

# GETTING STARTED WITH THE CONTROL SYSTEM

# Getting Started with the Control System

## Table of Contents

<b>Getting Started with the ScreenSteps Documentation .....</b>	<b>4</b>
Getting Started With the Screen Steps Documentation .....	5
<b>System Overview.....</b>	<b>7</b>
New for 2019!.....	8
Known Issues.....	14
Offline Installation Preparation.....	17
FRC Control System Hardware Overview.....	19
FRC Software Component Overview .....	40
3rd Party Libraries .....	56
<b>Getting Started: Step-By-Step.....</b>	<b>60</b>
Wiring the FRC Control System .....	61
Imaging your Classmate (Veteran Image Download).....	83
Installing LabVIEW for FRC 2019 (LabVIEW only) .....	104
Installing the FRC Update Suite (All Languages).....	127
Installing C++ and Java Development Tools for FRC.....	148
Imaging your roboRIO .....	163
FRC Driver Station Powered by NI LabVIEW .....	170
Running your Benchtop Test Program - Tethered.....	181
Programming your radio .....	183
Running your Benchtop Test Program - Wireless.....	199
Updating and Configuring Pneumatics Control Module and Power Distribution Panel.....	201

# Getting Started with the Control System

Getting Started .....	202
<b>System Details .....</b>	<b>204</b>
RoboRIO Networking.....	205
Networking Basics .....	209
2018 Game Data Details .....	219

# Getting Started with the ScreenSteps Documentation

# Getting Started With the Screen Steps Documentation

ScreenStepsLive is a new tool that FRC/WPI are using to create and present documentation. This document is a brief introduction to the ScreenStepsLive site and the documentation contained here.

## What's Here?

The documentation on the ScreenStepsLive site encompasses a number of potentially familiar documents from previous seasons such as the Getting Started with the 201X Control System, Getting Started with C++, Getting Started with Java, WPIlib Cookbook, Vision Whitepaper and more. It also includes quite a bit of brand new documentation such as the Control System Software and Hardware Overviews, documentation on new features or tools such as Robot Builder and Live Window/Test Mode, and new documentation on existing tools such as Getting Started With the SmartDashboard.

## Navigating the Site

### Navigating the Site

The documentation is organized into a hierarchy with Sections at the very top, followed by Manuals, Chapters, then Lessons. At any time while you are browsing through the documentation, you can use the navigation at the top of the screen to go back to the Manual or to the home screen. You can also use the navigation on the left side of the screen when viewing a Manual or Chapter to jump to a different Manual. Each article also has a Prev and Next link at the top and bottom of the article to take you to the previous article or next article in the Manual.

# Getting Started with the Control System

## Using the Search

### Using the Search

A search bar is located at the top of each page which you can use to search the site. After entering a search query you will be brought to the search results page. From this page you can refine your query by selecting whether to "match any" or "match all" terms in the search. You can also narrow your search to specific manuals by checking them in the left pane.

## Downloading PDFs

### Downloading PDFs

For offline viewing, every Manual in the documentation can be downloaded as a PDF. From the manual page or from any of the Lessons within the manual you can download the manual PDF by clicking the link on the left side of the window. Additionally, some individual Lesson PDFs can be downloaded from the lesson pages.

## Getting Started with the Control System

# System Overview

# Getting Started with the Control System

## New for 2019!

This article describes the major changes to the Control System software for FRC 2019

## Major Changes

- **New IDE - VSCode (replacing Eclipse). This also includes a change from the Ant build system to GradleRIO**
  - Vendor Libraries
- Pathweaver path planning software
- Raspberry PI image for image processing
- New Shuffleboard API
- New Java version - Java 11
- New roboRIO Webdashboard

## Game Specifics

- Bandwidth - Bandwidth limit has been reduced to 4 Mbps. Like airline and hotel booking, the 7 Mbps limit included some assumptions that not all teams would use it. With the Sandstorm driving expected camera usage, we have lowered the limit in attempt to limit the likelihood teams will need to reduce below the limit at an event.
- Sandstorm Coding
  - Mode - Despite the Sandstorm not truly being an "Autonomous" mode, all code (DS, robot code, etc.) will continue to refer to this period as "Autonomous" or "Auto" for continuity.
  - Mode Transitions - The field will transition through modes in the same way as prior seasons. Prior to the start of the match, robot's will be in Disabled Mode. Once the match starts, they will transition to Autonomous Enabled mode. At the end of the Sandstorm, robots will briefly transition through Disabled (~.5s), if team's are driving their robot with joysticks/ gamepads at this time, there will be a noticeable hitch. Finally the robot will transition to Teleop Enabled mode for the remainder of the match until the buzzer sounds. When the match expires, the robot will return to Disabled mode.

# Getting Started with the Control System

- Joystick Access - For this season only, joystick data will not be cached on the transition to Autonomous. Teams will be able to access updated joystick data throughout the Autonomous/Sandstorm period.
- Game Data - There is no game data for the 2019 season.

## VS Code IDE (C++\Java)

For the 2019 season, FRC is moving from Eclipse to VS Code as the officially supported IDE for C++\Java development. This setup is easier for us to develop the extensions for and we expect it to be more streamlined and functional for the majority of teams. If VS Code doesn't seem to be your thing, the accompanying switch to GradleRIO should make it easier to use alternate IDEs as well (though this is not officially supported).

You can find information about the new IDE in the **Setting up the Development Environment** and **Creating and Running Robot Programs** sections of the [C++](#) and [Java](#) manuals, including information about how to import old Eclipse projects into the new VSCode/GradleRIO system

## Vendor Libraries

- With the change in IDE\Build System there was also a change to how Vendor libraries integrate with the system. For more information, see here: [3rd Party Libraries](#)

## RoboRIO WebDashboard

- No longer requires Silverlight
- No longer supports CAN device status/configuration
- Recommended to be used in non-IE browser

More information [here](#)

## RoboRIO Imaging Tool

- Reworked UI
- Added firmware update capability

More information [here](#)

# Getting Started with the Control System

## CAN Device Configuration

- Removed from WebDashboard
- Provided by vendor specific tools.

