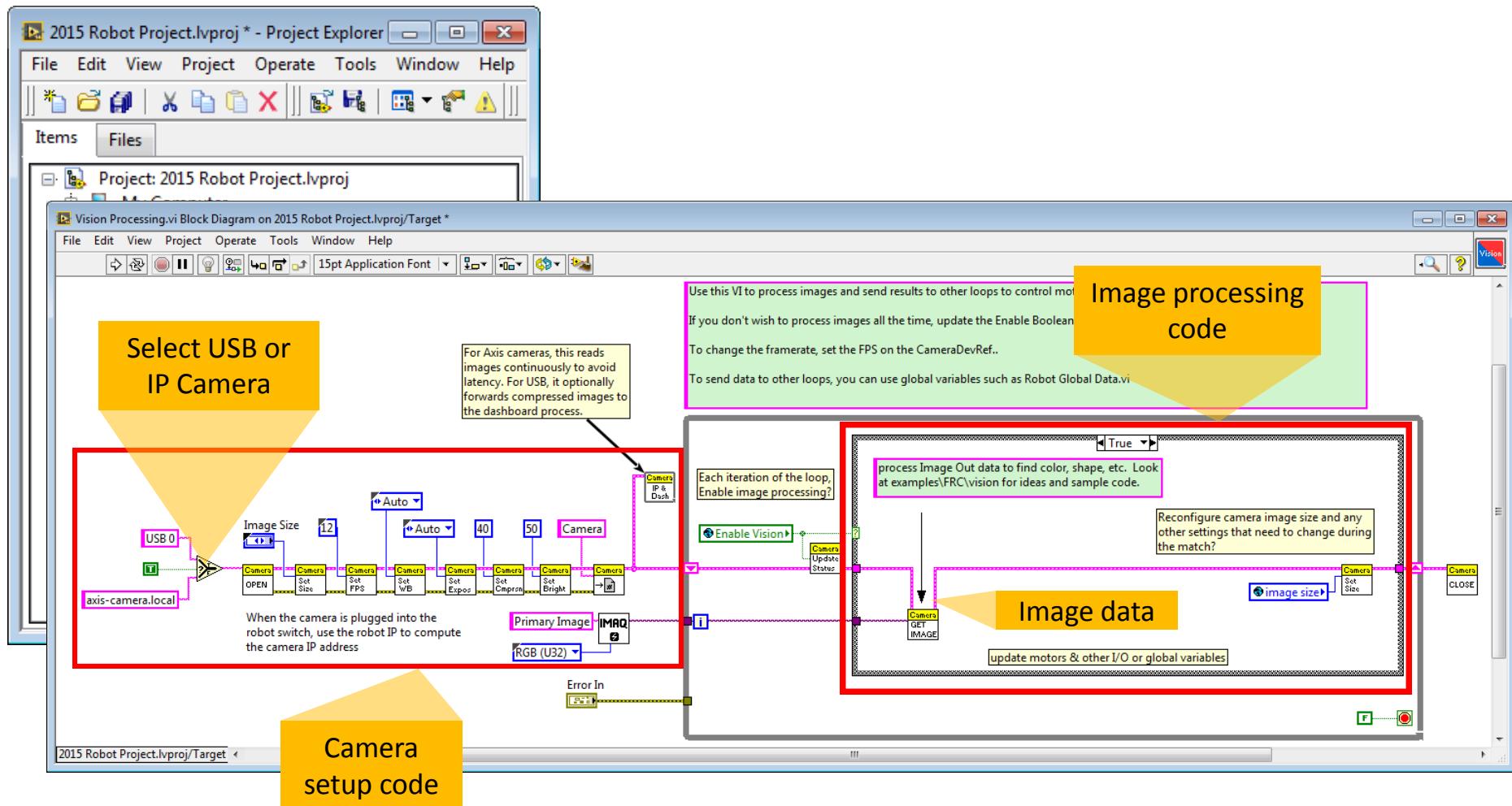
## Vision Uses

- Locate colored objects
- Locate shapes
- Locate game elements
- Calculate distances

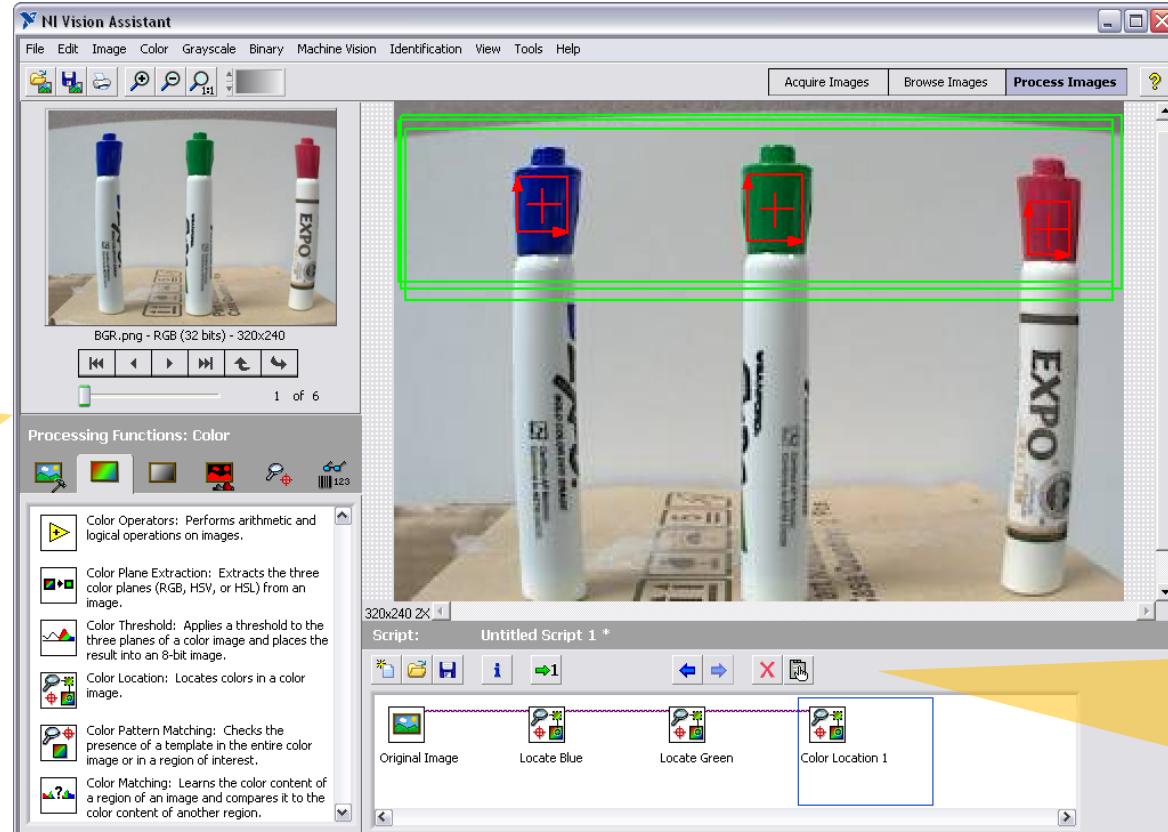


Dashboard with  
vision controls

# Vision Code

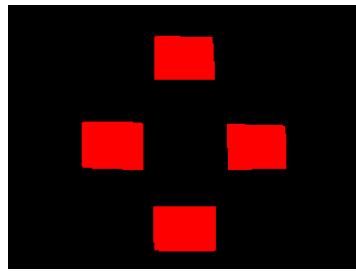
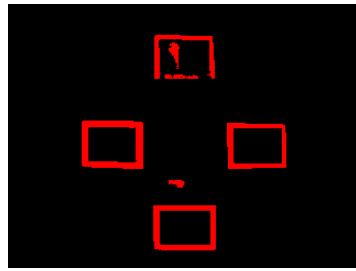
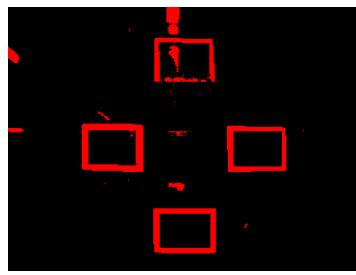


# NI Vision Assistant



Start»Programs»National Instruments»NI Vision Assistant

# Vision Processing Algorithm



- Develop a process to convert a camera image into useful data for your robot.

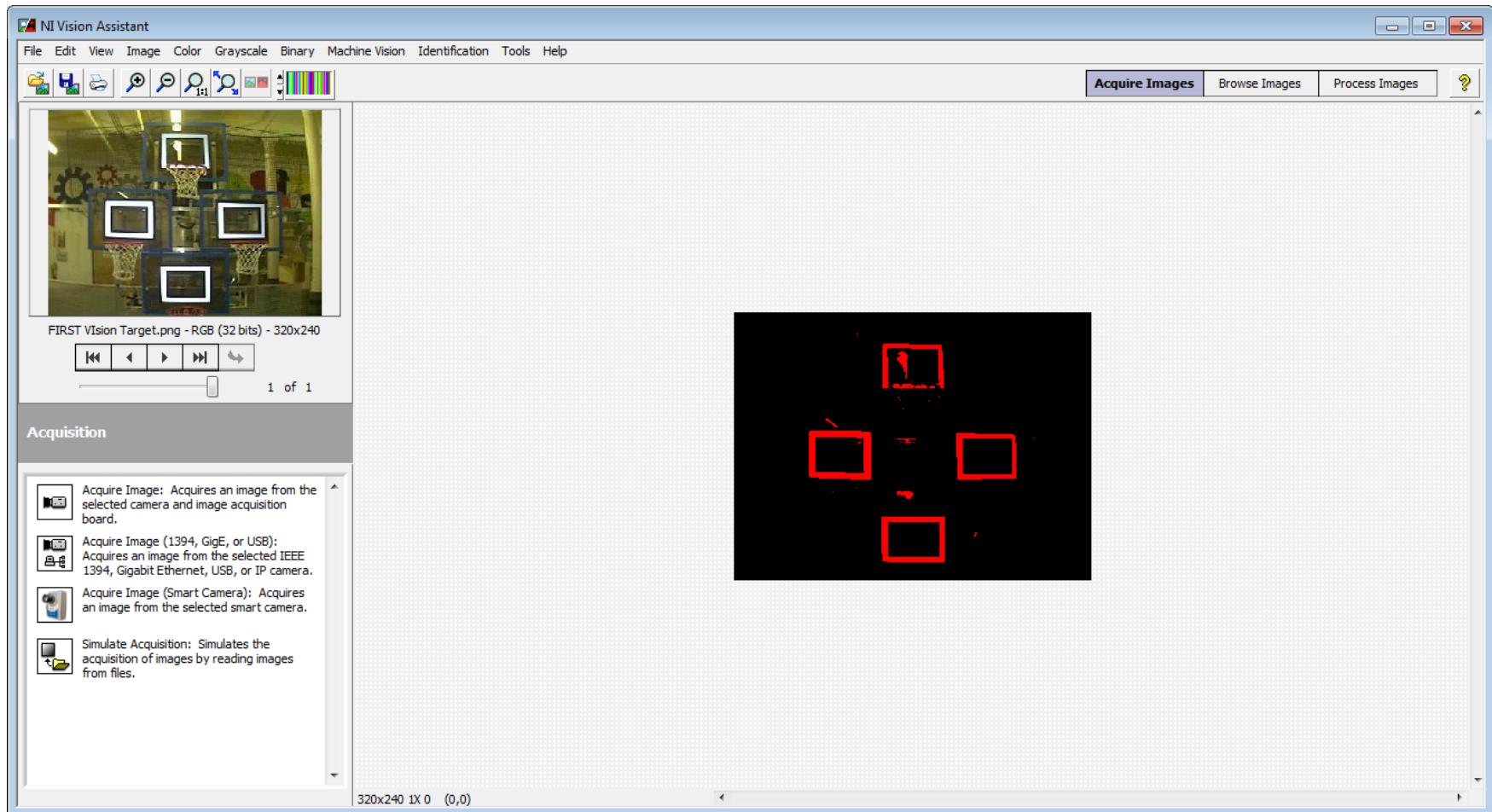
Segment/simplify  
Convert to binary image

