

FRC Team 1912 Combustion 2013 Controls System Beta Testing Presentation





About Team 1912



- Northshore High School
- Slidell, Louisiana
- 2013 will be our 8th year
- Three time Regional Chairman Award Winners
- Four years of Beta Testing
- Website: www.team1912.com
- Email: combustion@team1912.com





About Beta Testing



- Used by FIRST as a way to test new controls equipment, both hardware and software before it is released to the public at kickoff
- Beta teams work with FIRST to find and fix bugs in code or documentation
- Teams apply and are chosen based on geographic location as well as strength in the fields of controls, communication and mentoring
- Teams work through a series of preassigned tasks and post their results to FIRST forge



The Teams of Beta Testing

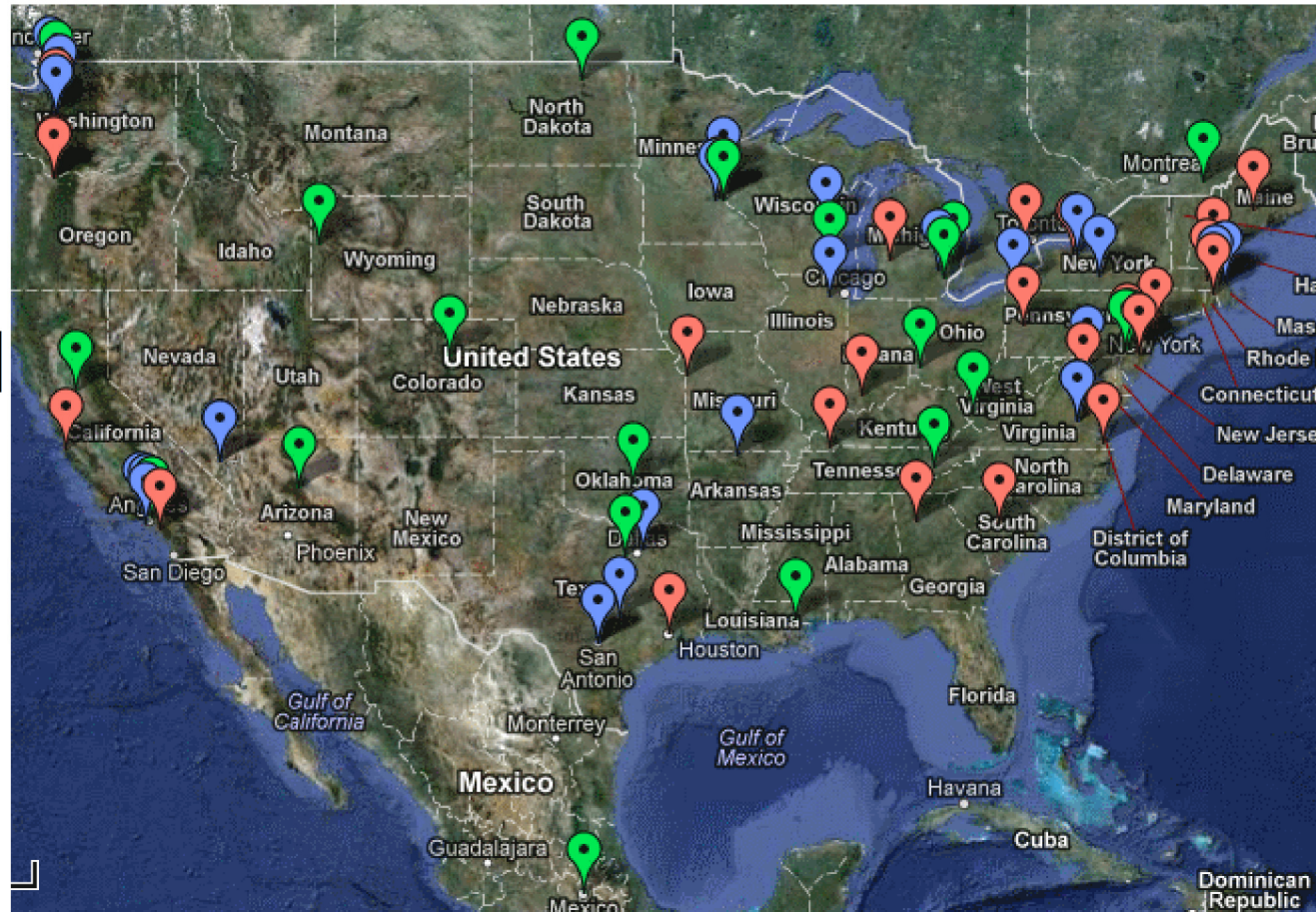


159 FRC Teams
and 1 FTC team
applied

Teams either used
C/C++, Java or
LabVIEW

25 teams per
language

Team 1912 was
selected to test LabVIEW
and hardware.



*In addition, 1 in Hawaii and 2 in Turkey



1912 Beta Test Team



- Controls Mentor: Wendy Holladay
- Beta Test Lead: Rachel Holladay
- Beta Test Members: Ben Rowley, Daniel Yue, James Yang, Yuanxia Lee, Alex Lew
- Special Thanks: Sam Holladay
(for chassis assistance)





LabVIEW Beta Tasks



- Task 1 – Install and Update Software
- Task 2 – Software Development Tools Verification
- Task 3 – Port Last year's Code into Beta Language
- Task 4 – Develop unique programs in Beta Language
- Task 5 – Hardware Evaluation - for teams evaluating 2013 hardware only



Software Changes: Overview



- New Image, v45
- Slight changes to Driver Stations
- Various tweaks to LabVIEW framework
- Addition of the 'Test.vi'
- Smart Dashboard and Dashboard Palette
- Robot Simulator





Re-Imaging Tool



FRC CompactRIO Imaging Tool : Version 2012.1.20

Select CompactRIO Device

MAC address	Name	Current IP	Current Image
00802F134ABA	thermite	10.19.12.2	FRC_2013_v45

Development Environment

☒ LabVIEW
☐ Always run deployed code at startup

☐ Wind River Workbench (C++)

☐ Java Technology

☐ Enable NetConsole

CAN Driver Plugin

☐ None

☒ Format Controller

Select Image

FRC_2013_v45.zip

Device name

thermite

Team ID

1912

cRIO Switches

<input checked="" type="checkbox"/>	SAFE MODE
<input checked="" type="checkbox"/>	CONSOLE OUT
<input checked="" type="checkbox"/>	IP RESET
<input checked="" type="checkbox"/>	NO APP
<input checked="" type="checkbox"/>	NO FPGA