See [this article for updated PDP\PCM configuration instructions](#)

## WPILib C++\Java

- A new high level CAN API was added. This enables easier creation of custom CAN devices for users.
- The OSSerialPort class was removed. It was a bugfix for some Serial port bugs, but never actually worked anyway.
- SensorBase was removed. If you were deriving from SensorBase you should now derive from SendableBase (C++) or SendableBase+ErrorHandler(Java). If you need methods previously contained in SensorBase, they are now in SensorUtil.
- The Watchdog class was added. This calls a callback function when the specified timeout is exceeded. This class is used by TimedRobot to warn on periodic loop time overruns (nominally 20ms) in user code. **Note: These Timed Robot messages are warnings. They inform you if your code overruns the specified timing. They are not errors.**
- IterativeRobot was deprecated for eventual removal. TimedRobot is a drop-in replacement that is strictly better with regard to execution jitter.
- Linux was updated to require Ubuntu 18.04 as its default OS to run WPILib tools
- Building the libraries:
  - All libraries require C++14 capability to build. On Linux, this means GCC 6, and on Windows MSVC 2015 or newer.
  - Linux was updated to require Ubuntu 18.04 as its default OS to link to the libraries.
  - Windows builds are done with MSVC 2017 update 9, so linking to the libraries requires 2017 update 9 or later.

## WPILib C++

- The namespace frc shim was removed. WPILib classes need to be accessed by their full namespaces, or using statements added.
- **Headers were moved to an frc folder.**

# Getting Started with the Control System

- HAL methods were removed from WPILib headers. In order to access HAL methods, their headers must be directly included.
- Llvm renamespaced to wpi, headers moved to wpi folder
- All classes now properly support move semantics (for those who know what that is and care).
- Doxygen comments were moved from the source files to the header files for easier reference.

## WPILib Java

- The CameraServer class was moved to a new package, and the class in the old location was deprecated. Please move to the new package location.
- The JNI classes were moved to a new package.
- Any Java class that had a free method was changed to implement AutoClosable and have a close method instead.
- The main method was moved from being internal in WPILib to explicitly defined by user code. This removes the reflection loading of the main robot class, and makes changing your robot class a compile time error rather than a runtime error. **Teams should not need to edit the Main.java file.**

## CameraServer (cscore)

- USB cameras are now supported on Windows desktop builds

## All WPILib Tools (Shuffleboard, Robot Builder, etc.)

- Are now installed to ~home\frcYYYY\tools (where YYYY is the year and ~home on Windows is C:\Users\Public).
- Run ToolsUpdater.bat (Windows) or ToolsUpdater.py (Mac/Linux) or **Install Tools from GradleRIO** to install tools.
- Folder contains .vbs files for Windows and .sh files for Mac/Linux that should be used to run the program. This sets the program up to run using the FRC specific JDK (which it has been tested with).

# Getting Started with the Control System

## Shuffleboard

- New roboRIO API for automatically placing widgets on tabs and setting formatting options.  
More info [here](#)
- Camera viewer widget with adjustable stream parameters

## PathWeaver\Pathfinder

- Added PathWeaver as UI to create paths for Pathfinder V1
- Generated paths are automatically downloaded to the RIO as part of the gradle configuration

Find more documentation on Pathweaver [here](#)

## Raspberry Pi Image for Cameras

A pre-made Raspberry Pi Image for camera streaming/image processing has been developed to lower the barrier to entry to off-board vision processing. This image contains all of the libraries required to implement FRC compatible camera streaming, as well as a helpful web dashboard, read only file system configuration to handle robot power off and more. Learn more in the new manual [here](#).

## SmartDashboard

- Support for plugins has been removed. It was prohibitively difficult to maintain this feature when moving to Java 11 for the minimal number of teams believed to still be using it.
- Requires Java 11 to run.

## Robot Builder

- Generates projects for the new VS Code\GradleRIO system
- Java/C++: encapsulates hardware in the specific subsystem. RobotMap is removed.
- C++: Updated to idiomatic C++ to match VSCode examples.

# Getting Started with the Control System

## Outline Viewer

- Requires Java 11 to run.

# Getting Started with the Control System

## Known Issues

This article details known issues (and workarounds) for FRC Control System Software.

### VSCode failing to launch offline (no network present)

There are reports that on some builds of Windows 10 (build 1809), VSCode may fail to launch if no network is present (note: this is different than no internet present).

Workaround: Install the Microsoft Loopback Adapter

1. Access the Device Manager (typically by clicking start and typing Device Manager)
2. Click the Network Adapter Category
3. Click the Action menu from the top bar and select Add Legacy Hardware
4. Click Next on the window that pops up
5. Select the second option to install manually
6. Select "Network Adapter" and click Next. You will need to wait a bit for the next screen to populate.
7. Select Microsoft in the Manufacturer pane, then Microsoft Loopback Adapter in the right pane.
8. Click Next twice to install the adapter, then Finish to close the window.

### Driver Station Dashboard launching

Issue: When selecting SmartDashboard or Shuffleboard in the FRC Driver Station, the programs fail to launch.

Workaround: Use the Desktop icons to launch the desired dashboard manually.

Solution: Fixed in [2019.1.0 NI Update Suite](#)

# Getting Started with the Control System

## C++ Intellisense - Files Open on launch don't work properly

Issue: In C++, files open when VS Code launches will have issues with Intellisense showing suggestions from all options from a compilation unit and not just the appropriate ones. This is a bug in VS Code

Workaround: Close the files in VS Code, close VS Code, wait ~ 1 min, re-launch VS Code.

## Auto SPI Does not work in v12 image (affects Analog Devices IMUs and Gyro)

The Auto SPI functionality used by Analog Devices IMUs (ADIS16448 and ADIS16470) and Gyro (ADXRS450) does not work correctly in the 2019-v12 image that is part of the kickoff release.

Workaround: After imaging, log into the robot console as admin (via serial or [SSH](#)) and run the command "updateNIDrivers", then reboot. A simplified tool to execute this is in progress and more information about a full update with this issue resolved will be coming soon.

Solution: Fixed in 2019v13 image in [2019.1.0 NI Update Suite](#)

## SmartDashboard and Simulation fail to launch on Windows N Editions

Issue: WPIlib code using CSCore (dashboards and simulated robot code) will fail to launch on Education N editions of Windows.

Solution: Install the Media Feature Pack from <https://www.microsoft.com/en-us/software-download/mediafeaturepack>

## Installing and Using GradleRIO Directly May not Deploy JRE

If user installs GradleRIO directly (i.e. not from the offline installers or zips provided by the WPIlib release) and then runs a deploy to the roboRIO without internet, the JRE may not be deployed. A single deploy with internet connected (connect to the roboRIO over USB) is needed to download

# Getting Started with the Control System

the JRE to the local cache. Subsequent deploys (or if the roboRIO already has the JRE deployed) will succeed.