Clean up  
Remove noise

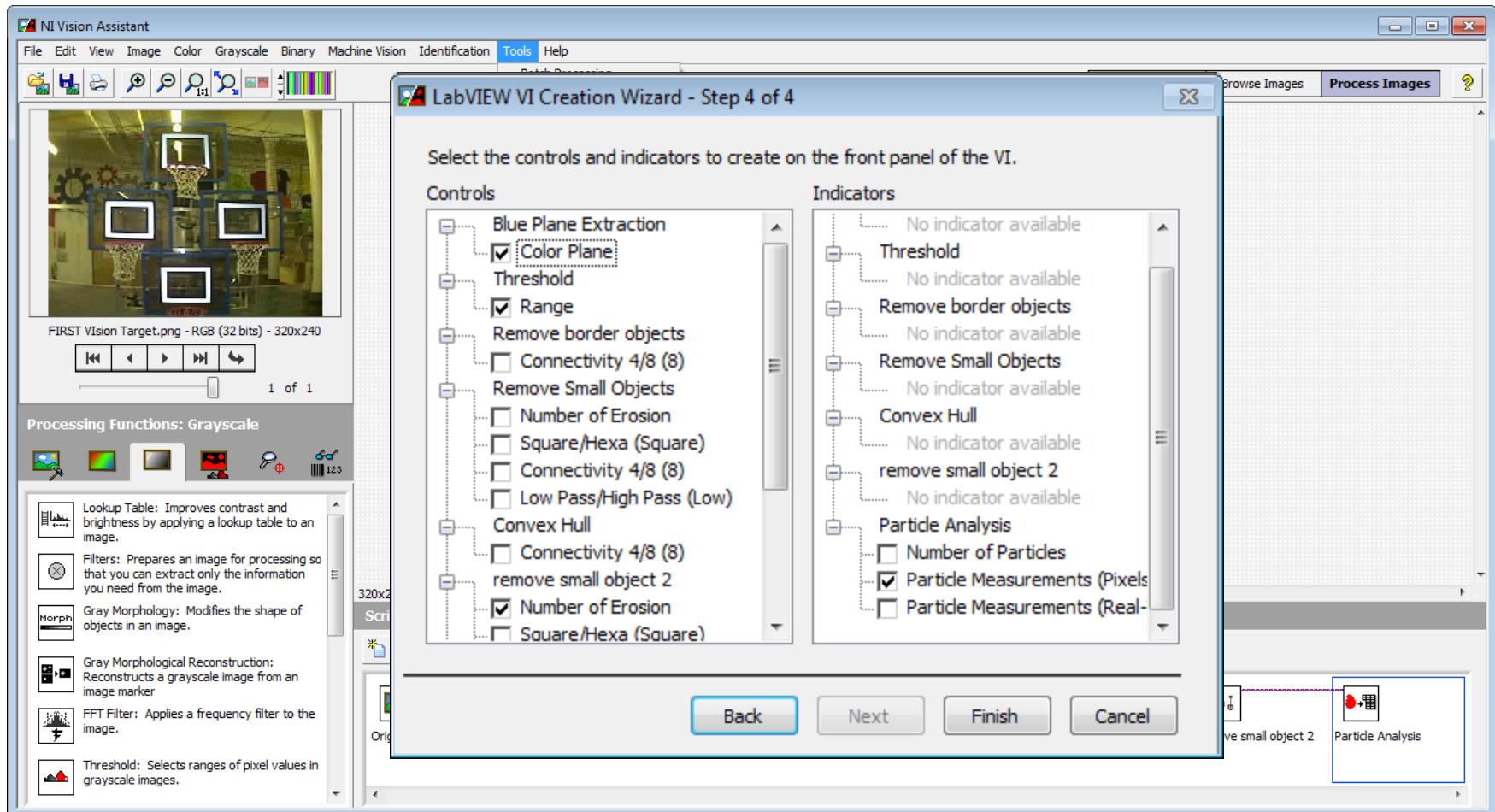
Extract

Output IE: X,Y coordinates of each object

# Vision Assistant Demonstration



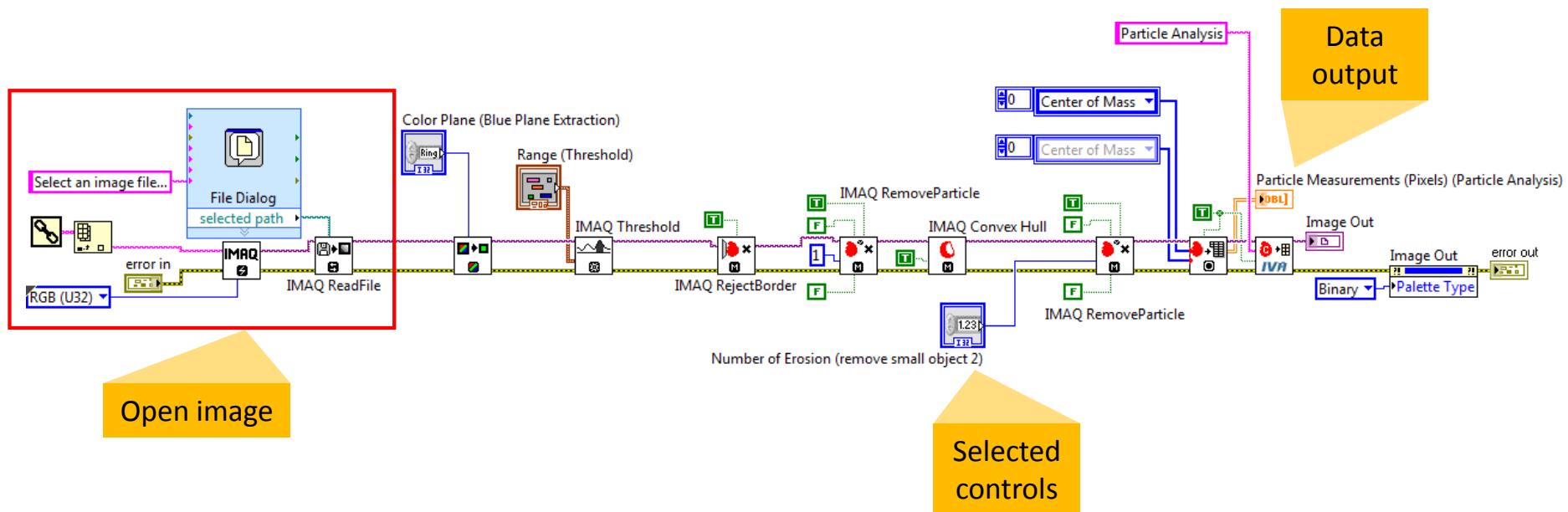
# Generate Vision Code



# Generated Code

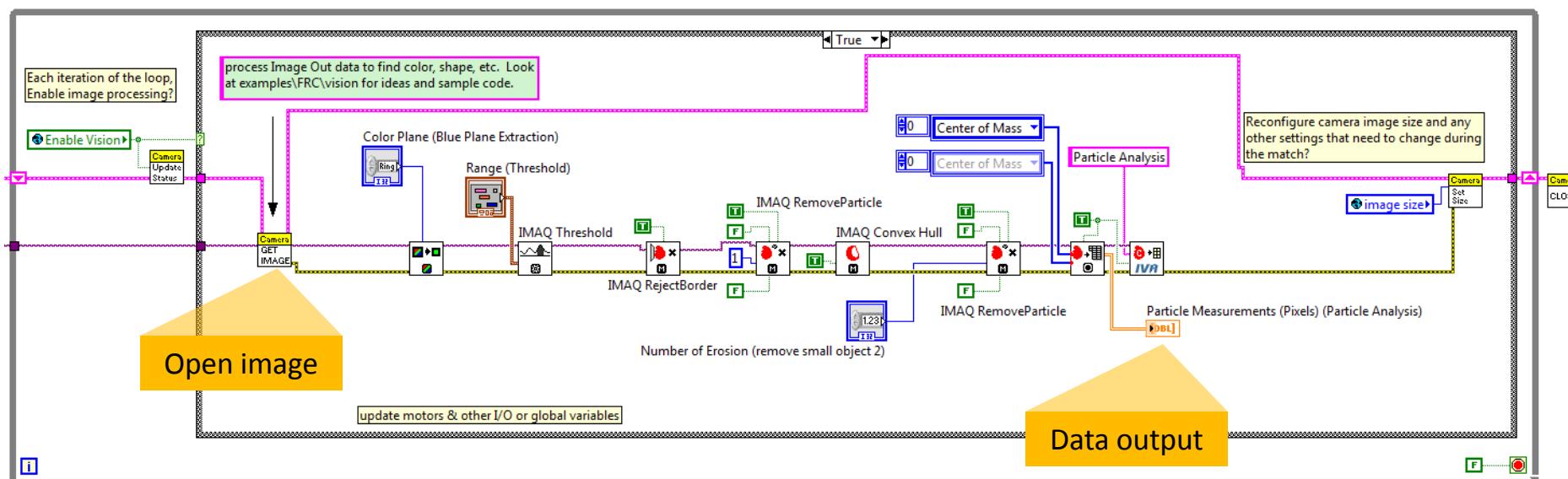
## Code Generated From Vision Assistant

- Delete the Open Image code and paste this into Vision Processing.vi in your team code.



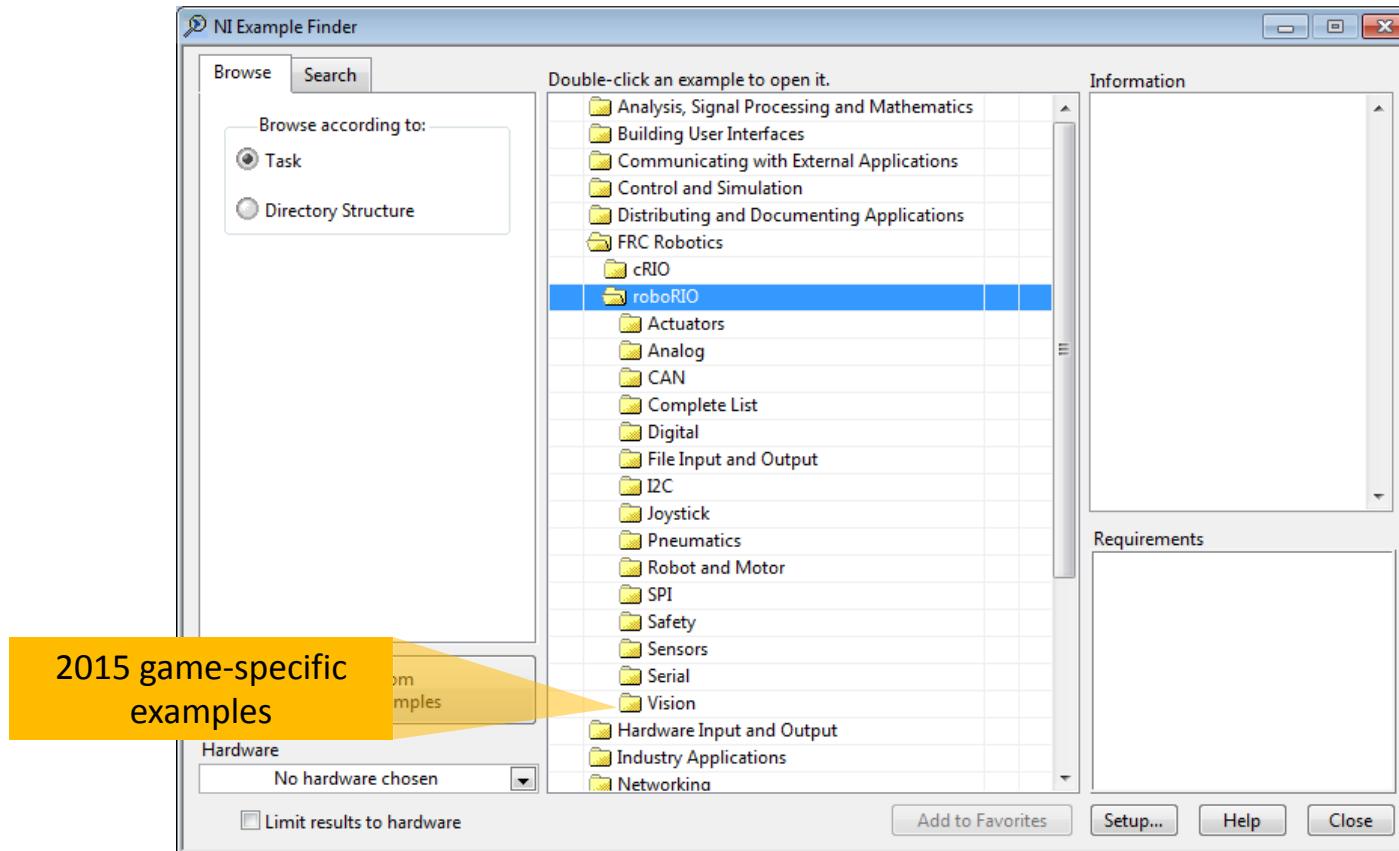
# Integrate Vision Code

- Paste generated code in Vision Processing.vi.
  - Use the Open Image function in Vision Processing.vi
  - Now you can do math and make decisions using the Data Output
  - Use Edit»Create SubVI to convert your algorithm to one function to keep your code clean



# LabVIEW Vision Examples

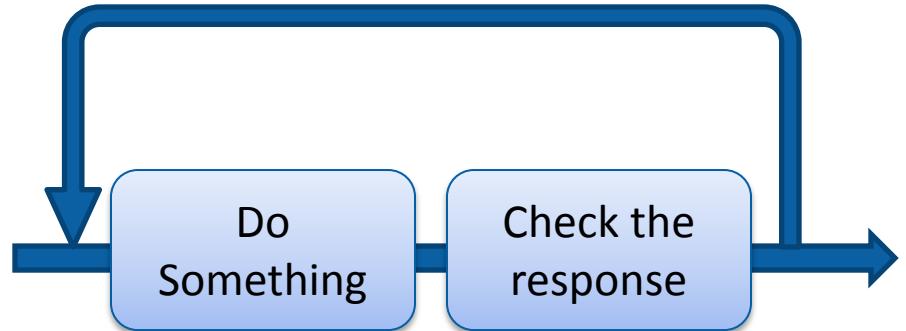
## Help>Find Examples



# Open Loop Vs Closed Loop



Open Loop  
Control



Closed Loop  
Control

# Open Loop Control

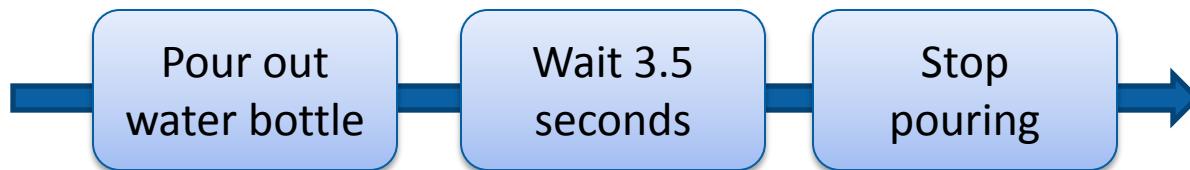
Imagine filling a glass of water without looking

## Potential Sources of Errors

- Bottle glugging causes water to come out at different rates
- There could already be some water in the glass
- If the bottle is full, it pours differently than when it is partially full



Unless everything works exactly as you expect, you'll end up with water on the floor or a half full glass (half empty?)