Rescan Apply Close

Modules Installed

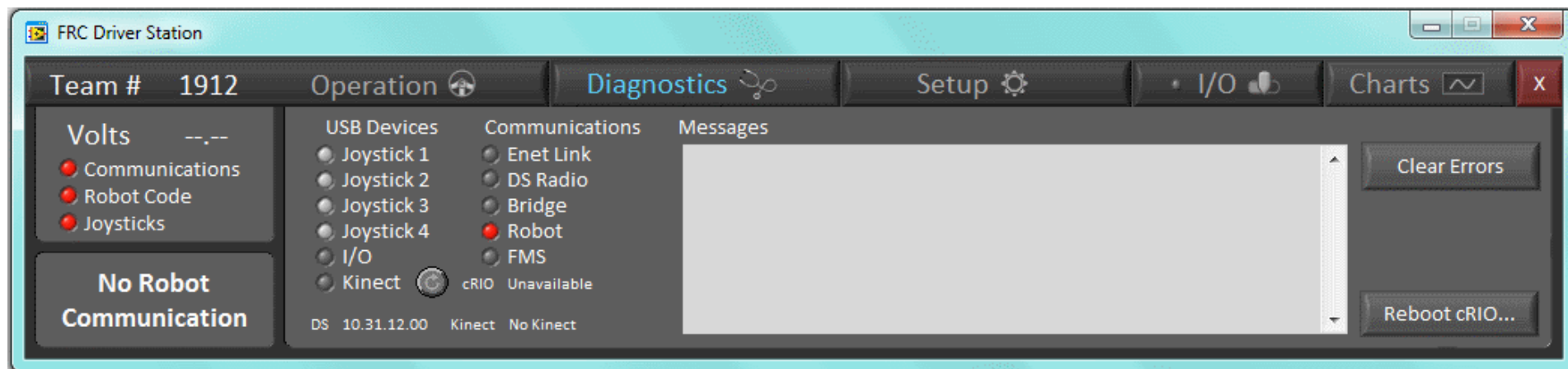
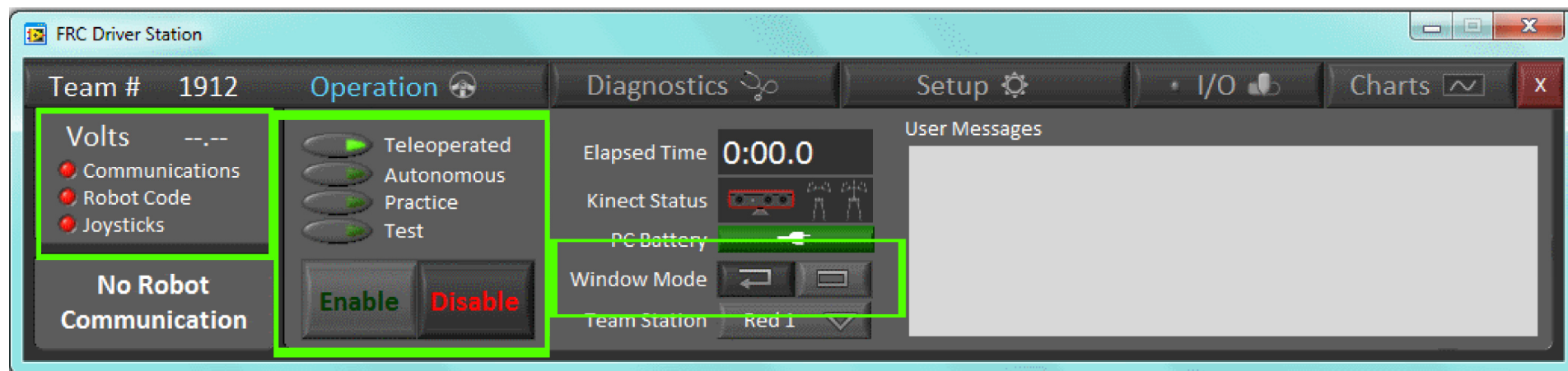
NI 9201
NI 9403
NI 9472
Empty

- The Re-Imaging tool had no significant changes from last year.



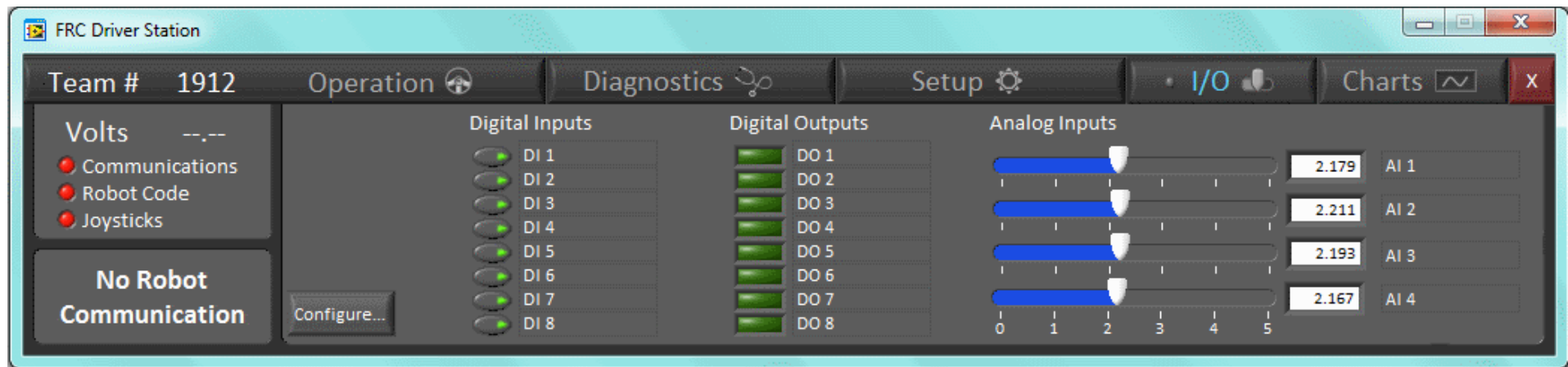
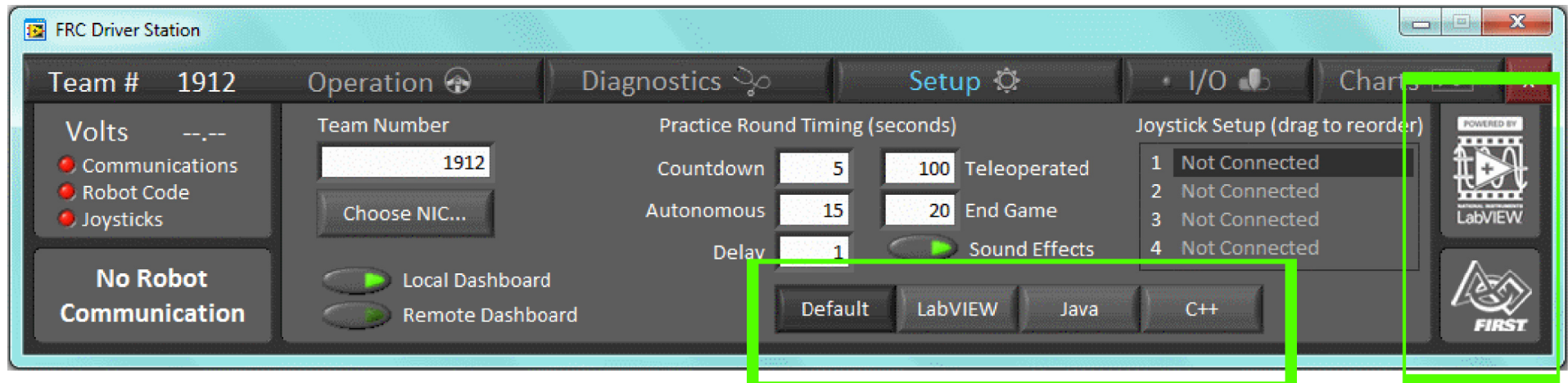