Solution: Fixed in WPILib 2019.2.1

## PathWeaver Tank Drive swapped paths

There is currently an issue with PathWeaver where the left and right sides of the drivetrain are swapped. This can be solved quickly with the instructions in this article: [Integrating path following into a robot program](#). This will be fixed in the next update.

## Network Tables Interoperability

There is currently an issue with inter-operating C++\Java Network Tables clients (dashboards or co-processors) with LabVIEW servers (LV robot code). In some scenarios users will see updates from one client fail to be replicated across to other clients (e.g. data from a co-processor will not be properly replicated out to a dashboard). Data still continues to return correctly when accessed by code on the server.

Workaround: Write code on the server to mirror any keys you wish to see on other clients (e.g. dashboards) to a separate key. For example, if you have a key named "targetX" being published by a co-processor that you want to show up on a dashboard, you could write code on the robot to read the key and re-write it to a key like "targetXDash".

# Getting Started with the Control System

## Offline Installation Preparation

This article contains instructions\links to components you will want to gather if you need to do offline installation of the FRC Control System software.

## Documentation

The screenshot shows the official FIRST FRC Control System documentation website. The sidebar on the left lists several topics:

- Getting Started with the ScreenSteps Documentation (1 topic)
- Getting Started with the 2016 Control System (19 topics)

Below the sidebar, there's a note about the documentation being powered by ScreenSteps Live, a "Learn more about ScreenSteps Live" button, and a "WPILib Supporters" section. A red arrow points from the text below to the "Download Manual PDF" link at the bottom of the sidebar.

The main content area displays the "Getting Started with the ScreenSteps Documentation" page, which includes a single bullet point: "Getting Started With the Screen Steps Documentation".

Further down, the "Getting Started with the 2016 Control System" page is shown, listing various sub-topics such as "New for 2016", "2016 FRC Control System Hardware Overview", and "Wiring the 2016 FRC Control System".

At the bottom of the main content area, there's a section titled "FRC Driver Station Powered by NI LabVIEW" with three bullet points: "Running your Benchtop Test Program - Tethered", "Programming your radio for home use", and "Running your Benchtop Test Program - Wireless".

The ScreenSteps documentation can be downloaded for offline viewing. At a minimum you will want to get a copy of the Getting Started with the Control System manual, you may also wish to download some or all of the other manuals as well. The link to download the PDF of a Manual is located on the left sidebar of the Manual page or any Lesson in the Manual. See [Getting Started With the Screen Steps Documentation](#) for more details about the ScreenSteps site and content organization.

# Getting Started with the Control System

## Installers

All Teams:

- [2019 NI Update Suite](#) (Note: Requires decryption key from kickoff broadcast!)
- [2019 FRC Radio Configuration Utility](#) **OR** [2019 FRC Radio Configuration Utility Israel Version](#)
- (Optional - Veterans Only!) [Classmate/Acer PC Image](#)

LabVIEW Teams

- LabVIEW USB (from FIRST Choice) or [Download](#)

C++\Java Teams

- [C++\Java WPILib Installer](#)
- VS Code (if using Windows, run the installer and use it to download the appropriate VS Code file. If using Mac/Linux, download VSCode from here: <https://code.visualstudio.com/download>)

## 3rd Party Libraries/Software

A number of software components were broken out of WPILib in 2017 and are now maintained by third parties. See [this blog](#) for more details.

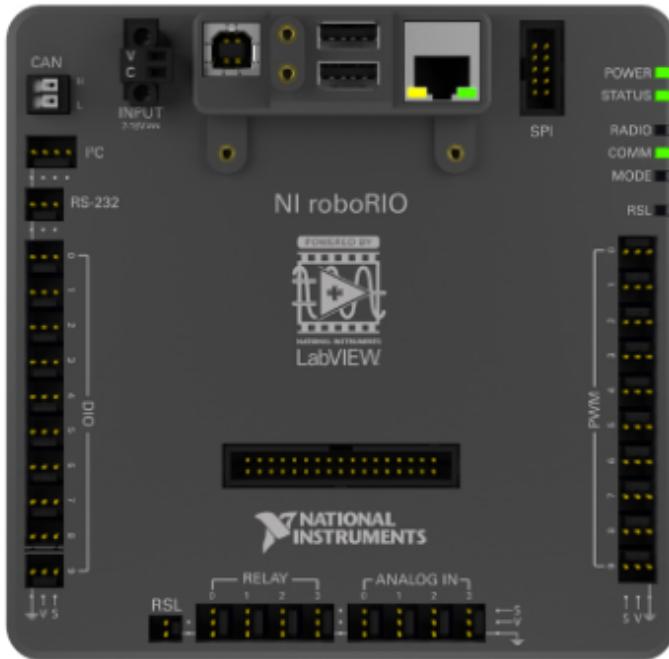
A "directory" of available 3rd party software that plugs in to WPILib can be found on [this page](#).

# Getting Started with the Control System

## FRC Control System Hardware Overview

The goal of this document is to provide a brief overview of the hardware components that make up the FRC Control System. Each component will contain a brief description of the component function, a brief listing of critical connections, and a link to more documentation if available. Note that for complete wiring instructions/diagrams, please see the [Wiring the FRC Control System](#) document.

### National Instruments roboRIO



The NI-roboRIO is the main robot controller used for FRC. The roboRIO includes a dual-core ARM Cortex™-A9 processor and FPGA which runs both trusted elements for control and safety as well as team-generated code. Integrated controller I/O includes a variety of communication protocols (Ethernet, USB, CAN, SPI, I2C, and serial) as well as PWM, servo, digital I/O, and analog I/O channels

# Getting Started with the Control System

used to connect to robot peripherals for sensing and control. The roboRIO should connect to the dedicated 12V port on the Power Distribution Panel for power. Wired communication is available via USB or Ethernet. Detailed information on the roboRIO can be found in the [roboRIO User Manual](#).