# Closed Loop Control

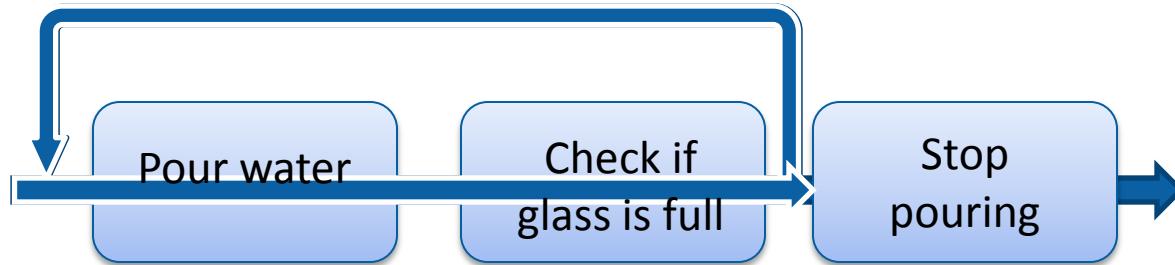
Imagine filling a glass of water while looking

1. Pour water out
2. Check if the glass is full
3. Keep pouring or stop

Potential sources of errors are mitigated

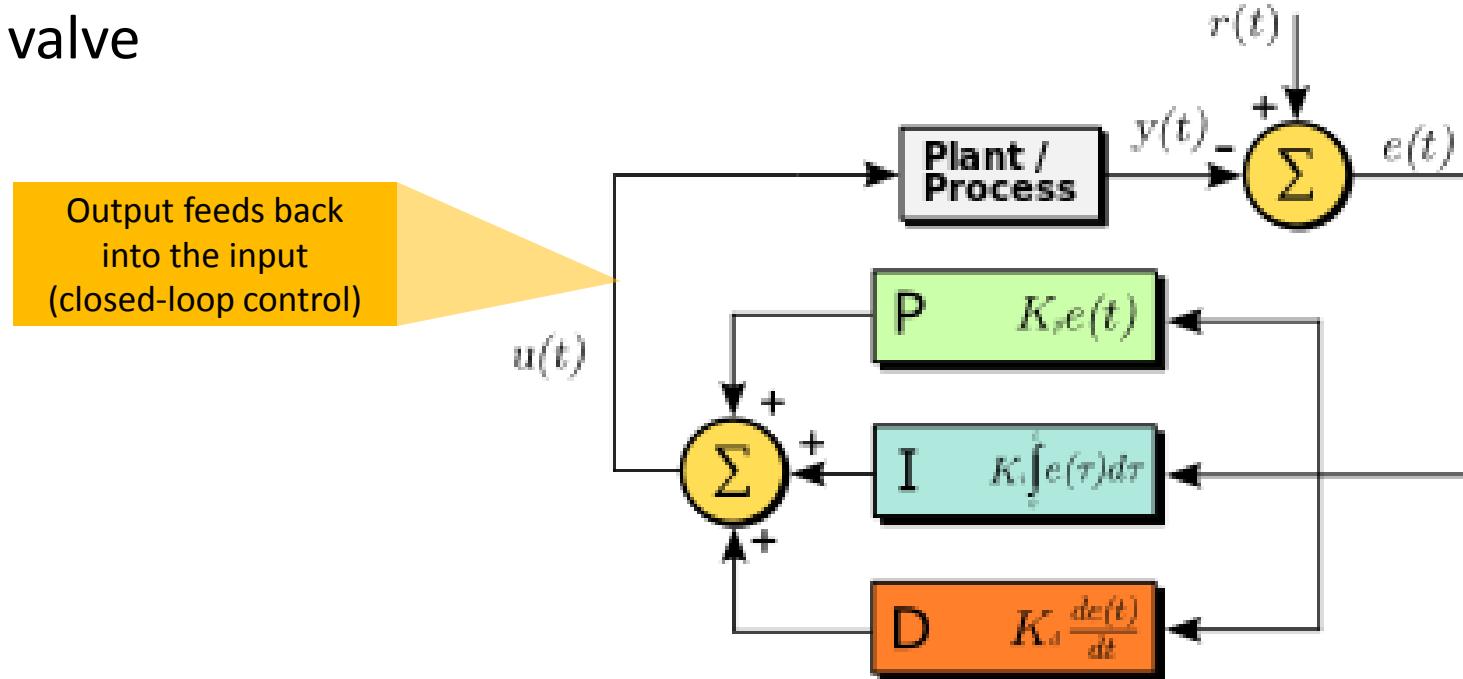


Makes a closed loop

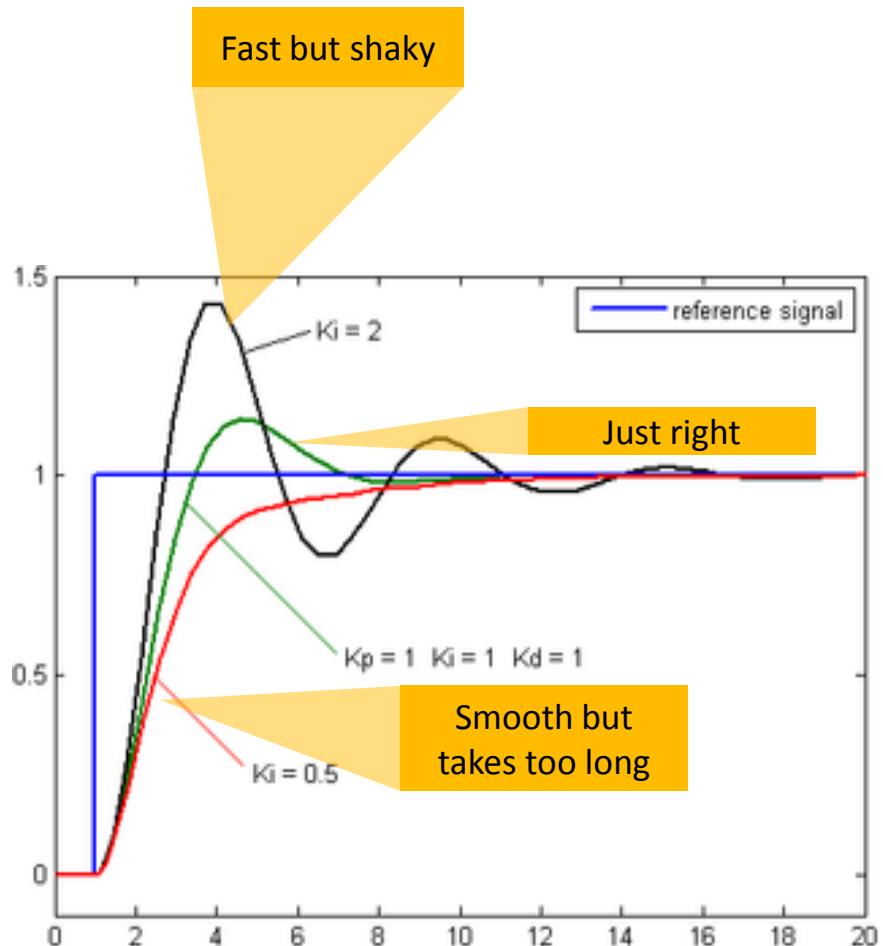


# PID

- PID is one methods of closed loop control
- Proportional integral derivative
- An algorithm used in a control feedback loop to regulate a process such as the motion of a motor or the flow through a valve

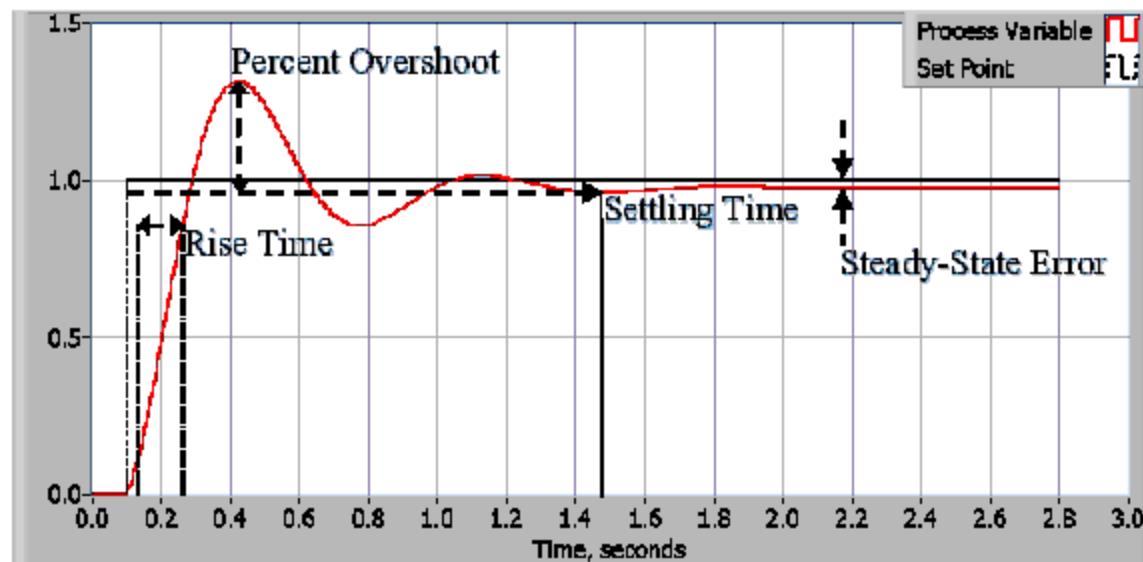


# PID in the Real World



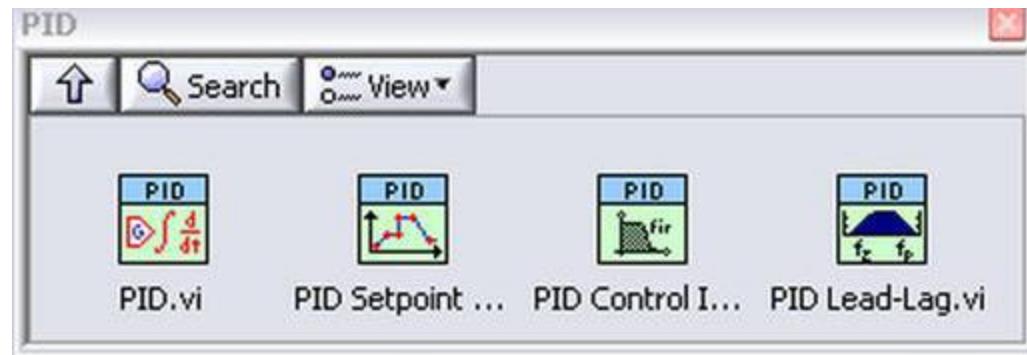
# Tuning a Controller

- Adjust the PID gains to the appropriate values for your specific system.
- Decrease rise time and steady-state error.

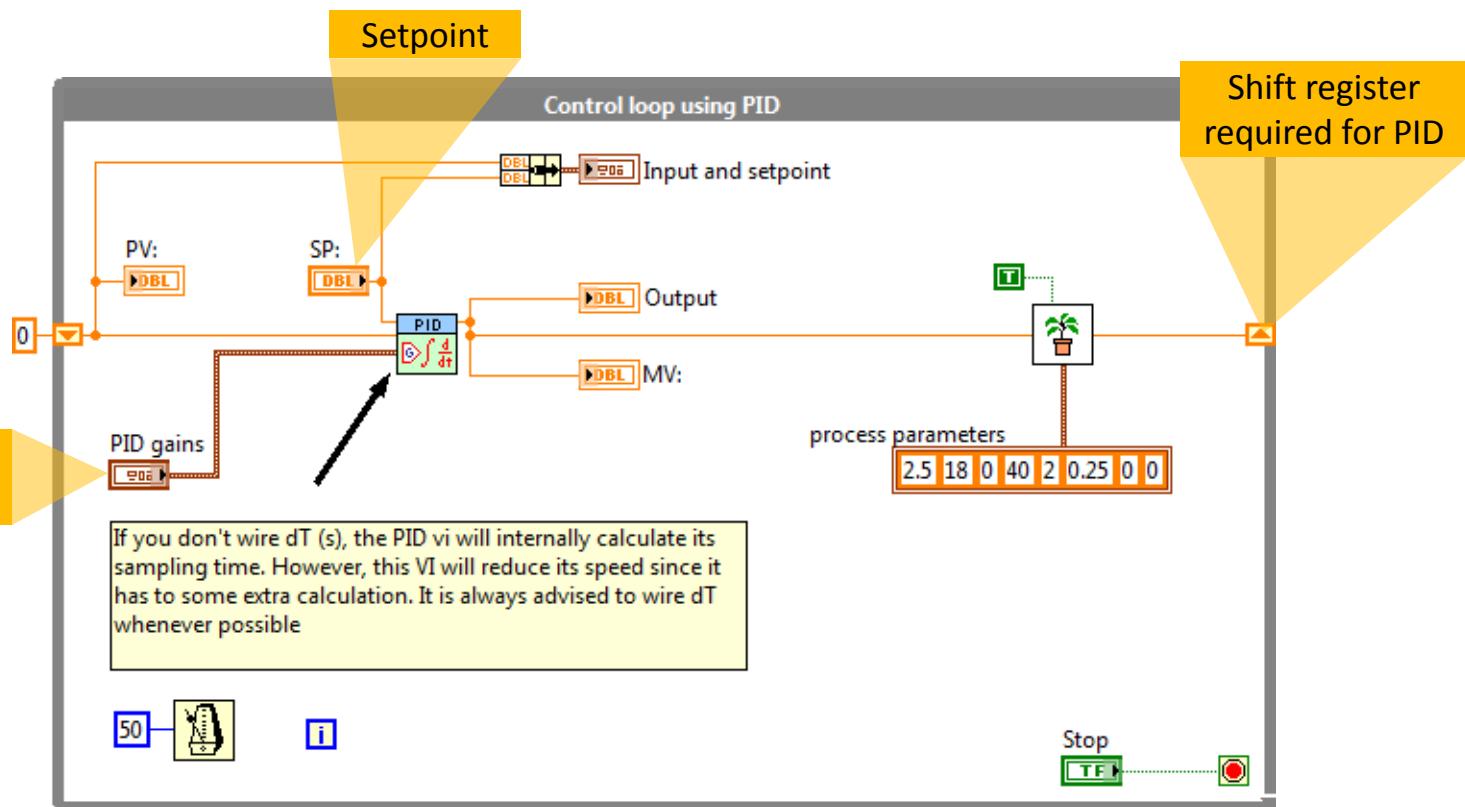


# Using PID in LabVIEW

- The LabVIEW FRC PID Palette consists of four VIs:
  - PID
  - PID Setpoint Profile
  - PID Control Input Filter
  - PID Lead-Lag