Driver Station Changes: Operation and Diagnostics



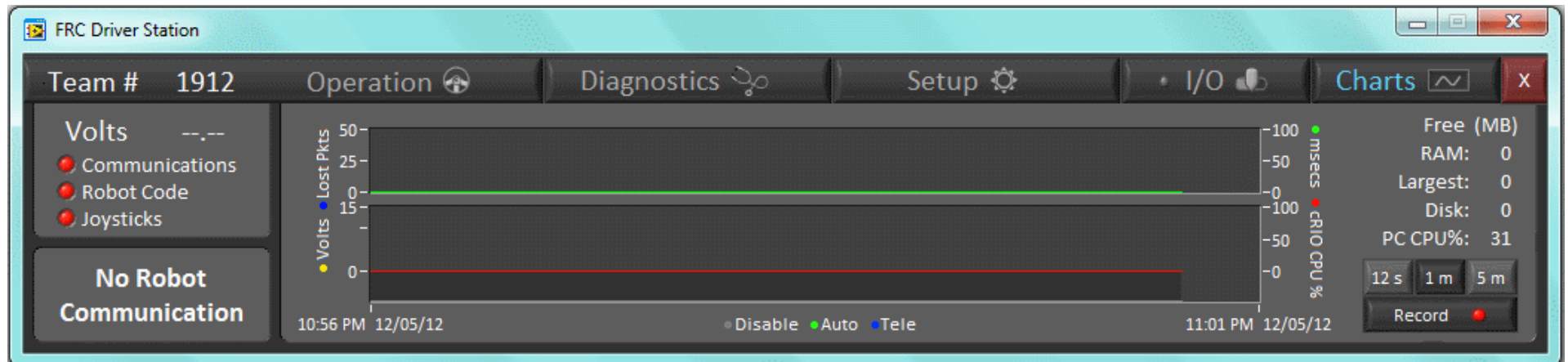


Driver Station Changes: Setup and I/O





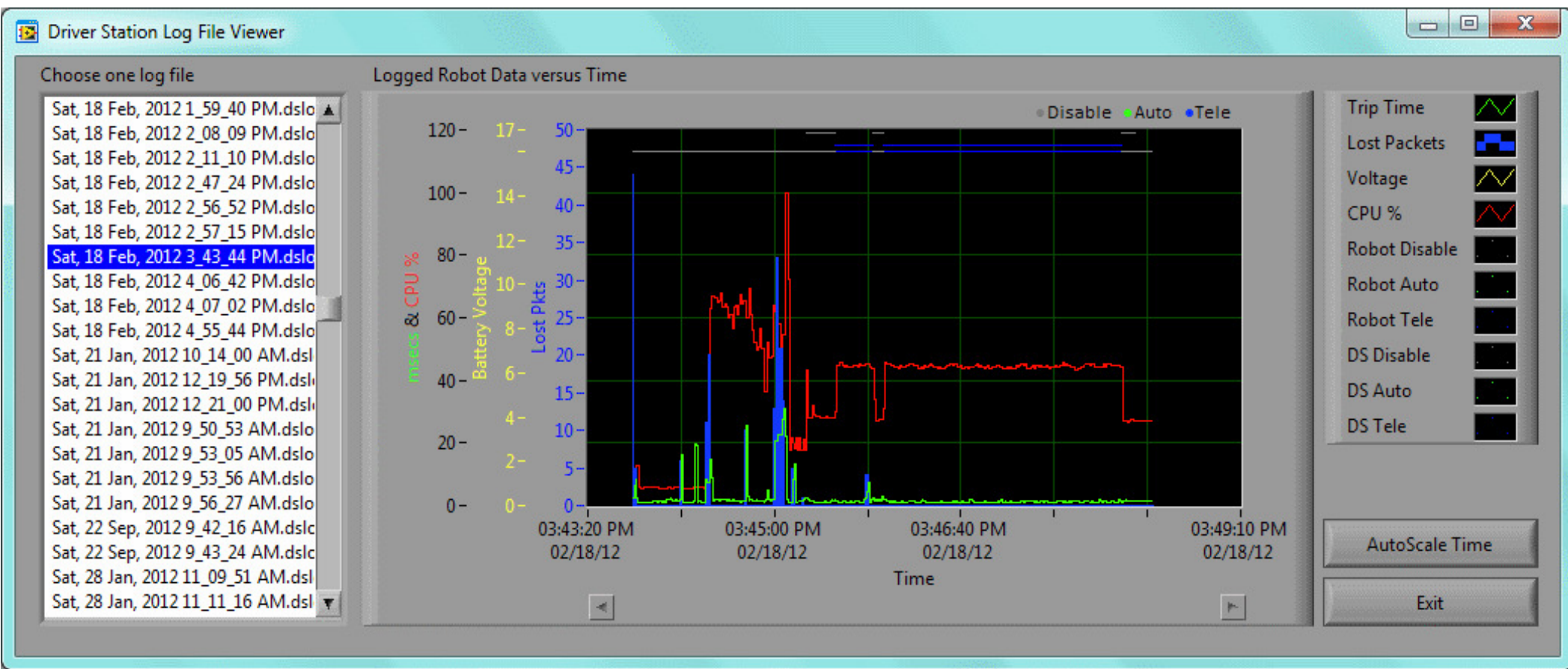
Driver Station Changes: Charts



- But now the Charts tab is not the only way to view Robot Communication data. The new Data Log Viewer gives an easy way to analyze data.



Driver Station Log Viewer





Dashboard Changes: Camera Image / Operation



FRC PC Dashboard

Camera Image | Kinect Skeleton | Variables

Drive & Motors

Operation | Test | Checklist

Customize by placing controls and indicators here and name to match Dashboard Variables on robot.

Checkbox 1 ☐ OFF/ON

Checkbox 2 ☐ OFF/ON

Slider 1

Random Numeric 1

Random Numeric 2

Random Numeric 3

Robot IP: 0.0.0.0

Battery

Enable ☒ 320x240 30fps 30% 0 Mbps 25 fps



Dashboard Changes: Kinect Skeleton / Test



FRC PC Dashboard

Camera Image Kinect Skeleton Variables

Drive & Motors

Robot IP: 0.0.0.0

Battery

Players 0
Version No Kinect

Kinect 1
X Y Z 1 2 3

Kinect 2
X Y Z 1 2 3

Operation Test Checklist

I/O Name	Value
Robot not in Test Mode	

Select the test to run...



Dashboard Changes: Variables / Checklist



FRC PC Dashboard

Camera Image Kinect Skeleton **Variables**

Variable Name	Value	Type
SmartDashboard		
Checkbox 1	False	Bool
Checkbox 2	False	Bool
Slider 1	0.000	Numeric

Drive & Motors

Rear Left

Rear Right

Front Left

Front Right

Robot IP: 0.0.0.0

Battery

Operation Test Checklist

1. Secure battery and connection
2. Ethernet cable from radio to cRIO
3. Radio switch position and LEDs
4. Remove safety pins

This list is read from a text file in
Public/Documents/FRC/Checklist.txt.

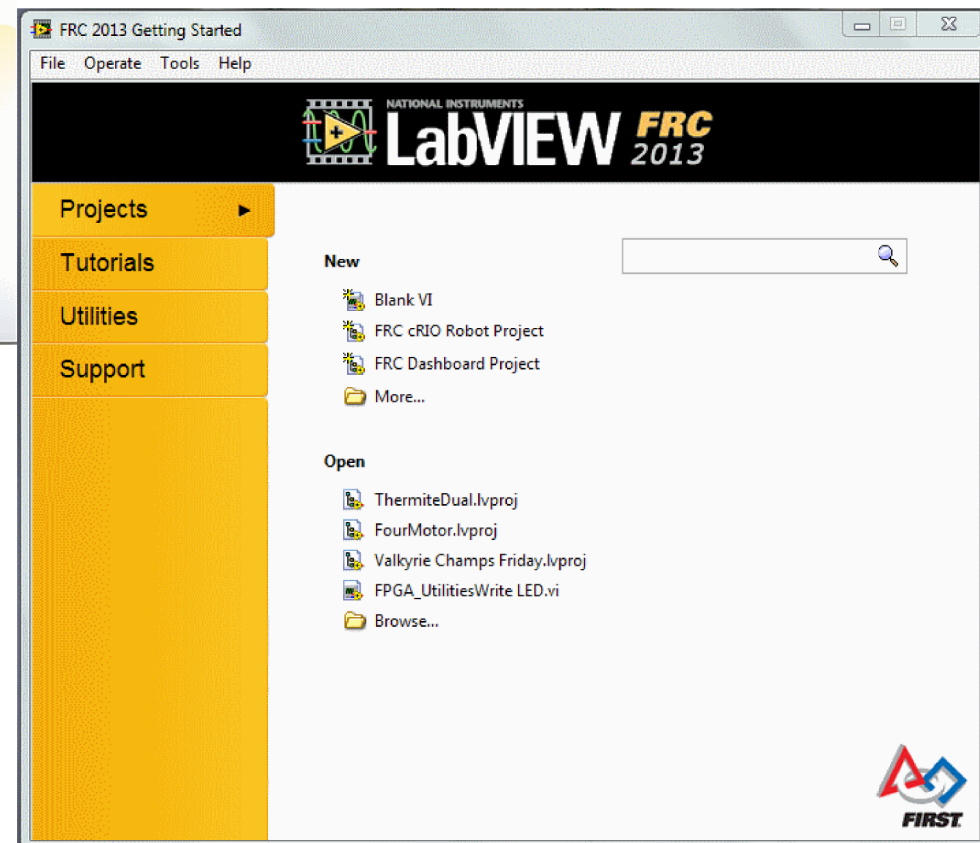
Modify the file to customize your list.

Double clicking checkmarks a list item.
Shift Double Click clears the checkmarks.

Rachel can edit this. Isn't that cool?