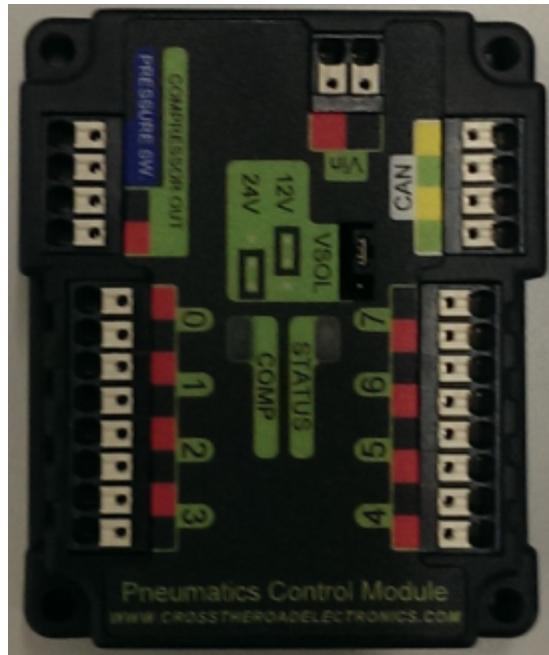
## Power Distribution Panel



The Power Distribution Panel (PDP) is designed to distribute power from a 12VDC battery to various robot components through auto-resetting circuit breakers and a small number of special function fused connections. The PDP provides 8 output pairs rated for 40A continuous current and 8 pairs rated for 30A continuous current. The PDP provides dedicated 12V connectors for the roboRIO, as well as connectors for the Voltage Regulator Module and Pneumatics Control Module. It also includes a CAN interface for logging current, temperature, and battery voltage. For more detailed information, see the [PDP User Manual](#).

# Getting Started with the Control System

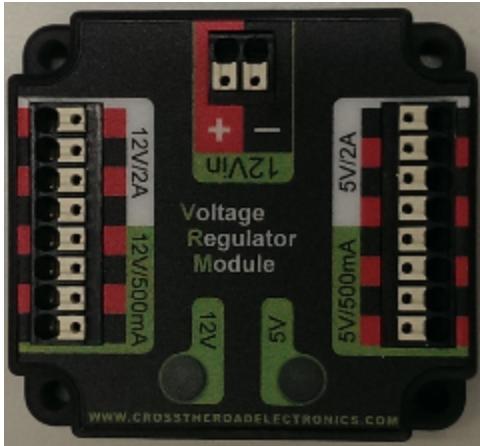
## Pneumatics Control Module



The PCM is a device that contains all of the inputs and outputs required to operate 12V or 24V pneumatic solenoids and the on board compressor. The PCM is enabled/disabled by the roboRIO over the CAN interface. The PCM contains an input for the pressure sensor and will control the compressor automatically when the robot is enabled and a solenoid has been created in the code. The device also collects diagnostic information such as solenoid states, pressure switch state, and compressor state. The module includes diagnostic LED's for both CAN and the individual solenoid channels. For more information see the [PCM User Manual](#).

# Getting Started with the Control System

## Voltage Regulator Module



The VRM is an independent module that is powered by 12 volts. The device is wired to a dedicated connector on the PDP. The module has multiple regulated 12V and 5V outputs. The purpose of the VRM is to provide regulated power for the robot radio, custom circuits, and IP vision cameras.

**Note: The two connector pairs associated with each label have a combined rating of what the label indicates (e.g. 5V/500mA total for both pairs not for each pair). The 12V/2A limit is a peak rating, the supply should not be loaded with more than 1.5A continuous current draw.** For more information, see the [VRM User Manual](#).

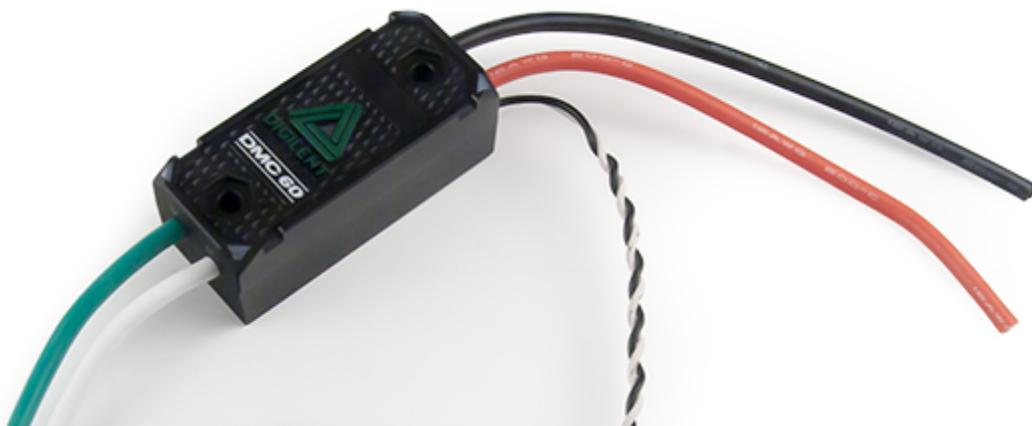
## Motor Controllers

There are a variety of different motor controllers which work with the FRC Control System and are approved for use. These devices are used to provide variable voltage control of the brushed DC motors used in FRC. They are listed here in alphabetical order.

# Getting Started with the Control System

## DMC-60 and DMC-60C Motor Controller

---



The DMC-60 is a PWM motor controller from Digilent. The DMC-60 features integrated thermal sensing and protection including current-foldback to prevent overheating and damage, and four multi-color LEDs to indicate speed, direction, and status for easier debugging. For more information, see the DMC-60 reference manual: <https://reference.digilentinc.com/dmc-60/reference-manual>

The DMC-60C adds CAN smart controller capabilities to the DMC-60 controller. This enables closed loop control features and other intelligent control options. For more information see the DMC-60C Product Page: <https://store.digilentinc.com/dmc60c-digital-motor-controller-approved-for-first-robotics/>

# Getting Started with the Control System

## Jaguar Motor Controller



The Jaguar Motor Controller from VEX Robotics (formerly made by Luminary Micro and Texas Instruments) is a variable speed motor controller for use in FRC. For FRC, the Jaguar may only be controlled using the PWM interface. For more information, see the Jaguar Getting Started Guide, Jaguar Datasheet and Jaguar FAQ on [this page](#).