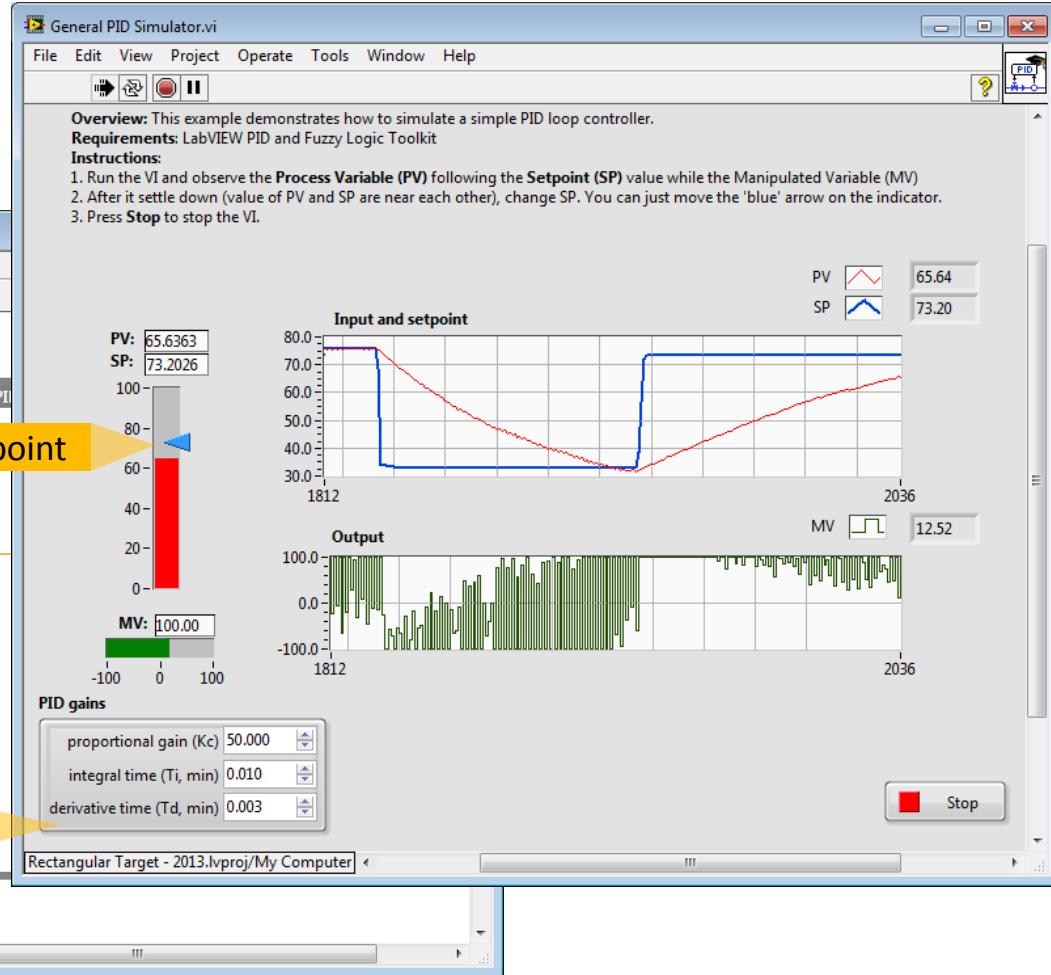
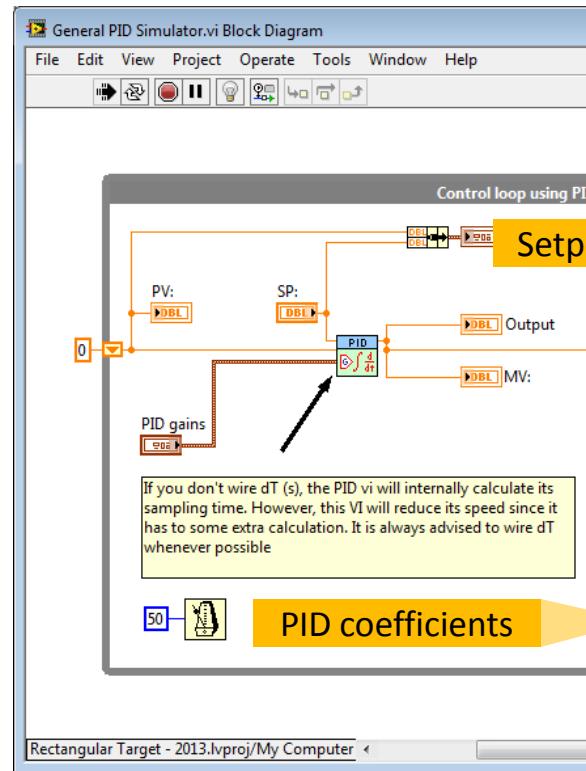
# PID in LabVIEW



NI Example Finder»General PID Simulator.vi

# PID Example

- Try the PID Simulator example.



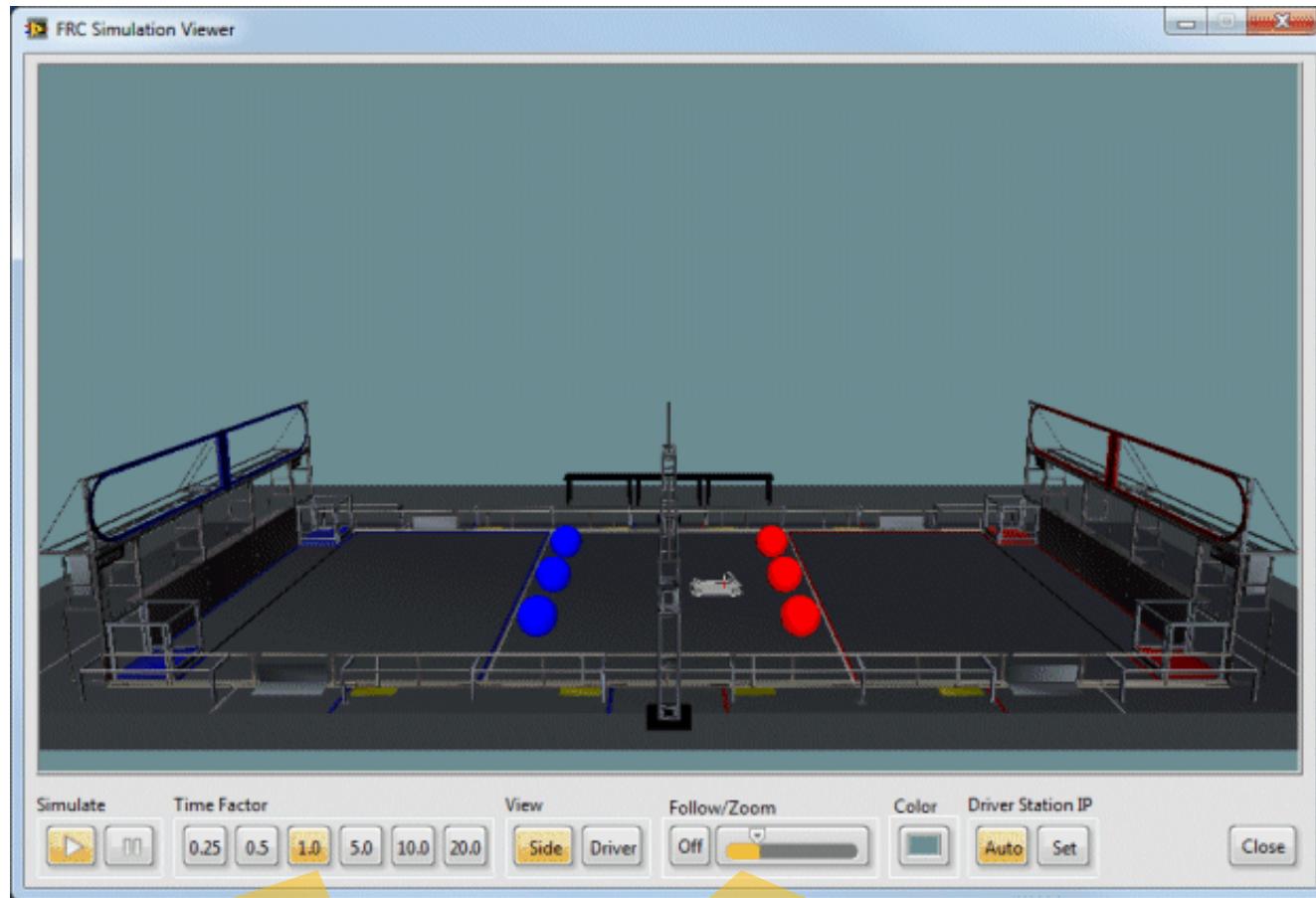
NI Example Finder»General PID Simulator.vi

# Closed Loop is not perfect

- Encoders won't detect if a wheel slips, they just measure how far the wheel rotates, Meacham wheels are even harder to measure as a distance.
- Encoders don't always work perfectly
- If you are waiting for a bumper or line sensor and it doesn't trigger, your robot may get stuck in its current mode
- Light levels can change in different lighting scenarios (workshop VS arena)
- Ultrasonic sensors can give bad readings

The key to good autonomous closed loop control is LOTS of testing in different scenarios.