Framework Changes: LabVIEW Start-Up





Framework Changes: Create a New Project



Create New FRC Robot Project

Select project name, folder, and IP address

Project name
2013 Robot Project

Project folder
C:\Users\Rachel Holladay\Documents\LabVIEW Data\2013 Robot Project

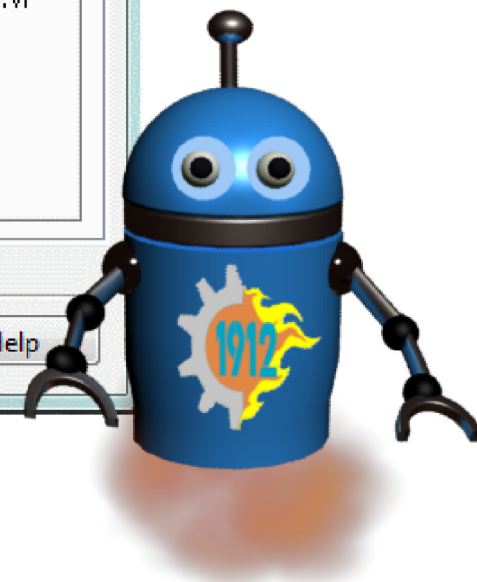
cRIO IP address
10.xx.yy.2

☒ Simple Arcade Drive Robot
☐ Arcade with Arm Robot
☐ Mecanum with Arm Robot

Project: Robot Project.lvproj

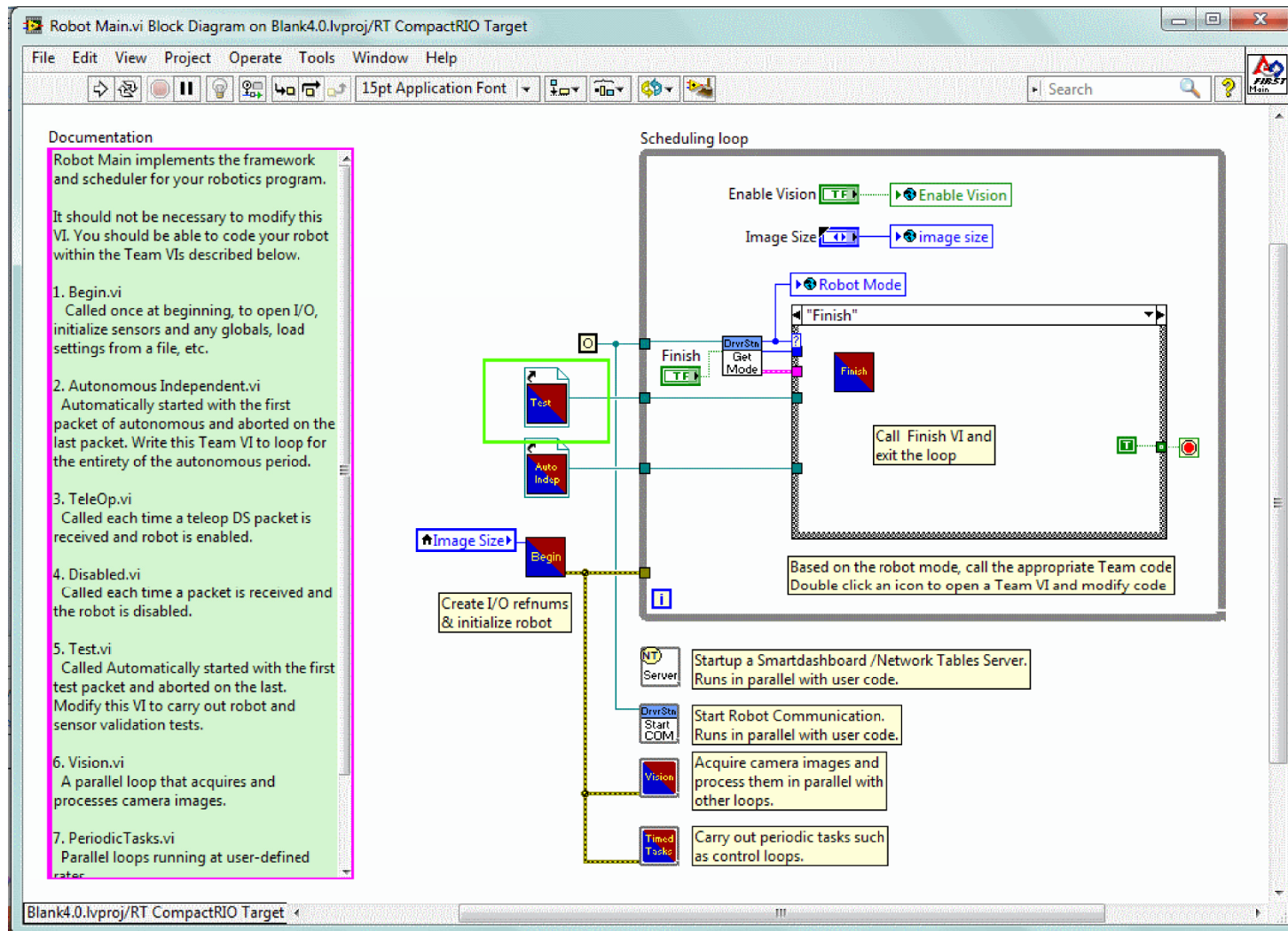
- My Computer
- RT CompactRIO Target (10.xx.yy.
- TypeDefs
- Team Code
 - Begin.vi
 - Autonomous Independent.v
 - Teleop.vi
 - Vision Processing.vi
 - Disabled.vi
 - Periodic Tasks.vi
 - Robot Global Data.vi
 - Build DashBoard Data.vi
 - Finish.vi
- Vision Support Code
- Support Code
- Robot Main.vi
- Dependencies