# Getting Started with the Control System

## SD540B and SD540C Motor Controllers



The SD540 Motor Controller from Mindsensors is a variable speed motor controller for use in FRC. The SD540B is controlled using the PWM interface. The SD540C is controllable over CAN. Limit switches may be wired directly to the SD540 to limit motor travel in one or both directions. Switches on the device are used to flip the direction of motor travel, configure the wiring polarity of limit switches, set Brake or Coast mode, and put the device in calibration mode. For more information see the Mindsensors FRC page: <http://www.mindsensors.com/68-frc>

# Getting Started with the Control System

## SPARK Motor Controller



The SPARK Motor Controller from REV Robotics is a variable speed motor controller for use in FRC. The SPARK is controlled using the PWM interface. Limit switches may be wired directly to the SPARK to limit motor travel in one or both directions. The RGB status LED displays the current state of the device including whether the device is currently in Brake mode or Coast mode. For more information, see the REV Robotics SPARK product page: <http://www.revrobotics.com/rev-11-1200/>

# Getting Started with the Control System

## SPARK MAX Motor Controller



The SPARK MAX Motor Controller from REV Robotics is a variable speed motor controller for use in FRC. The SPARK MAX is capable of controlling either the traditional brushed DC motors commonly used in FRC or the new brushless REV Robotics NEO Brushless Motor. The SPARK MAX can be controlled over PWM, CAN or USB (for configuration/testing only). The controller has a data port for sensor input and is capable of closed loop control modes when controlled over CAN or USB. For more information see the REV Robotics SPARK MAX product page: <http://www.revrobotics.com/rev-11-2158/>

# Getting Started with the Control System

## Talon Motor Controller



The Talon Motor Controller from Cross the Road Electronics is a variable speed motor controller for use in FRC. The Talon is controlled over the PWM interface. The Talon should be connected to a PWM output of the roboRIO and powered from the Power Distribution Panel. For more information see the [Talon User Manual](#).

# Getting Started with the Control System

## Talon SRX



The Talon SRX motor controller is a CAN-enabled "smart motor controller" from Cross The Road Electronics/VEX Robotics. The Talon SRX has an electrically isolated metal housing for heat dissipation, making the use of a fan optional. The Talon SRX can be controlled over the CAN bus or PWM interface. When using the CAN bus control, this device can take inputs from limit switches and potentiometers, encoders, or similar sensors in order to perform advanced control such as limiting or PID(F) closed loop control on the device. For more information see the [Talon SRX User Manual](#).

**Note:** CAN Talon SRX has been removed from WPILib. See [this blog](#) for more info and find the CTRE Toolsuite installer here: [http://www.ctr-electronics.com/control-system/hro.html#product\\_tabs\\_technical\\_resources](http://www.ctr-electronics.com/control-system/hro.html#product_tabs_technical_resources)

# Getting Started with the Control System

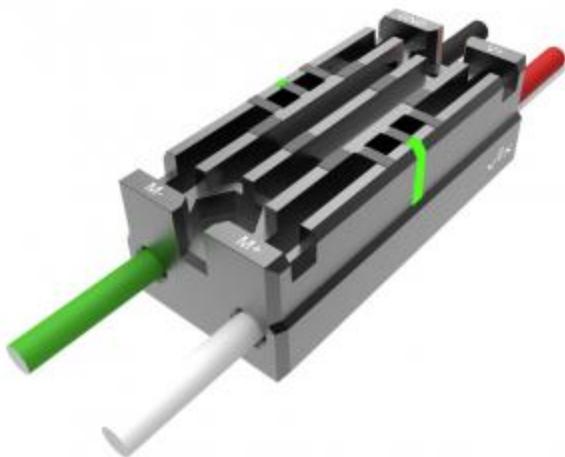
## Victor 888 Motor Controller / Victor 884 Motor Controller



The Victor 888 Motor Controller from VEX Robotics is a variable speed motor controller for use in FRC. The Victor 888 replaces the Victor 884, which is also usable in FRC. The Victor is controlled over the PWM interface. The Victor should be connected to a PWM output of the roboRIO and powered from the Power Distribution Panel. For more information, see the [Victor 884 User Manual](#) and [Victor 888 User Manual](#).

# Getting Started with the Control System

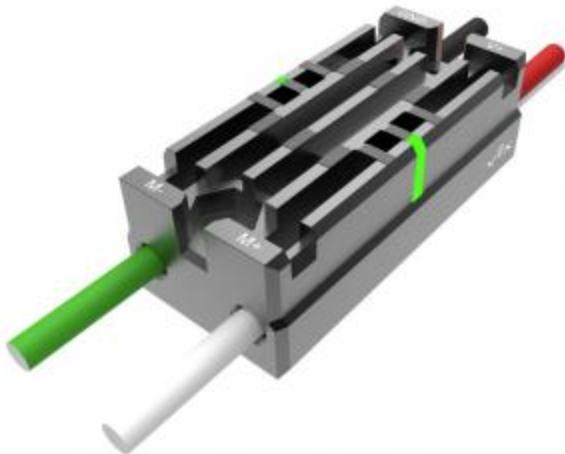
## Victor SP



The Victor SP motor controller is a PWM motor controller from Cross The Road Electronics/VEX Robotics. The Victor SP has an electrically isolated metal housing for heat dissipation, making the use of the fan optional. The case is sealed to prevent debris from entering the controller. The controller is approximately half the size of previous models. For more information, see the [Victor SP User Manual](#).

# Getting Started with the Control System

## Victor SPX



The Victor SPX motor controller is a CAN or PWM controlled motor controller from Cross The Road Electronics/VEX Robotics. The device is connectorized to allow easy connection to the roboRIO PWM connectors or a CAN bus chain. When controlled over the CAN bus, the device has a number of the closed loop features also present in the Talon SRX. The case is sealed to prevent debris from entering the controller. For more information, see the [Victor SPX Webpage](#).

**Note:** Victor SPX CAN control is not supported from WPILib. See [this blog](#) for more info and find the CTRE Toolsuite installer here: [http://www.ctr-electronics.com/control-system/hro.html#product\\_tabs\\_technical\\_resources](http://www.ctr-electronics.com/control-system/hro.html#product_tabs_technical_resources)

# Getting Started with the Control System

## Spike H-Bridge Relay



The Spike H-Bridge Relay from VEX Robotics is a device used for controlling power to motors or other custom robot electronics. When connected to a motor, the Spike provides On/Off control in both the forward and reverse directions. The Spike outputs are independently controlled so it can also be used to provide power to up to 2 custom electronic circuits. The Spike H-Bridge Relay should be connected to a relay output of the roboRIO and powered from the Power Distribution Panel. For more information, see the [Spike User's Guide](#).

# Getting Started with the Control System

## Servo Power Module



The Servo Power Module from Rev Robotics is capable of expanding the power available to servos beyond what the roboRIO integrated power supply is capable of. The Servo Power Module provides up to 90W of 6V power across 6 channels. All control signals are passed through directly from the roboRIO. For more information, see the [Servo Power Module webpage](#).