# The Robotics Simulator

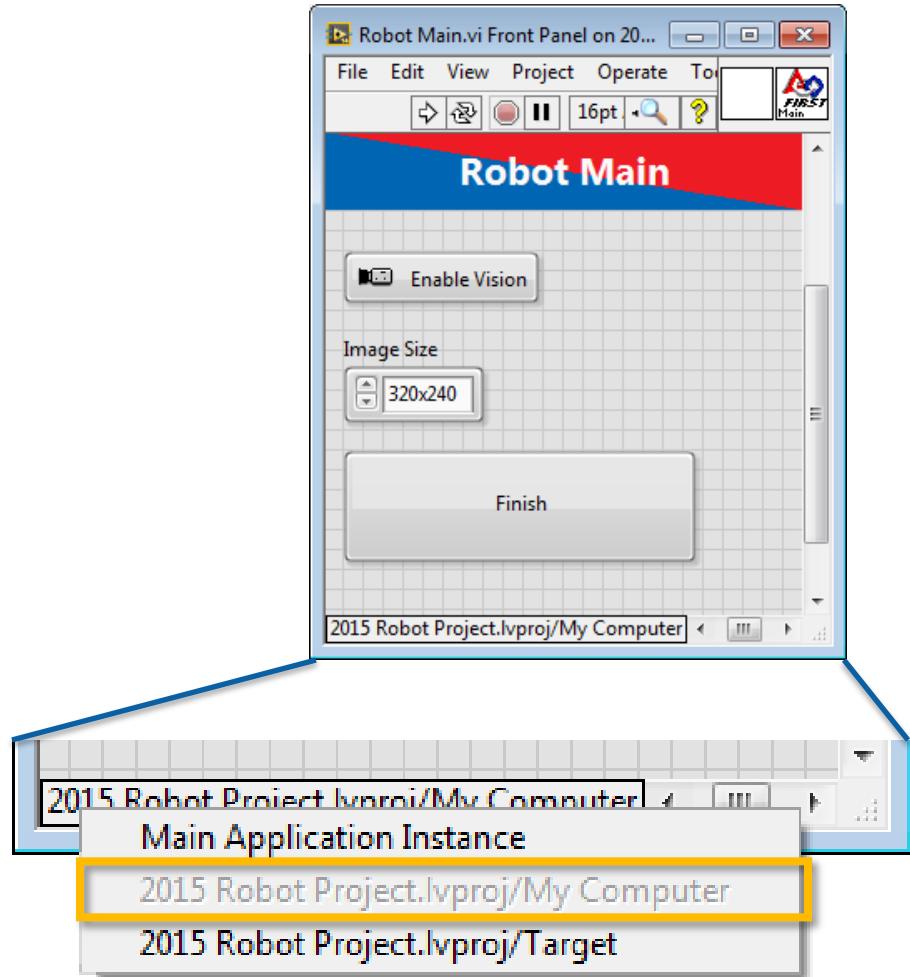


Speed up/slow  
down simulation

View settings

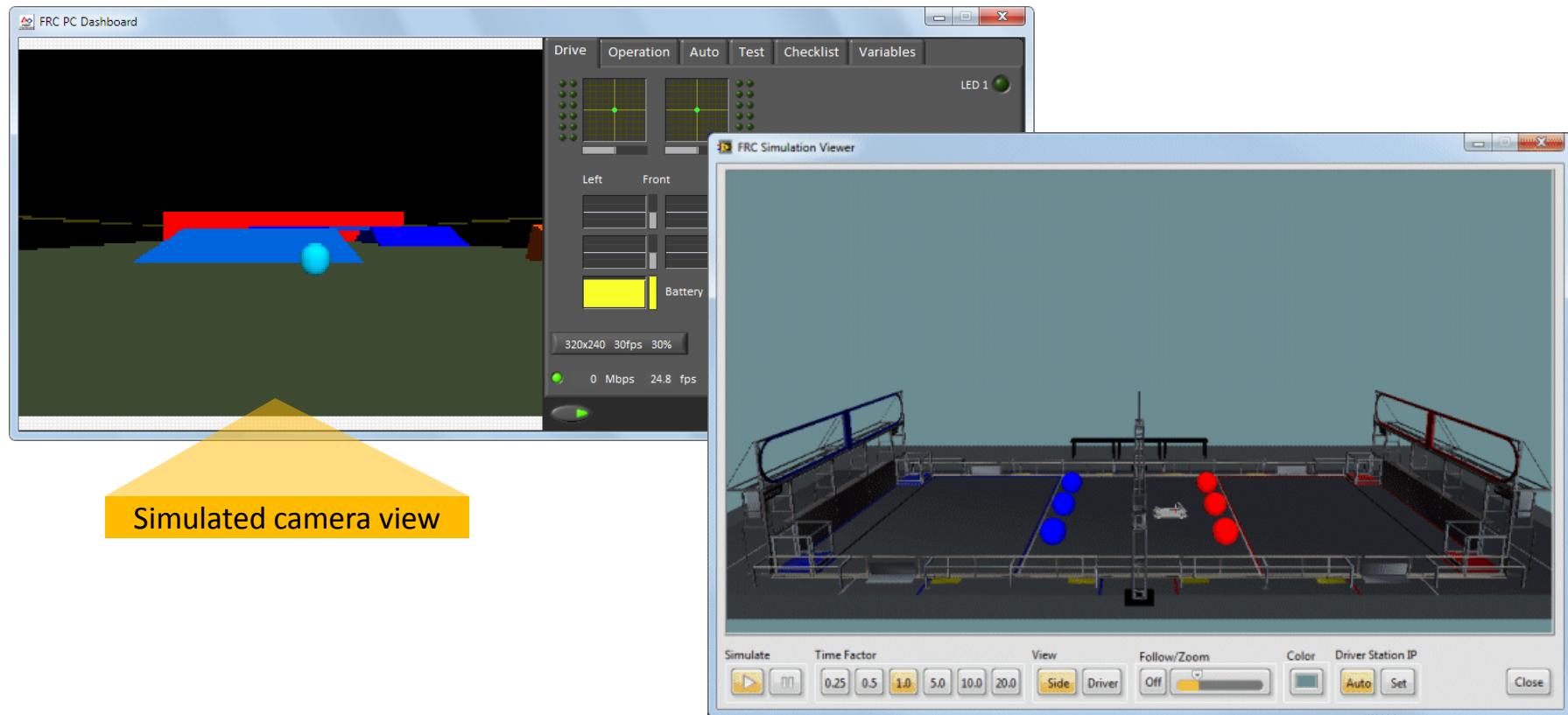
# Deploy Code to Simulator

- Start the Driver Station. You need it for real and simulated robots.
- Right-click in the lower left corner of Robot Main.vi and choose **2015 Robot Project.lvproj/My Computer**.
- Run Robot Main.vi and note that the FRC Simulation Viewer opens.
- Enable Teleop Mode to drive the simulated robot or enable Autonomous Mode for independent control.



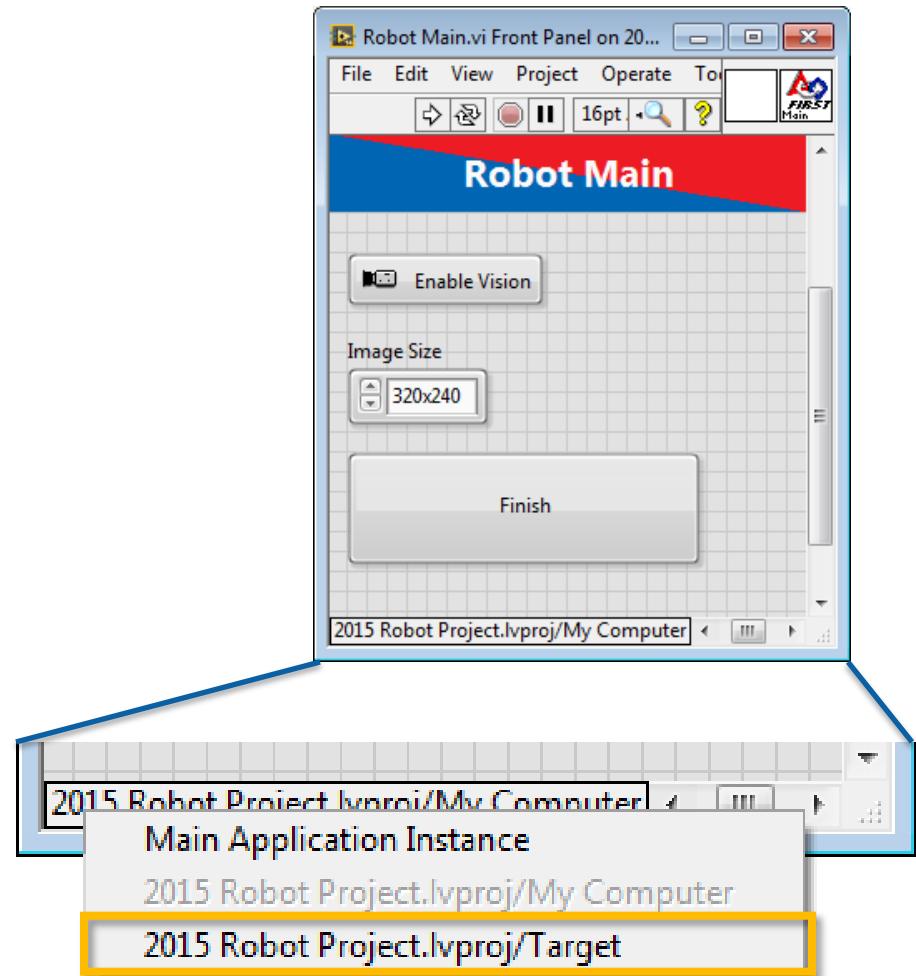
# Exercise: Deploy FRC roboRIO Robot Project

- Deploy the default robot project to the simulator.
  - The Driver Station must be running.



# Deploy to Real FRC Robot

- Right-click in the lower left corner of Robot Main.vi and choose **2015 Robot Project.lvproj/RT roboRIO Target**.



# Vision, PID, and Simulation Resources

- [Configure an AXIS Camera](#)
- [Image Processing Tutorial](#)
- [Using Vision Targets](#)
- [List of NI Vision Functions](#)
- [LabVIEW PID Tutorial](#)
- [PID Theory Explained](#)
- [Robot Simulator Tutorial](#)

All links available at  
[ni.com/frc](#) or  
click this button



- Forum Support (year round)
  - [ni.com/frc](#)
- Phone Support (during build season only)
  - (866) 511-6285 from 1:00 to 7:00 p.m. (Central Time)

# FRC LabVIEW Quick Start Guide

Setup

NI LabVIEW  
Basics

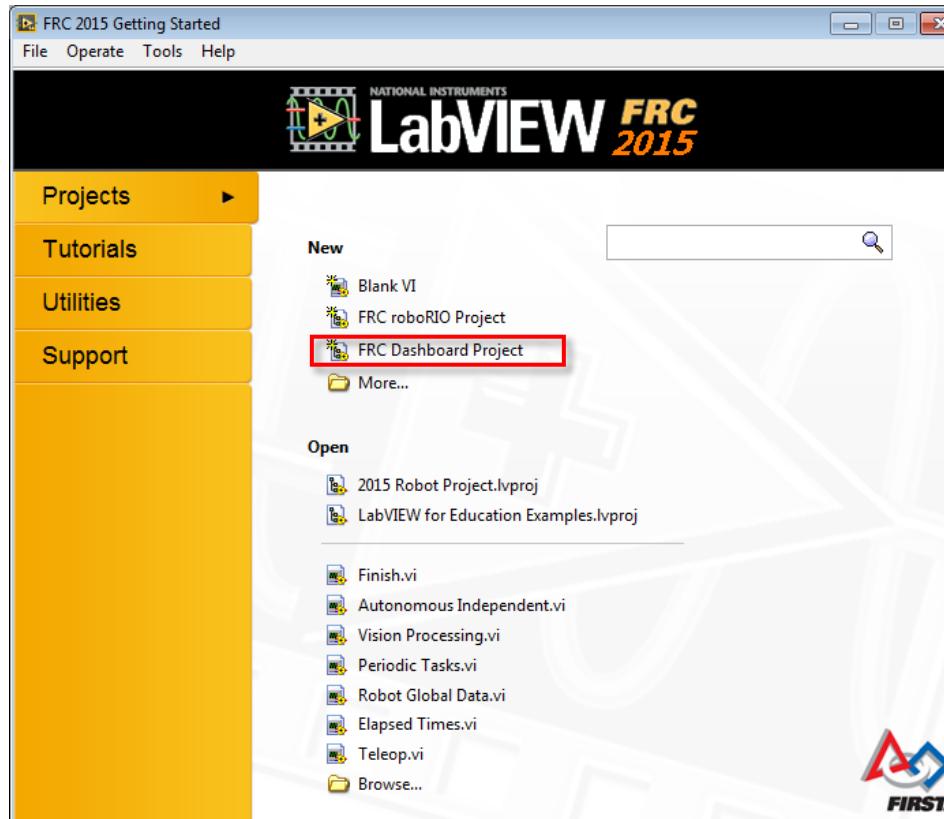
Vision,  
PID, and  
Simulation

Advanced  
Programming

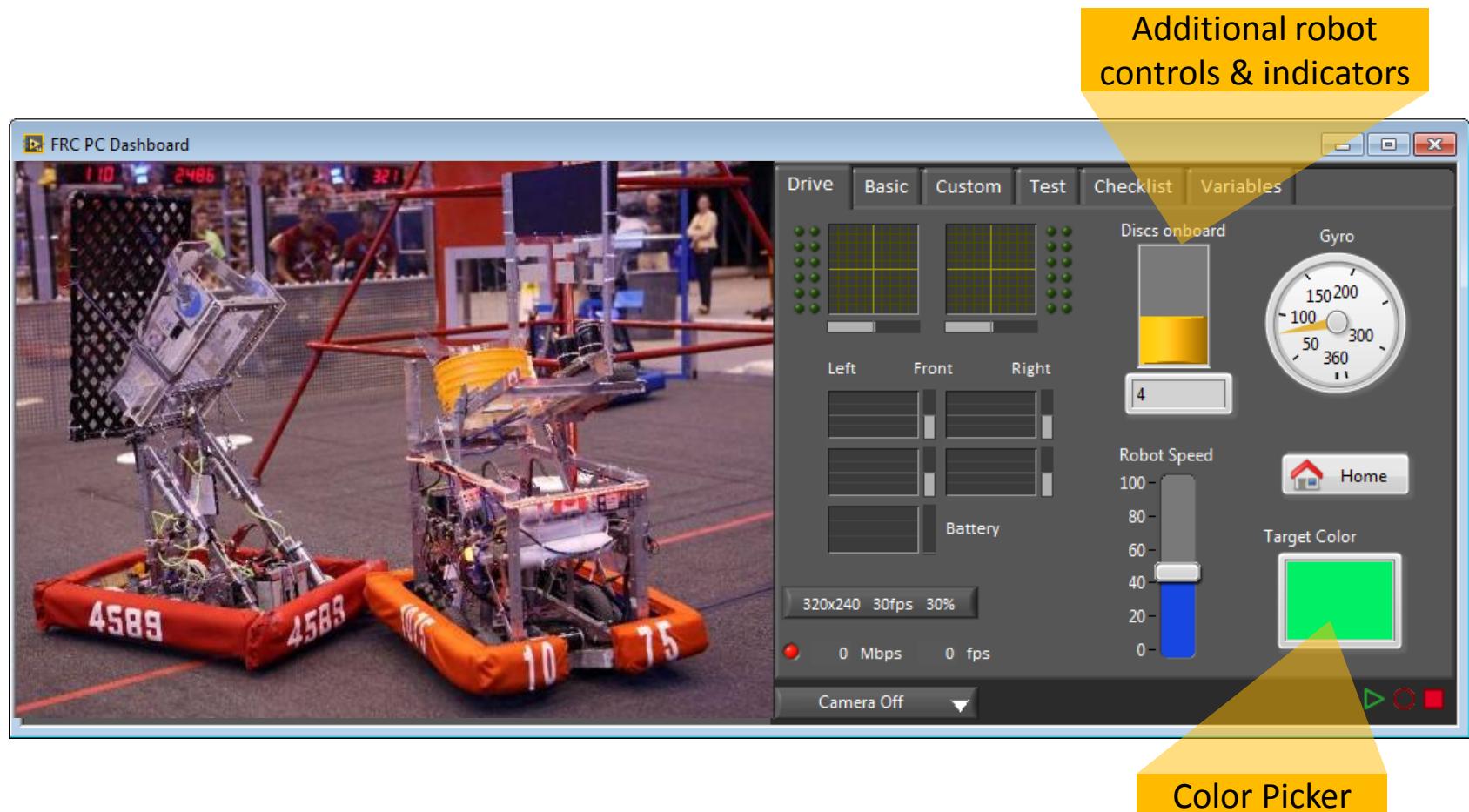
- Dashboard Customization
- Dashboard Record and Playback
- Event Error Logs
- Performance
- Autonomous Strategies