< Back Next > Finish Cancel Help



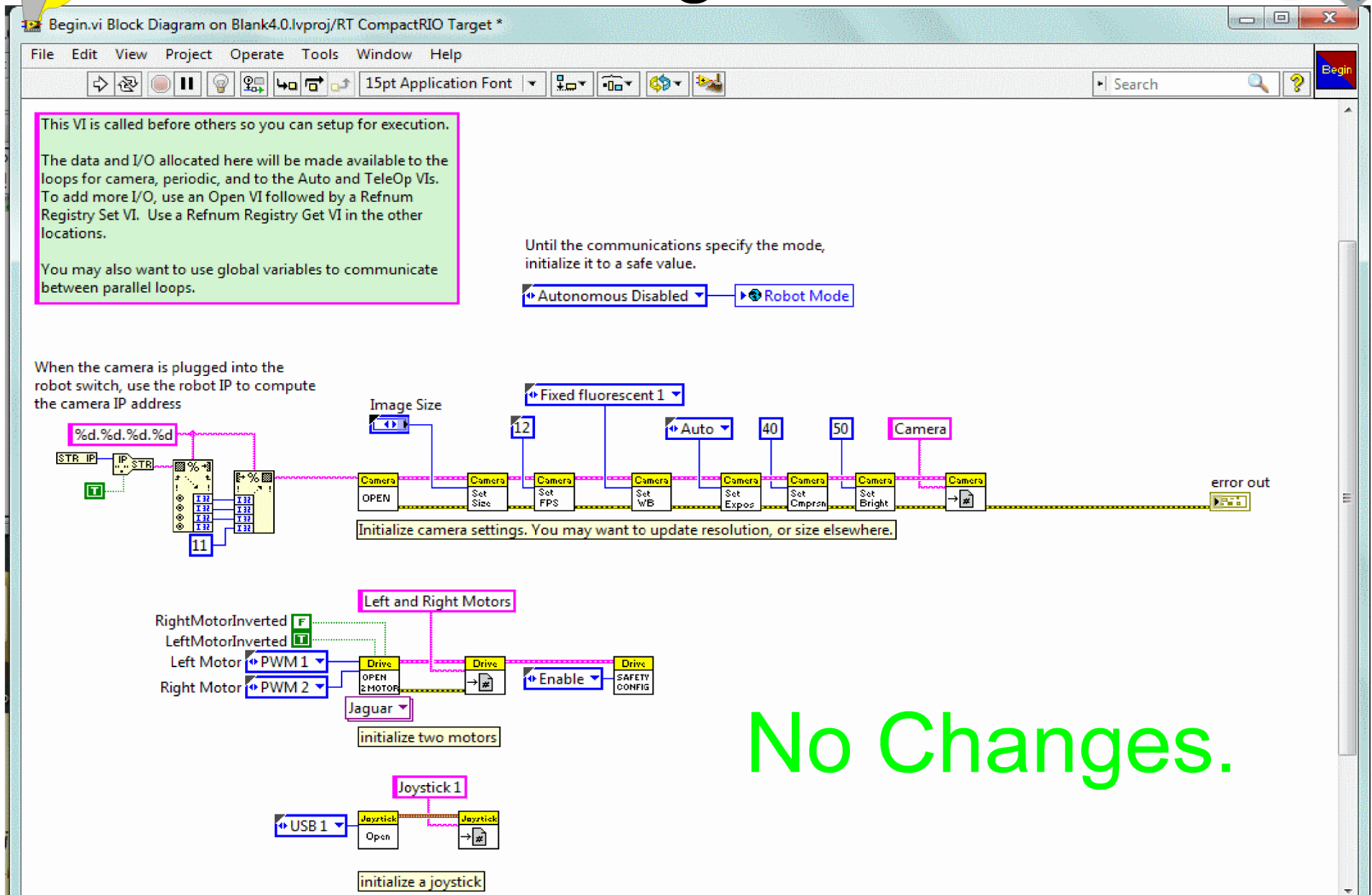


Framework Changes: Robot Main





Framework Changes: Begin





Framework Changes: Autonomous



Auto...endent.vi Block Diagram on Blank4.0.lvproj/RT CompactRIO Target *

File Edit View Project Operate Tools Window Help

9pt Application Font

Search

This function is run in parallel when the autonomous period begins. It will be terminated automatically when the period ends.

Note that all Driver Station inputs are automatically latched and will not change once auto begins.

DriverStn Get Alliance

This tells you the driver position and alliance color

DriverStn Get Dig In

This gives you the eight Driver Station digital switch settings

DriverStn Get Anlg In

This gives you the four Driver Station sliders or analog voltages

Joystick 1/Axes

Joystick 1

This gives you the joystick value that is often the throttle

Every 20ms, report that we are running autonomous code. Loop will abort the same time as the VI.

500 LED toggle rate (ms)

This code sets the toggle rate used by the Periodic Tasks VI. For autonomous mode we slowly flash the USER1 LED on the cRIO. Look at the Periodic Tasks VI to see how this global value is used

Autonomous sample code is disabled so your robot doesn't unexpectedly run in Autonomous mode. To enable this code, right-click on the edge of the frame and choose either "Remove Diagram Disable Structure" or "Enable This Subdiagram". You should only enable one of the samples.

Use time and/or sensors to control how the robot drives.

Disabled

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

RobotDrive Motors

Left and Right Motors

Drive

0.25

0.5

Standard

turn right for 1 second

0.5

0.25

Standard

turn left for 0.5 second

0

0

Standard

stand still

Disabled

Left and Right Motors

RobotDrive Motors

Array of robot movements

X Joystick Value

Y Joystick Value

Delay (sec)

Movement Name

Current Movement, for debugging

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.



Framework Changes: Tele-Op



Teleop.vi Block Diagram on Blank4.0.lvproj/RT CompactRIO Target *

File Edit View Project Operate Tools Window Help

9pt Application Font

Search

This VI is called each time a TeleOp DS packet is received. Use it to respond to new joystick or Driver Station values.

Common tasks include reading the joysticks and updating motors, and updating setpoints for periodic loops.

You can open I/O on the FIRST Call, or in the Begin.vi.

Match Info

Left → TeleOp Elapsed Seconds

This can help determine what has been run and for how long

Call Context

Use to differentiate between First, Last, and Intermediate calls

Each time we enter, report that we are running teleop

DriverSta Report Code State

100 → LED toggle rate (ms)

This code sets the toggle rate used by the Periodic Tasks VI. For teleop mode we flash the USER1 LED on the cRIO at a fast rate. Look at the Periodic Tasks VI to see how this global value is used.

Read Joystick X and Y values and update motor values

Left and Right Motors

Joystick 1

RobotDrive Motors

Drive Arcade Drive

Publish the Robot Drive Motor values to the dashboard

axis 1 (x)

axis 2 (y)

Joystick 1/Axes

Joystick 1/Buttons

Publish the joystick data the robot sees to the dashboard

Random Numeric 1

Random Numeric 2

Random Numeric 3

To the left are examples of writing dashboard variables. To the right are examples of reading them.