# Getting Started with the Control System

## Axis M1013/M1011/206 Ethernet Camera



The Axis M1013, M1011 and Axis 206 Ethernet cameras are used for capturing images for vision processing and/or sending video back to the Driver Station laptop. The camera should be wired to a 5V power output on the Voltage Regulator Module and an open ethernet port on the robot radio. For more information, see [Configuring an Axis Camera](#) and the [Axis 206](#), [Axis M1011](#), [Axis M1013](#) pages.

# Getting Started with the Control System

## Microsoft Lifecam HD3000



The Microsoft Lifecam HD3000 is a USB webcam that can be plugged directly into the roboRIO. The camera is capable of capturing up to 1280x720 video at 30 FPS. For more information about the camera, see the [Microsoft product page](#). For more information about using the camera with the roboRIO, see the [Vision Processing section](#) if this documentation.

## OpenMesh OM5P-AN or OM5P-AC Radio



# Getting Started with the Control System

Either the OpenMesh OM5P-AN or OpenMesh OM5P-AC wireless radio is used as the robot radio to provide wireless communication functionality to the robot. The device can be configured as an Access Point for direct connection of a laptop for use at home. It can also be configured as a bridge for use on the field. The robot radio should be powered by one of the 12V/2A outputs on the VRM and connected to the roboRIO controller over Ethernet. For more information, see [Programming your radio for home use](#) and the [Open Mesh OM5P-AC product page](#).

The OM5P-AN is [no longer available for purchase](#). The OM5P-AC is slightly heavier, has more cooling grates, and has a rough surface texture compared to the OM5P-AN.

## 120A Circuit Breaker



The 120A Main Circuit Breaker serves two roles on the robot: the main robot power switch and a protection device for downstream robot wiring and components. The 120A circuit breaker is wired to the positive terminals of the robot battery and Power Distribution boards. For more information, please see the [Cooper Bussmann 18X Series Datasheet](#) (PN: 185120F)

# Getting Started with the Control System

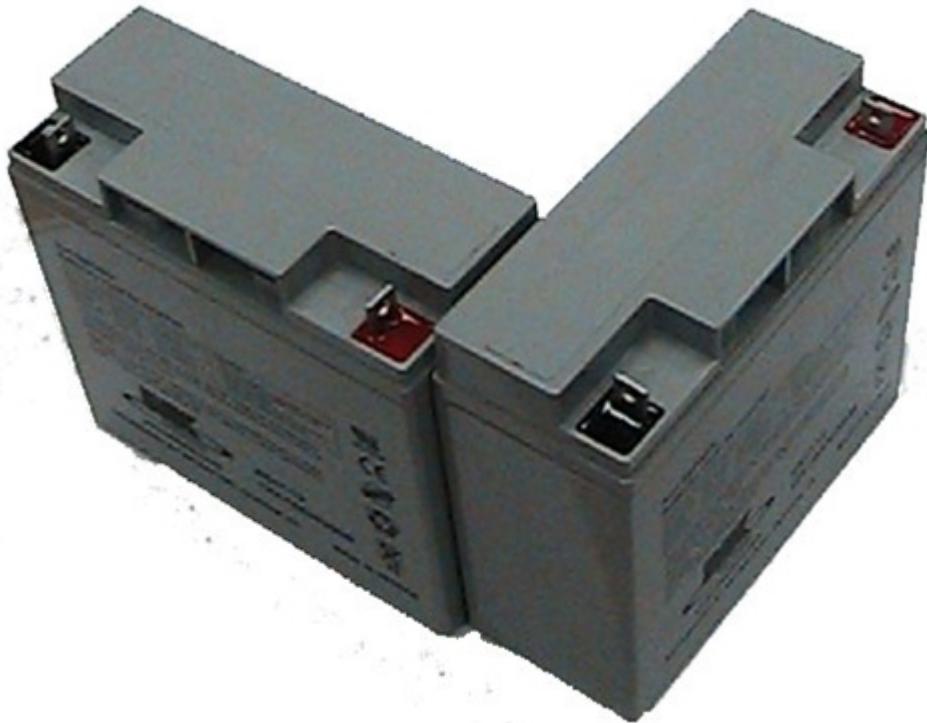
## Snap Action Circuit Breakers



The Snap Action circuit breakers, MX5-A40 and VB3 series, are used with the Power Distribution Panel to limit current to branch circuits. The MX5-A40 40A MAXI style circuit breaker is used with the larger channels on the Power Distribution Panel to power loads which draw current up to 40A continuous. The VB3 series are used with the smaller channels on the PDP to power circuits drawing current of 30A or less continuous. For more information, see the Datasheets for the [MX5 series](#) and [VB3 Series](#).

# Getting Started with the Control System

## Robot Battery



The power supply for an FRC robot is a single 12V 18Ah battery. The batteries used for FRC are sealed lead acid batteries capable of meeting the high current demands of an FRC robot. For more information, see the Datasheets for the [MK ES17-12](#) and [Enersys NP18-12](#). Note that other battery part numbers may be legal, consult the 2015 FRC Manual for a complete list.

## Image credits

Image of roboRIO courtesy of [National Instruments](#). Image of DMC-60 courtesy of [Digilent](#). Image of SD540 courtesy of [Mindsensors](#). Images of Jaguar Motor Controller, Talon SRX, Victor 888, Victor SP, Victor SPX, and Spike H-Bridge Relay courtesy of [VEX Robotics, Inc](#). Image of SPARK MAX courtesy of [REV Robotics](#). Lifecam, PDP, PCM, SPARK, and VRM photos courtesy of FIRST. All other photos courtesy of [AndyMark Inc.](#).

# FRC Software Component Overview

The FRC control system consists of a wide variety of mandatory and optional software components. These elements are designed to assist you in the design, development, and debugging of your robot code as well as assist with control robot operation and to provide feedback when troubleshooting. For each software component this document will provide a brief overview of its purpose, a link to the package download, if appropriate, and a link to further documentation where available.