# Dashboard Customization

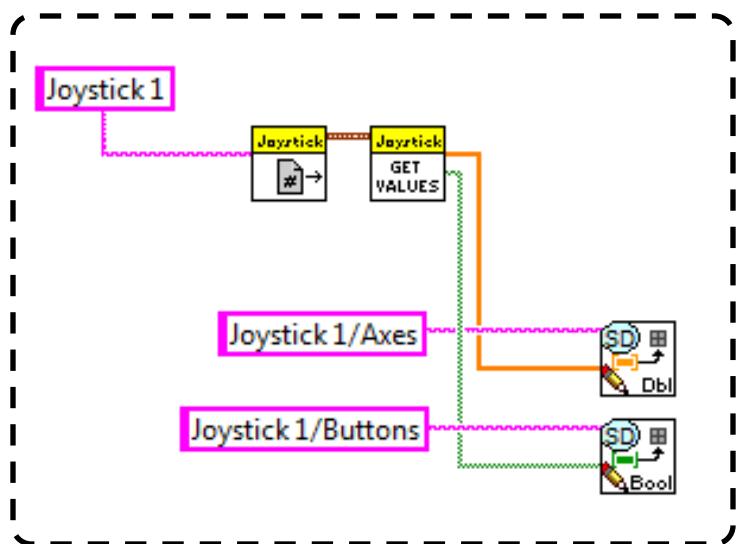
- Start by opening an FRC Dashboard Project.



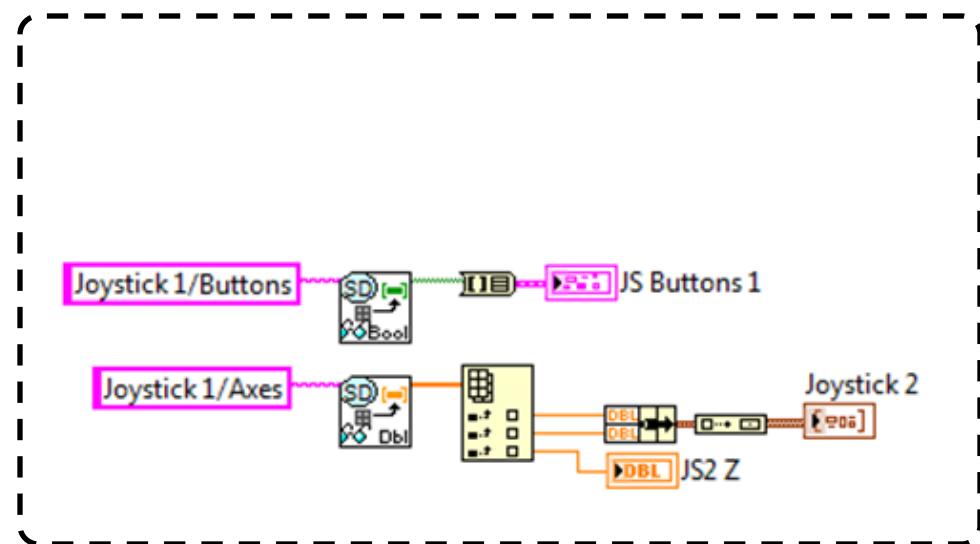
# Dashboard Customization



# Connecting Custom Variables to the Dashboard

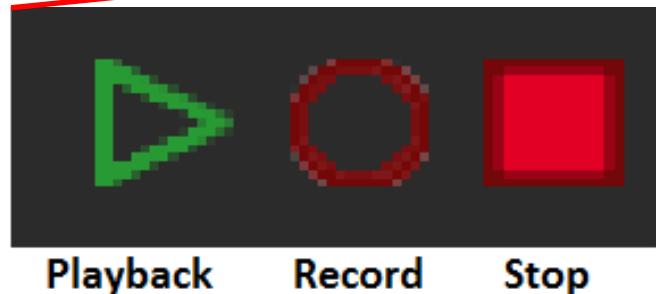
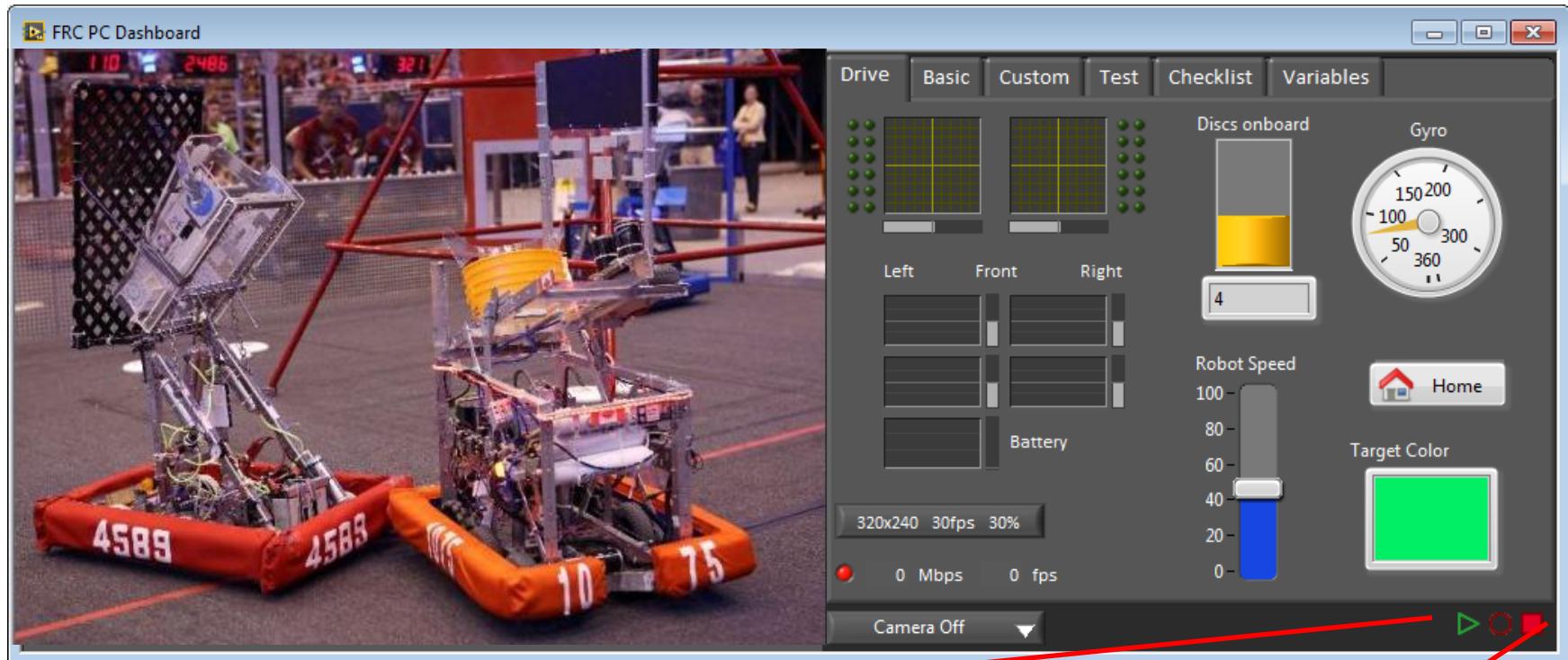