Checkbox 1

Checkbox 2

Slider 1

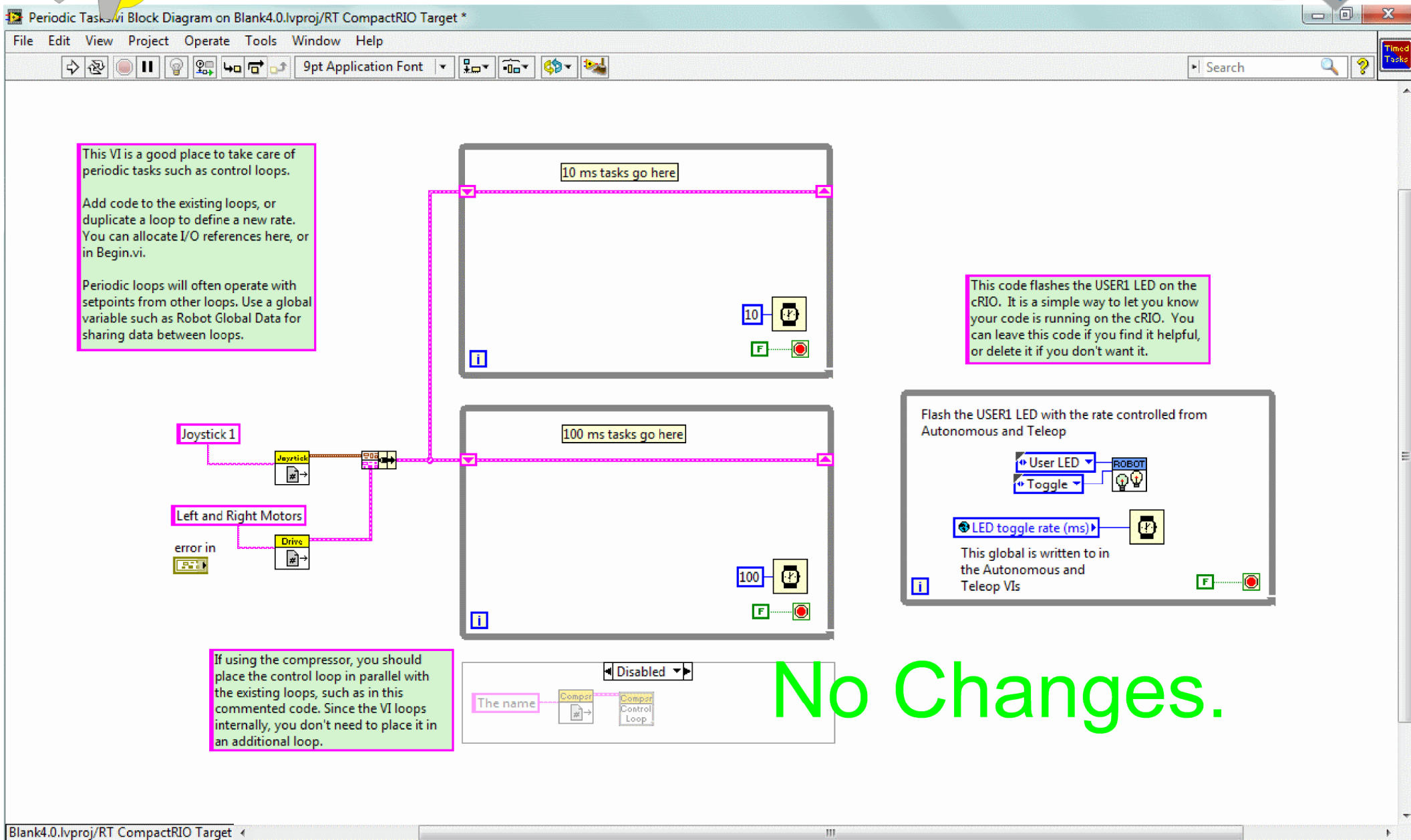
Boolean 1

Boolean 2

Numeric 1

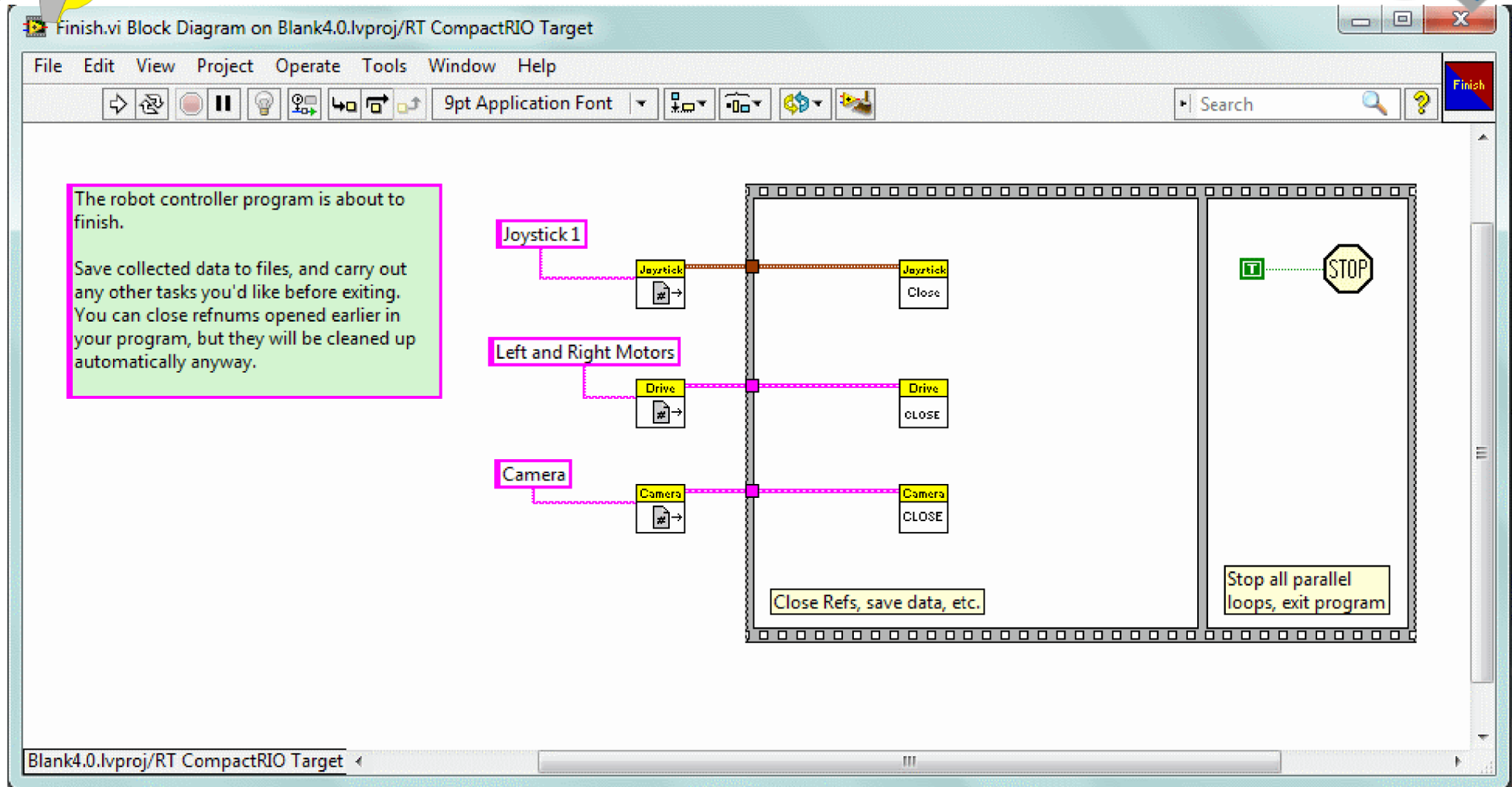


Framework Changes: Periodic Tasks

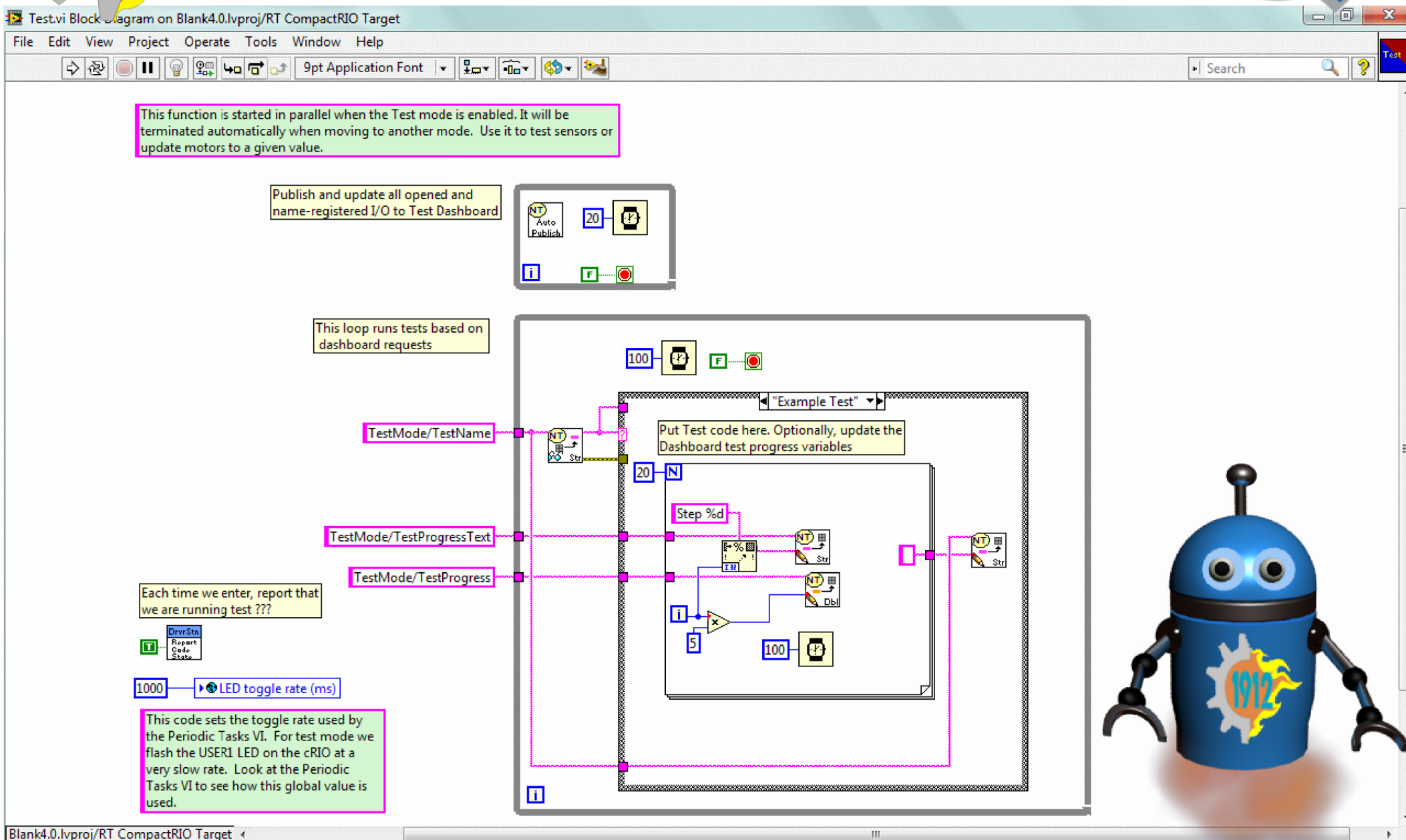




Framework Changes: Finish

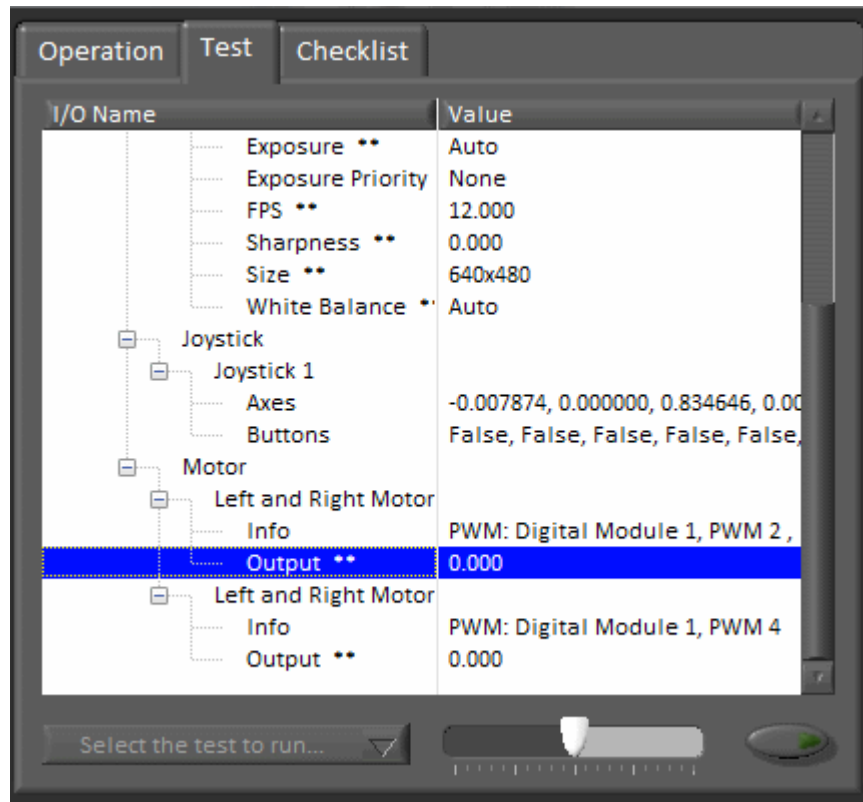


No Changes.

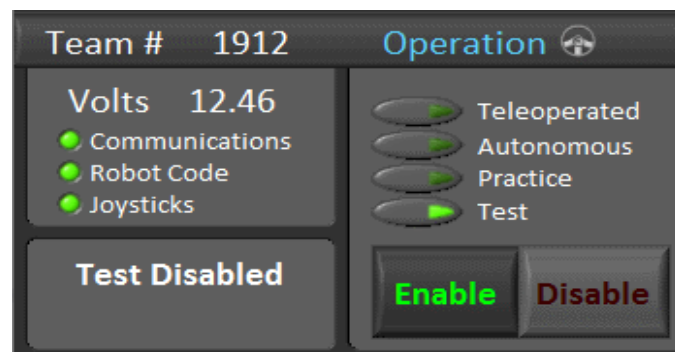




Purpose and Application of Test.vi

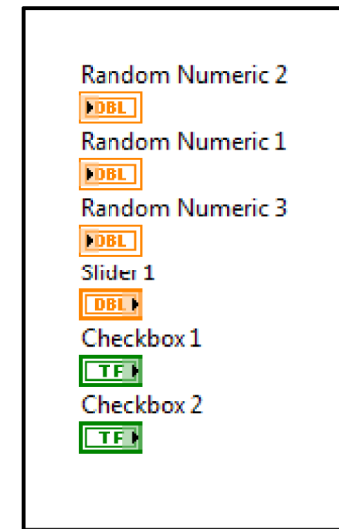
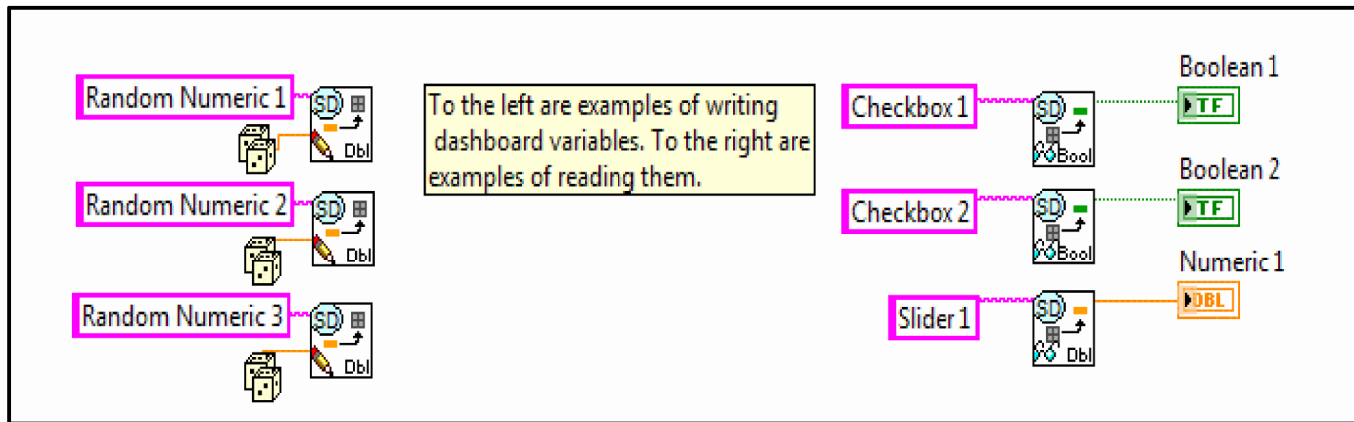


The new 'Test.vi' allows the coder to experiment by directly setting actuators (like motors) to certain values. You could also create little test snippets of code to run. Meant purely for debugging and can be in an enabled or disabled state.





Smart Dashboard



Inside Tele-Op values are fed into the Smart Dashboard, each with a unique name. The framework comes with some examples of how to read and write variables, as seen above. With read variables, the code includes an indicator on the Front Panel of the Tele-Op VI