## Operating System Compatibility

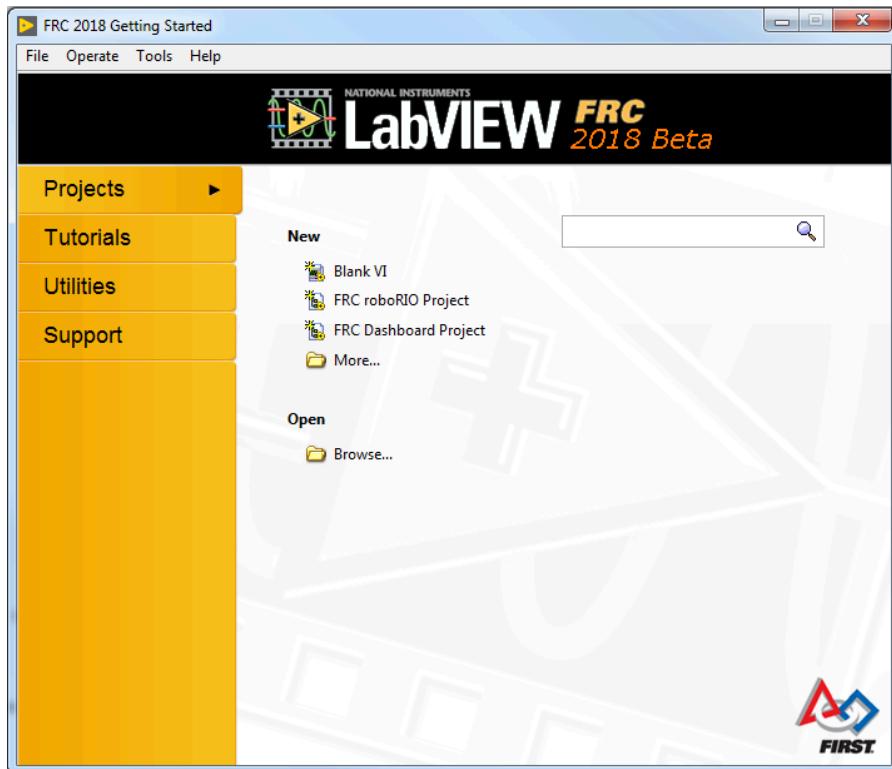
The primary supported OS for FRC components is Windows. All required FRC software components have been tested on Windows 7, 8, and 10. Windows XP is not supported.

Having said that, many of the tools for C++/Java programming are also supported and tested on Mac and Linux. Teams programming in C++/Java should be able to develop using these systems, using a Windows system for the Windows-only operations such as Driver Station, radio programming, and roboRIO imaging.

Components supported on all OS's have been marked with an \* below. All other items are Windows only, unless noted.

# Getting Started with the Control System

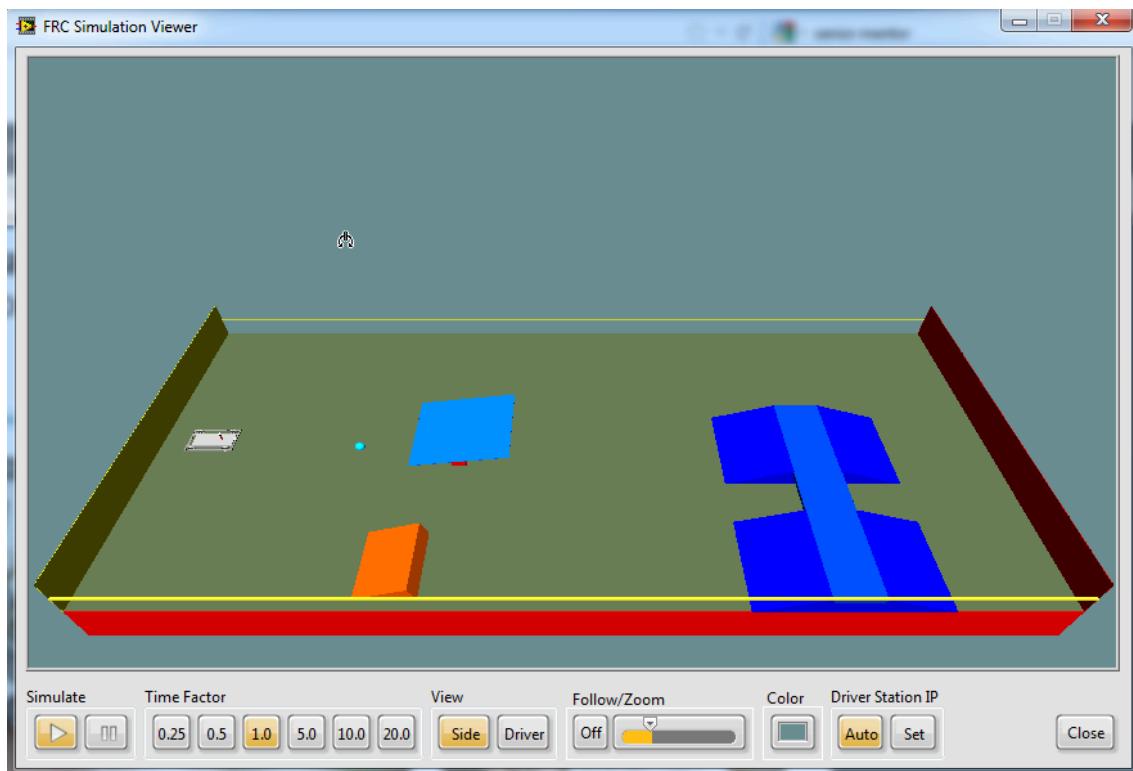
## LabVIEW FRC



LabVIEW FRC, based on National Instruments' LabVIEW 2018, is the development environment for LabVIEW, one of the three officially supported languages for programming an FRC robot. LabVIEW is a graphical, dataflow-driven language. LabVIEW programs consist of a collection of icons, called VIs, wired together with wires which pass data between the VIs. The LabVIEW FRC installer is distributed on a DVD found in the Kickoff Kit of Parts and is also available for download (see installation instructions page linked below). Instructions for installing the FRC libraries (package also includes Driver Station and Utilities) can be found [here](#). A guide to getting started with the LabVIEW FRC software, including installation instructions can be found [here](#).