Teleop.VI

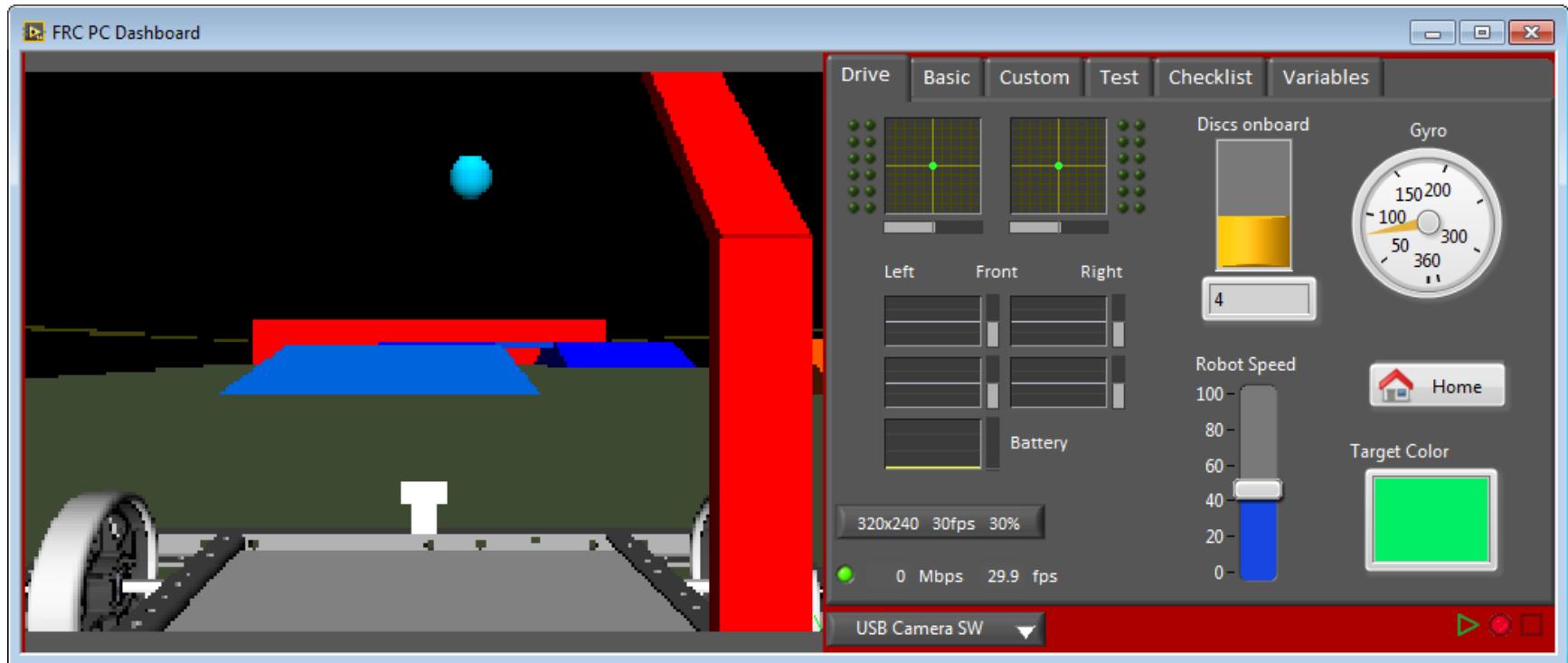


PC Dashboard.VI

# Dashboard Record and Playback

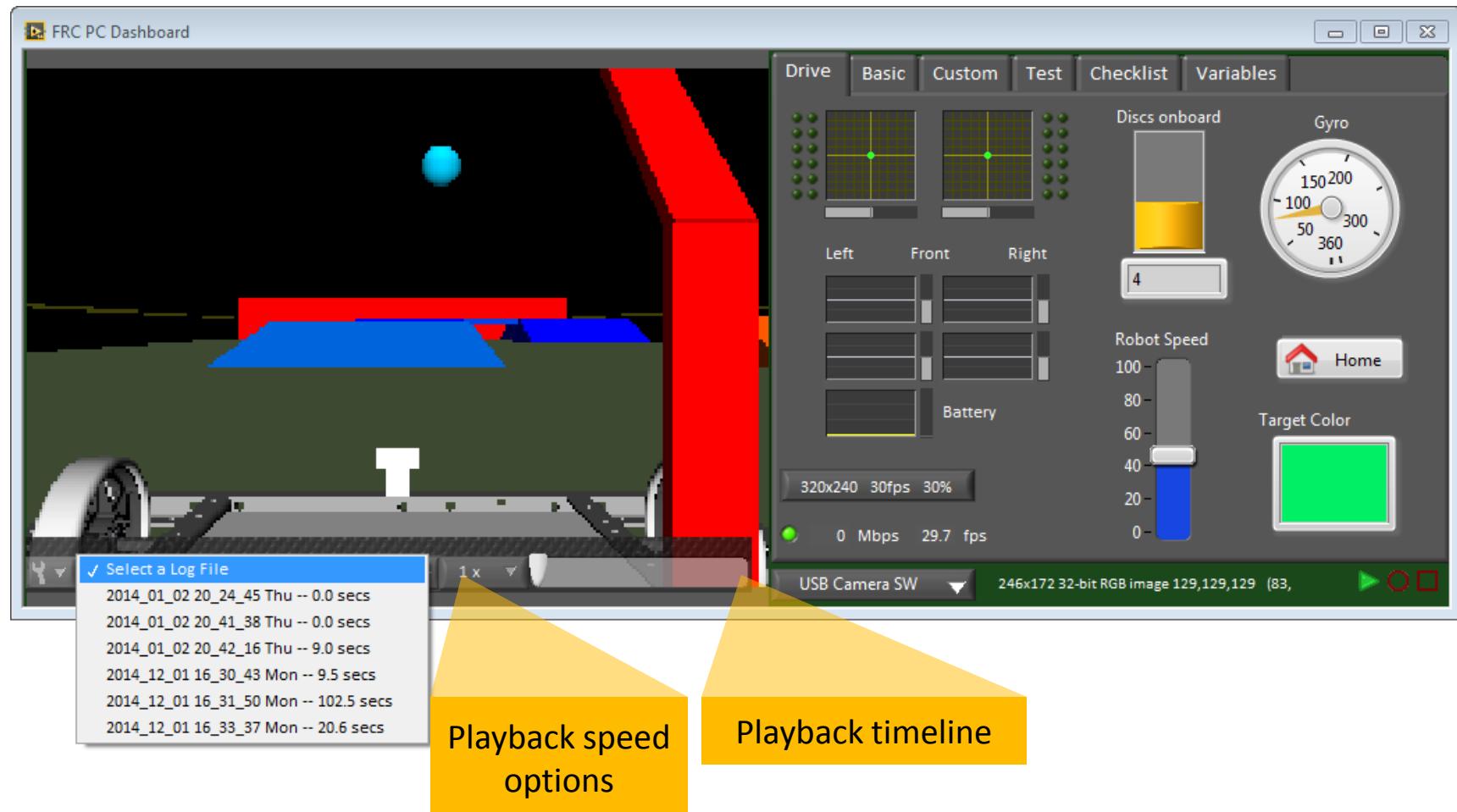


# Record



Log Location: C:\Users\Public\Documents\FRC\Log Files\Dashboard

# Playback



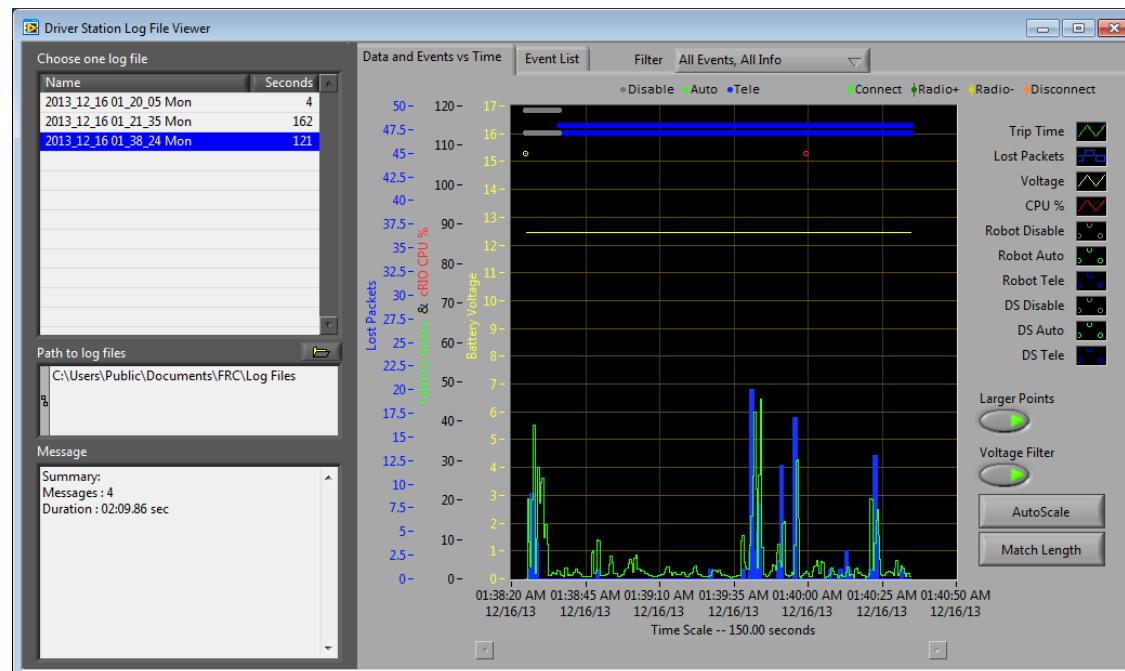
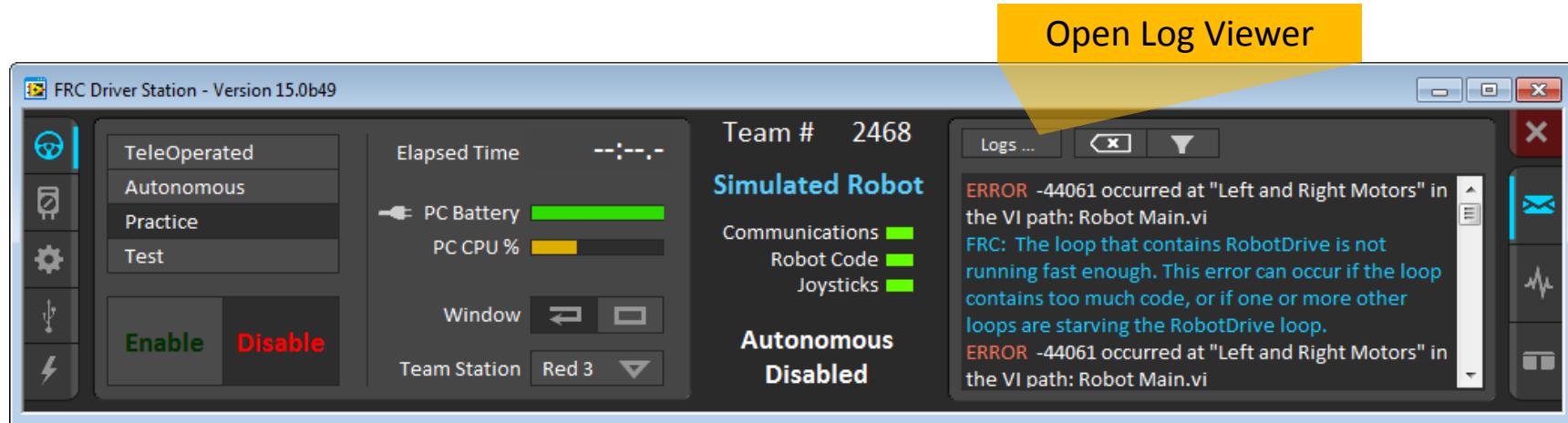
# Troubleshooting: Driver Station Charts

- Network Issues
  - Lost data packets
  - Data packet trip time
- Programming Issues
  - CPU usage
  - Free RAM on roboRIO

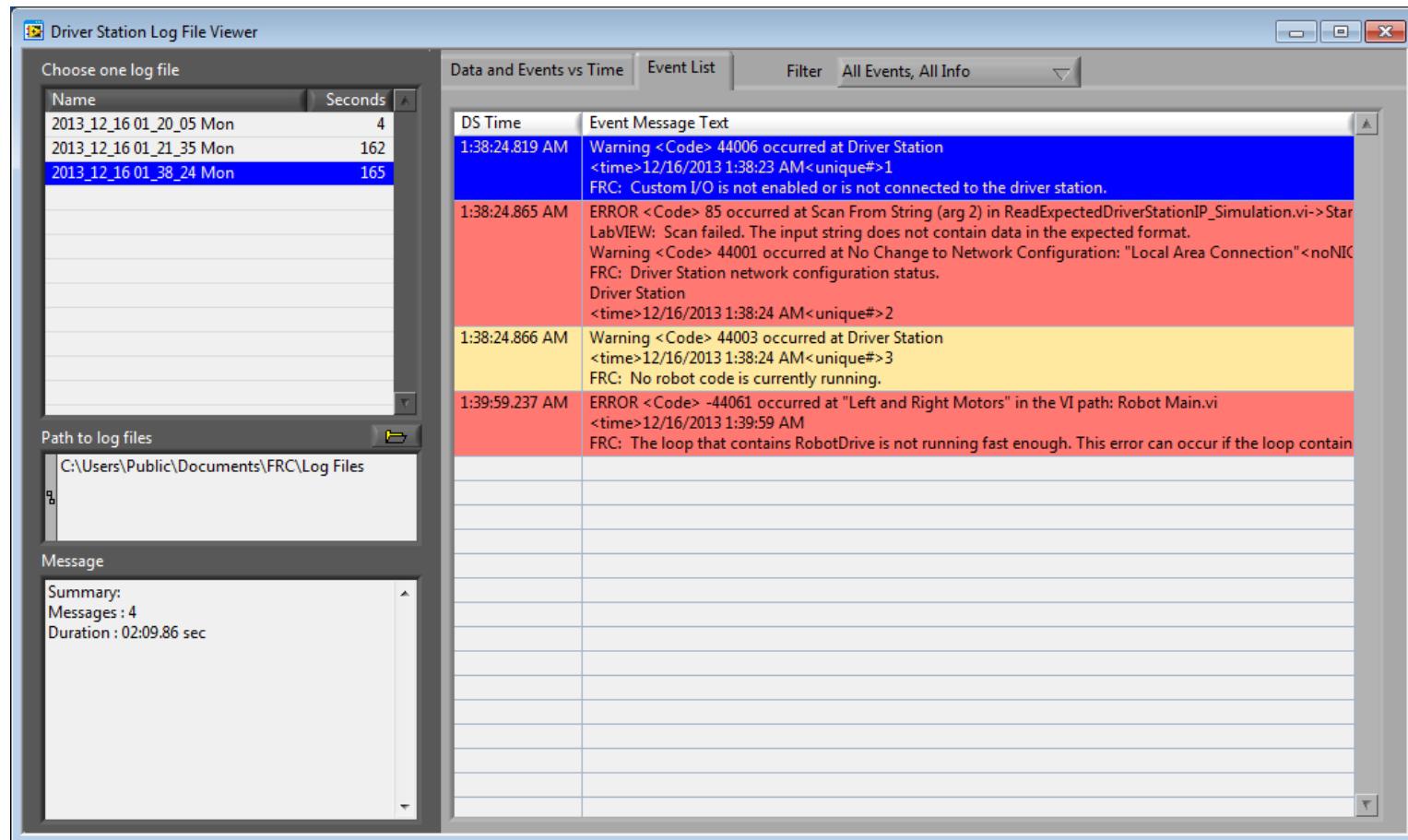
- Hardware Issue
  - Robot battery voltage



# Troubleshooting: Event Error Logs



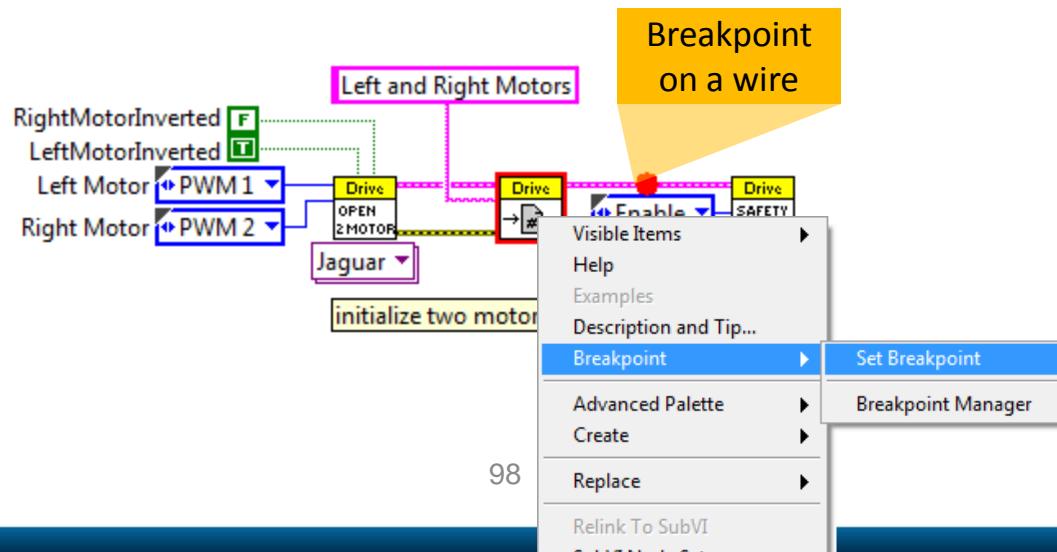
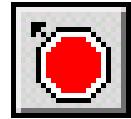
# Troubleshooting: Event Error Logs



Log Location: C:\Users\Public\Documents\FRC\Log Files

# Debugging: Breakpoints

- When you reach a breakpoint during execution, the VI pauses and the **Pause** button appears red.
  - (right-click Breakpoint»Set Breakpoint)
- You can take the following actions at a breakpoint:
  - Single-step through execution using the single-stepping buttons.
  - Probe wires to check intermediate values.
  - Change the values of the front panel controls.
  - Click the **Pause** button to continue running to the next breakpoint or until the VI finishes running.



# Performance

Profile Performance and Memory - 2014 Robot Project.lvproj

Timing statistics     Profile memory usage  
 Timing details     Memory usage

Time unit: milliseconds    Size unit: kilobytes    Select Application Instances...