Inside the Dashboard code those unique names are used to make indicators or controls



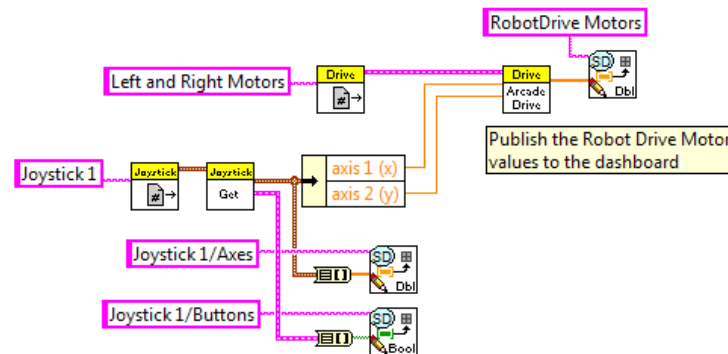
Thermite Smart Dash: Robot Code



This VI is called each time a TeleOp DS packet is received. Use it to respond to new joystick or Driver Station values.

Common tasks include reading the joysticks and updating motors, and updating setpoints for periodic loops.

You can open I/O on the FIRST Call, or in the Begin.vi.



Match Info

TeleOp Elapsed Seconds

This can help determine what has been run and for how long

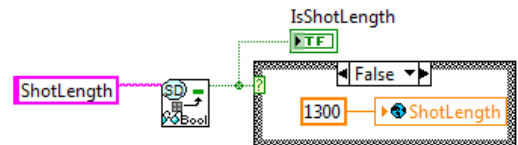
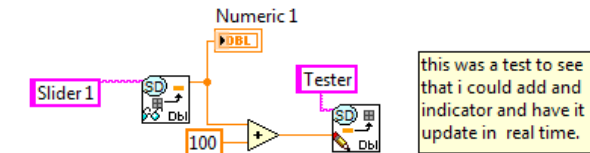
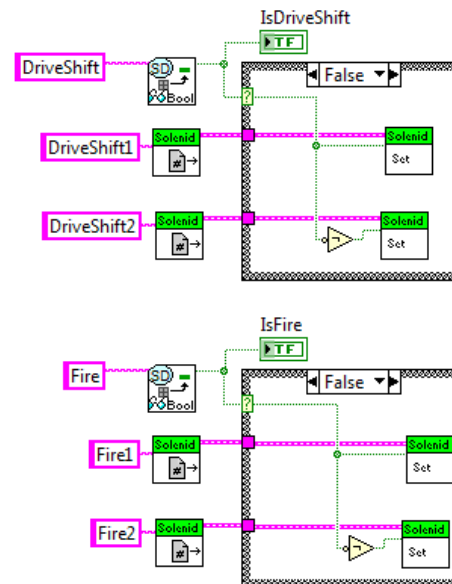
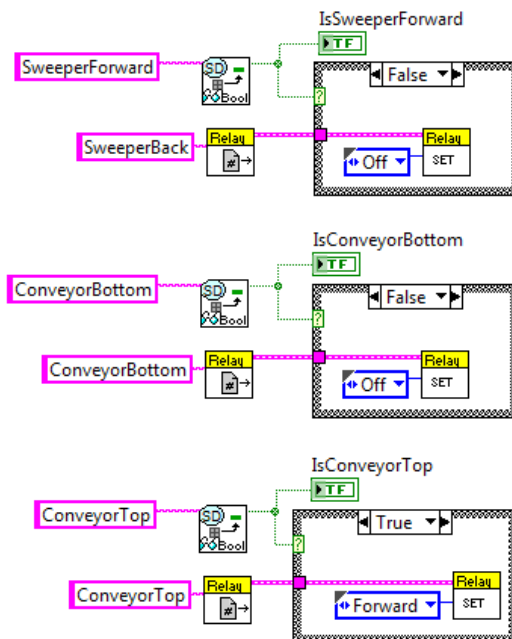
Call Context

Use to differentiate between First, Last, and Intermediate calls

Each time we enter, report that we are running teleop

100 LED toggle rate (ms)

This code sets the toggle rate used by the Periodic Tasks VI. For teleop mode we flash the USER1 LED on the cRIO at a fast rate. Look at the Periodic Tasks VI to see how this global value is used.





Thermite Smart Dash: Dashboard Code



FRC PC Dashboard

Camera Image | Kinect Skeleton | Variables

Drive & Motors

Operation | Test | Checklist

Customize by placing controls and indicators here and name to match Dashboard Variables on robot.

I can change the type of button and plan to later.

SweeperForward
☐ OFF/ON

ConveyorBottom
☐ OFF/ON

ConveyorTop
☐ OFF/ON

DriveShift
☐ OFF/ON

Fire
☐ OFF/ON

ShotLength
☐ OFF/ON

EncoderRate
0

Slider 1

Tester
0

Robot IP: 0.0.0.0

Battery

0 30fps 30% 0 Mbps 29.4 fps

Legend:

- DBL Slider 1
- DBL Tester
- TF SweeperForward
- TF ConveyorBottom
- TF ConveyorTop
- TF DriveShift
- TF Fire
- DBL EncoderRate
- TF ShotLength



Simulator: Overview



- The simulator allows you to program a predefined robot without having an RT CompactRIO Target. Programming is the same except only the specific I/O of the predefined robot is supported.
- Robot code that works in simulation mode can be moved to the RT Target and run on a real robot.
- Interfaces and controls in a video game like matter where you control the robot



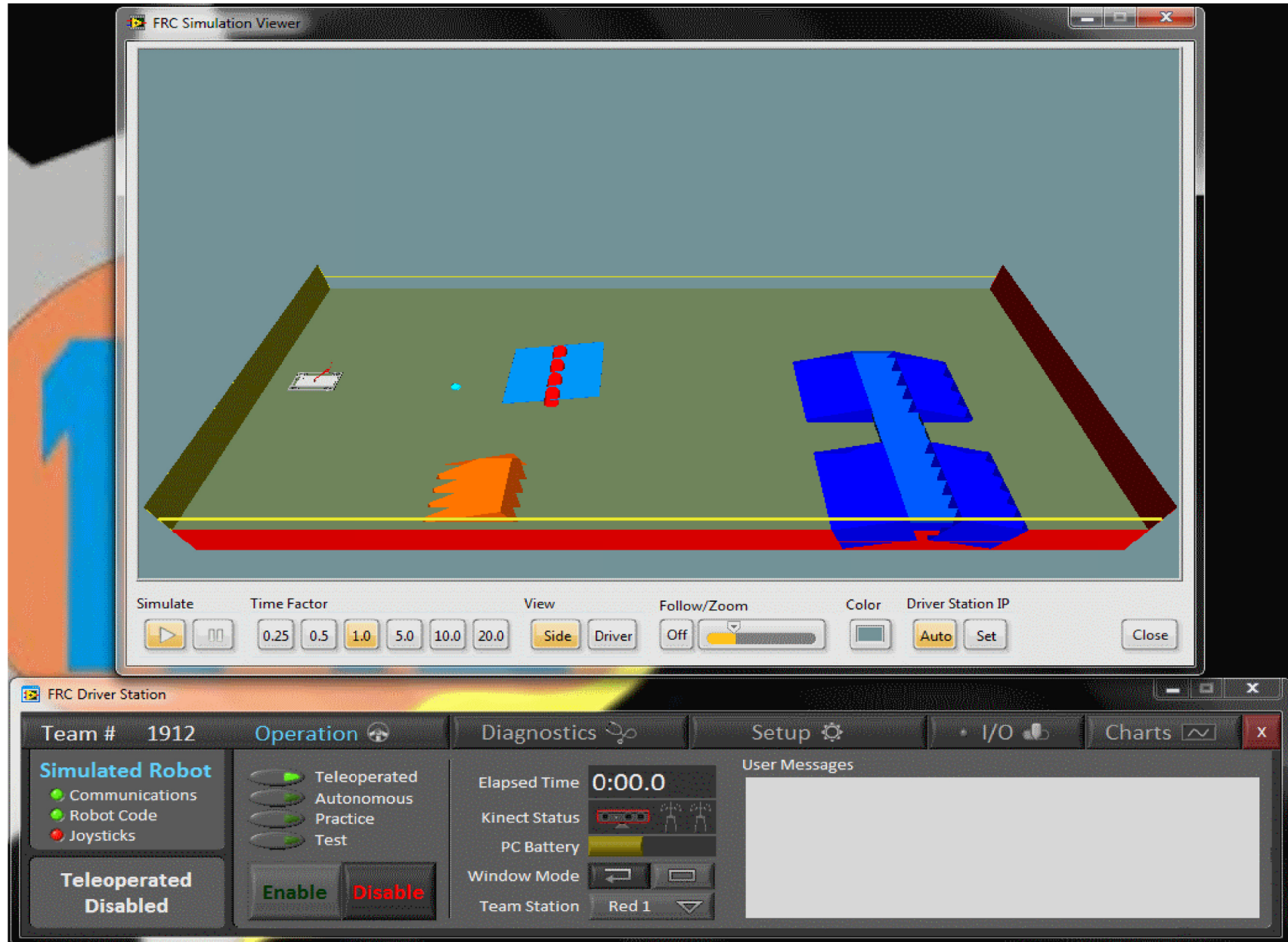
Simulator: Set-up



- To change from regular mode to simulator, simply change in Robot Main where the code runs (from RT cRIO to My Computer)
- The simulator is run with the Driver Station
- The code comes with a drive and then you can program in set actuators and sensors such as a claw servo, gripper servo, camera, encoder, gyro, etc.
- The outputs (such as the camera) read out to the dashboard like on the real robot



Simulator: Example Graphic





Simulator: A Teaching Tool



- The simulator allows the coder to practice coding without a physical robot, therefore allowing the learner to concentrate exclusively on the programming. The range of actuators and sensors gives the programmer the opportunity to experiment with many different systems. The video game like setting is not only interesting and fun but also gives immediate feedback. Currently the Robot Simulator is only available for LabVIEW.



Kinect



- Last year for Beta Testing we conducted extensive Kinect testing. We renewed our testing of the Kinect this year as Kinect testing was a part of Beta Testing but not a separate sub-set. Aside from some exploration into how the Kinect views players, there was little testing to do. There was little sizable changes to the Kinect software and performance. The only notable difference was that we had to install this years Kinect software.



New Motor Controllers



- Legal Motor Controllers for this year: (As seen on the 3 October 2013 FRC Blog Entry)
 - Jaguars
 - Talons
 - Victor 888
- Rookies will get 1 Jaguar and 2 Victors in their kit and Veterans will get a PDV from IFI for 2 motor controllers of their choice
- We beta tested the new Victor 888



Victor 888



- Essentially it looks very similar to the Victor 884 and has a similar footprint
- Required a simple calibration
- In comparative qualitative testing the Victor 888 seems to respond a little faster than the Jaguar



Victor 888 Light Patterns



- When code is not running: Orange Blink
- When the controller is neutral: Orange Solid
 - Partial forward or reverse: No Light
 - Full Forward: Green Solid
 - Full Reverse: Red Solid
- The follows the same pattern as Victor 884s and Jaguars with the exception of partial control



Questions?