Application Instances:  
 My Computer     RT CompactRIO

	Avg Bytes	Min Bytes	Max Bytes	Avg Blocks	Min Blocks	Max Blocks	Project Lst
NI_Robotics_Simulator.lvlib:ODESimulator_Engine.lvclass:Rendering.vi	3.69k	3.69k	3.69k	1	1	1	NI_Roboti
NI_Robotics_Simulator.lvlib:ODESimulator_Engine.lvclass:TakeOneSimStep.vi	5.34k	5.34k	5.34k	1	1	1	NI_Roboti
NI_Robotics_Simulator.lvlib:3D_Physics_Base.lvclass:Synchronize 3D.vi	5.24k	5.24k	5.24k	12	12	12	NI_Roboti
NI_Robotics_Simulator.lvlib:ODESimulator_Engine.lvclass:Read Robots.vi	3.00k	3.00k	3.00k	1	1	1	NI_Roboti
NI_Robotics_Simulator.lvlib:Load Engine From XML.vi	216.07k	216.07k	216.07k	36	36	36	NI_Roboti
NI_Robotics_Simulator.lvlib:Start Simulator Service.vi	10.12k	10.12k	10.12k	46	46	46	NI_Roboti
NI_Robotics_Simulator.lvlib:Get Robot By ID.vi	7.05k	7.05k	7.05k	17	17	17	NI_Roboti
ReadExpectedDriverStationIP_Simulation.vi	6.99k	6.99k	6.99k	13	13	13	
Start_Simulation_Engine.vi	13.38k	13.38k	13.38k	10	10	10	
NI_Robotics_Simulator.lvlib:ODESimulator_Distance.lvclass:Step.vi	14.34k	14.34k	14.34k	17	17	17	NI_Roboti
FPGA_SystemStart Async Agent.vi	4.82k	4.82k	4.82k	10	10	10	
NI_Robotics_Simulator.lvlib:Get AABB.vi	7.59k	7.46k	7.72k	14	14	14	NI_Roboti
NI_Robotics_Simulator.lvlib:Load CAD Model.vi	7.62k	7.56k	7.66k	16	16	21	NI_Roboti
NI_FileType.lvlib:Get File Type.vi	6.86k	6.86k	6.86k	14	14	14	NI_FileTyp
NI_Robotics_Simulator.lvlib:Find Object (with invisible).vi	4.39k	4.39k	4.39k	3	3	3	NI_Roboti
FindMatchingCloseTag.vi	216.20k	216.20k	216.20k	14	14	14	
IMAQ Create	3.24k	3.24k	3.24k	4	4	4	
Read From XML File(string).vi	633.55k	633.55k	633.55k	33	33	33	
FindEmptyElement.vi	4.14k	4.14k	4.14k	4	4	4	
NI_Robotics_Simulator.lvlib:3D_Physics_Plane.lvclass>Create Physics Frame.vi	184.31k	184.31k	184.31k	9	9	9	NI_Roboti
NI_Robotics_Simulator.lvlib:Open Reference (Cylinder).vi	4.16k	4.16k	4.16k	2	2	2	NI_Roboti
NI_Robotics_Simulator.lvlib:3D_Physics_Base.lvclass:Draw 3D Object.vi	19.45k	19.45k	19.45k	42	42	42	NI_Roboti
NI_Robotics_Simulator.lvlib:Open Reference (Mass).vi	3.09k	3.09k	3.09k	2	2	2	NI_Roboti
NI_Robotics_Simulator.lvlib:3D_Physics_Base.lvclass:Read Physics Refnum.vi	3.69k	3.69k	3.69k	2	2	2	NI_Roboti
FindCloseTagByName.vi	213.30k	213.30k	213.30k	6	6	6	
NI_Robotics_Simulator.lvlib:3D Names.vi	3.05k	3.05k	3.06k	22	22	22	NI_Roboti
NI_Robotics_Simulator.lvlib:ODESimulator_Engine.lvclass:RenderContactPoints.vi	12.80k	12.80k	12.80k	7	7	7	NI_Roboti
NI_Matrix.lvlib:Matrix To Array - RM.vi	2.00k	2.00k	2.00k	0	0	0	NI_Matrix.
NI_Robotics_Simulator.lvlib:ODESimulator_Race.Robot Interface:Get Rank.vi	7.50k	7.50k	7.50k	102	102	102	NI_Roboti

Stop    Snapshot    Save    Close    Help

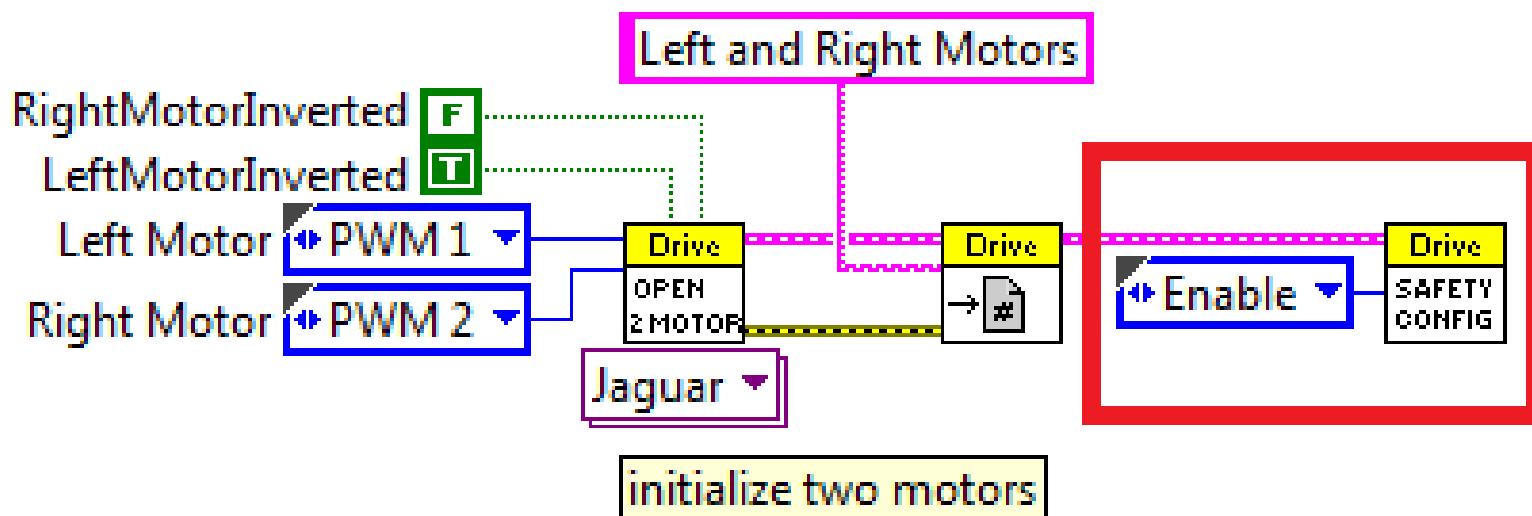
Tools»Profile»Performance and Memory

# Watchdog

- System Watchdog
  - Monitors communication from Driver Station
  - Turns off outputs if there is no robot communication in 100 ms
  - Causes
    1. No communication—radio issues?
    2. Slow communication—CPU overload?
    3. Driver Station running too slowly—dashboard hogging CPU

# Safety Config Timeouts

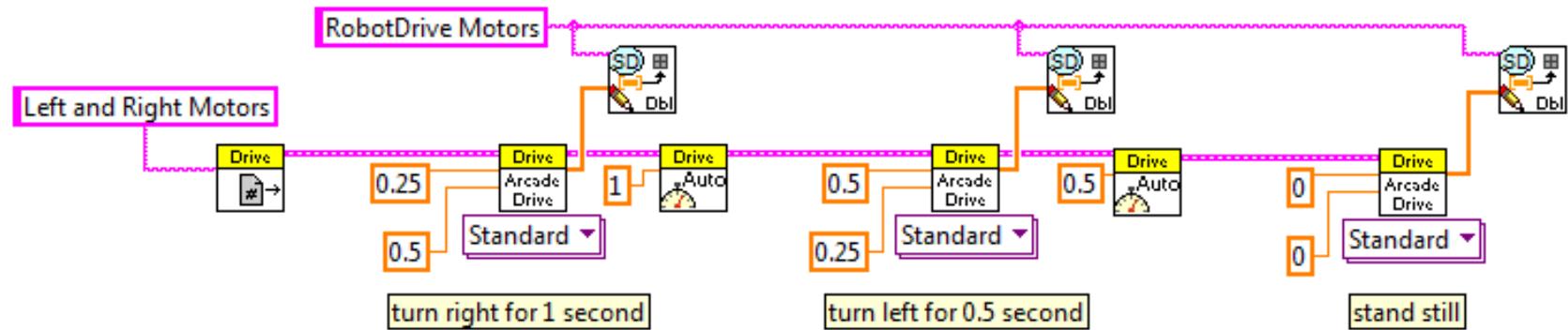
- You can turn on/off safety timeouts
- Available for RobotDrive, MotorControl, PWM, Relay, and Solenoid



# Autonomous Strategies

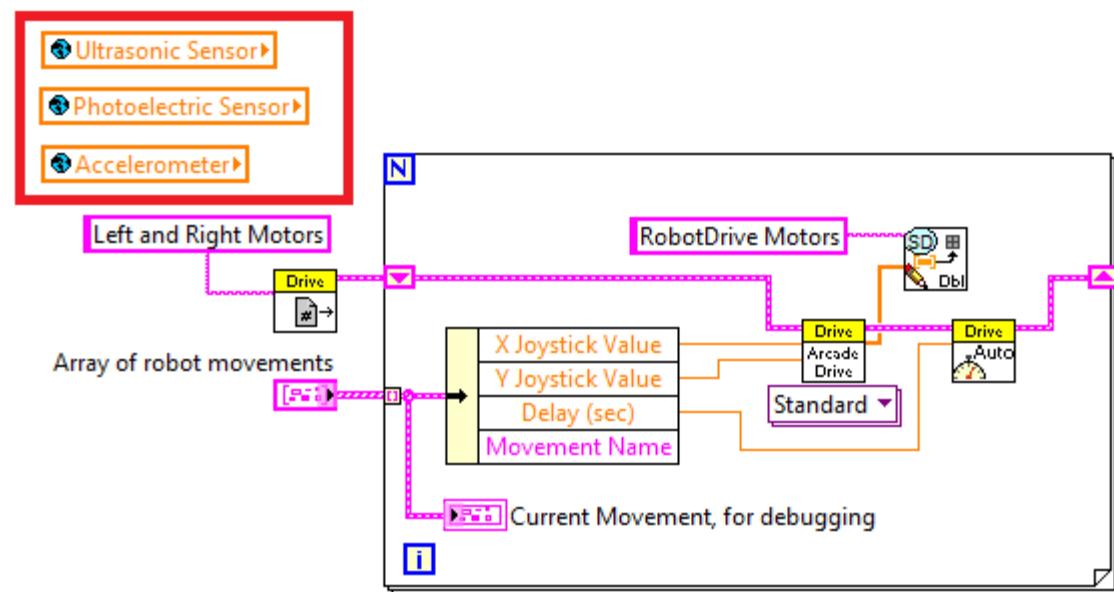
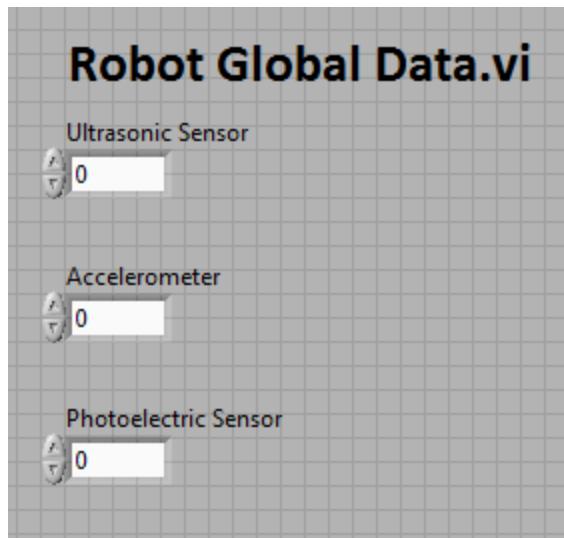
- Easiest Autonomous—Tell the robot to do the same thing every time!

This is some simple sample code to drive your robot curving one direction and then the other.



# Autonomous Strategies

- Closed-loop feedback



This sample code describes robot movements in an array. You can extend it to control other motors on your robot. You can also use file I/O to move the data to a text file.

# Advanced Programming Resources

- Full FRC LabVIEW Training (Beginner and Advanced—4.5 Hours Each)
- Programming for Performance
- Sensor Fusion Tutorial
- Autonomous Timed Movement

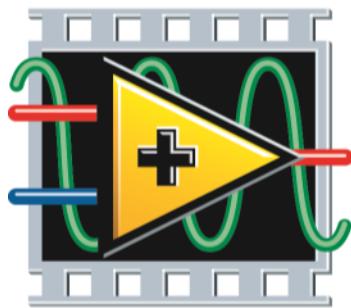
All links available at  
[ni.com/frc](http://ni.com/frc) or  
click this button



- Forum Support (year round)
  - [ni.com/frc](http://ni.com/frc)
- Phone Support (during build season only)
  - (866) 511-6285 from 1:00 to 7:00 p.m. (Central Time)

# LabVIEW Hacker.com

Driver Station Ideas, and Offseason Projects



NATIONAL INSTRUMENTS  
LabVIEW™



# NI.com/FRC

Download LabVIEW FRC and Update Suite

Download FRC 2015 Software



Additional training



Learn more about roboRIO & MXP



Review this PowerPoint and a list of all links

Get help from NI engineers and other teams