# Getting Started with the Control System

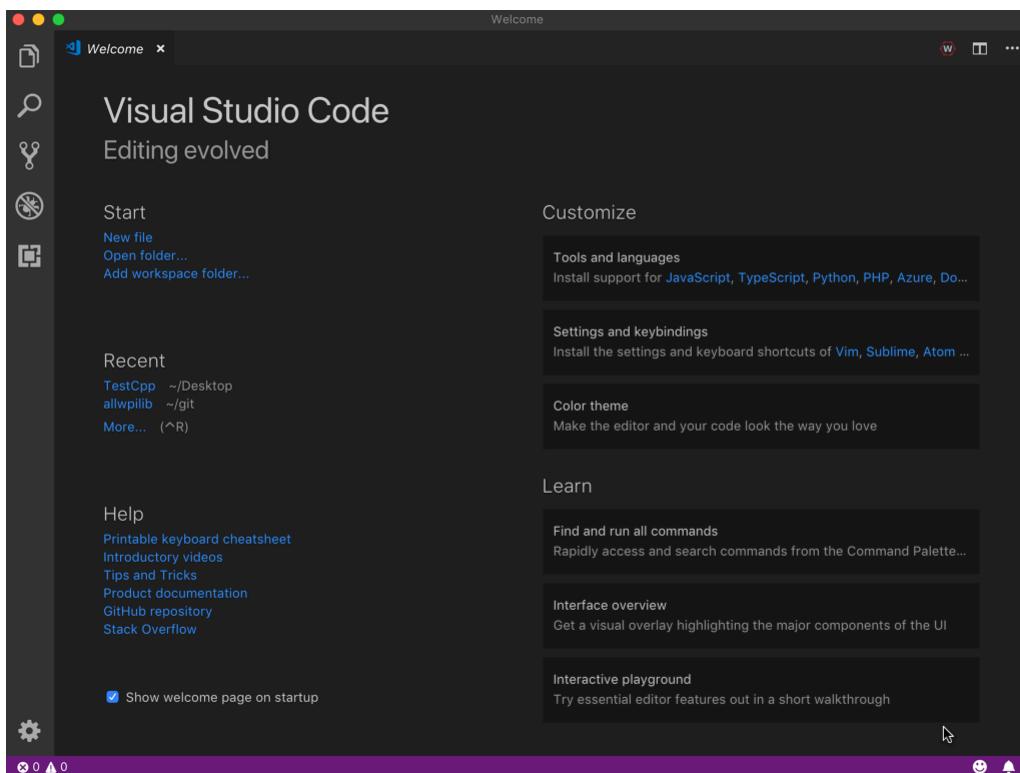
## FRC Robot Simulator



The FRC Robot Simulator is a component of the LabVIEW programming environment that allows you to operate a predefined robot in a simulated environment to test code and/or Driver Station functions. It utilizes a LabVIEW code project as the robot code and communicates with the FRC Driver Station for robot control and the FRC Default Dashboard for robot feedback. The FRC Robot Simulator is installed with the LabVIEW FRC package. Information on using the FRC Robot Simulator can be found by opening the Robot Simulation Readme.html file in the LabVIEW Project Explorer.

# Getting Started with the Control System

## Visual Studio Code\*



Visual Studio Code is the supported development environment for C++ and Java, two of the three supported languages used for programming an FRC robot. Both are object-oriented text based programming languages. A program in C++ (for FRC) consists of a number of header (.h) and implementation (.cpp) files whereas a program in Java consists of .java files contained in one or more packages. A guide to getting started with C++ for FRC, including the installation and configuration of Visual Studio Code can be found [here](#). A guide to getting started with Java for FRC, including the installation and configuration of the Visual Studio Code can be found [here](#).

# Getting Started with the Control System

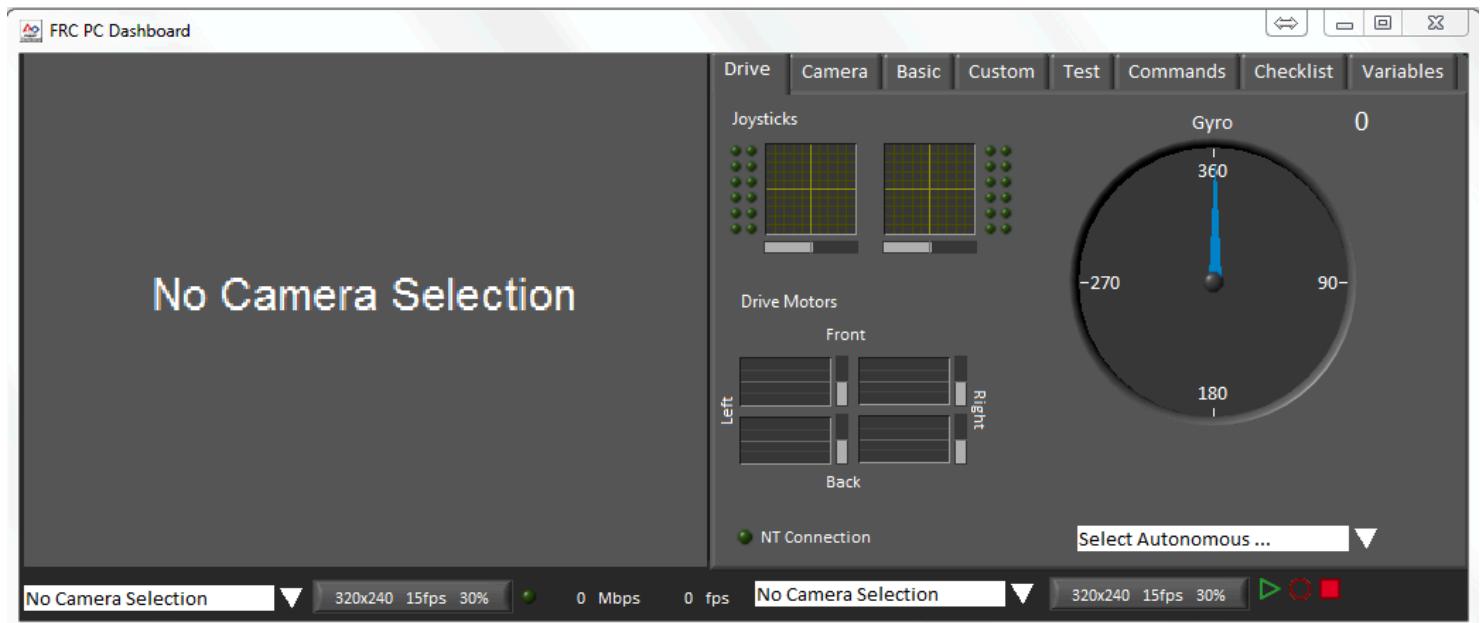
## FRC Driver Station Powered by NI LabVIEW



The FRC Driver Station Powered by NI LabVIEW is the only software allowed to be used for the purpose of controlling the state of the robot during competition. This software contains the code necessary to send data to your robot from a variety of input devices such as joysticks, gamepads, and customizable IO boards. It also contains a number of tools used to help troubleshoot robot issues such as status indicators and log file creation. Instructions for installing the FRC Driver Station Powered by NI LabVIEW (included in the FRC Update Suite) can be found [here](#), More information about the FRC Driver Station Powered by NI LabVIEW can be found [here](#).

# Getting Started with the Control System

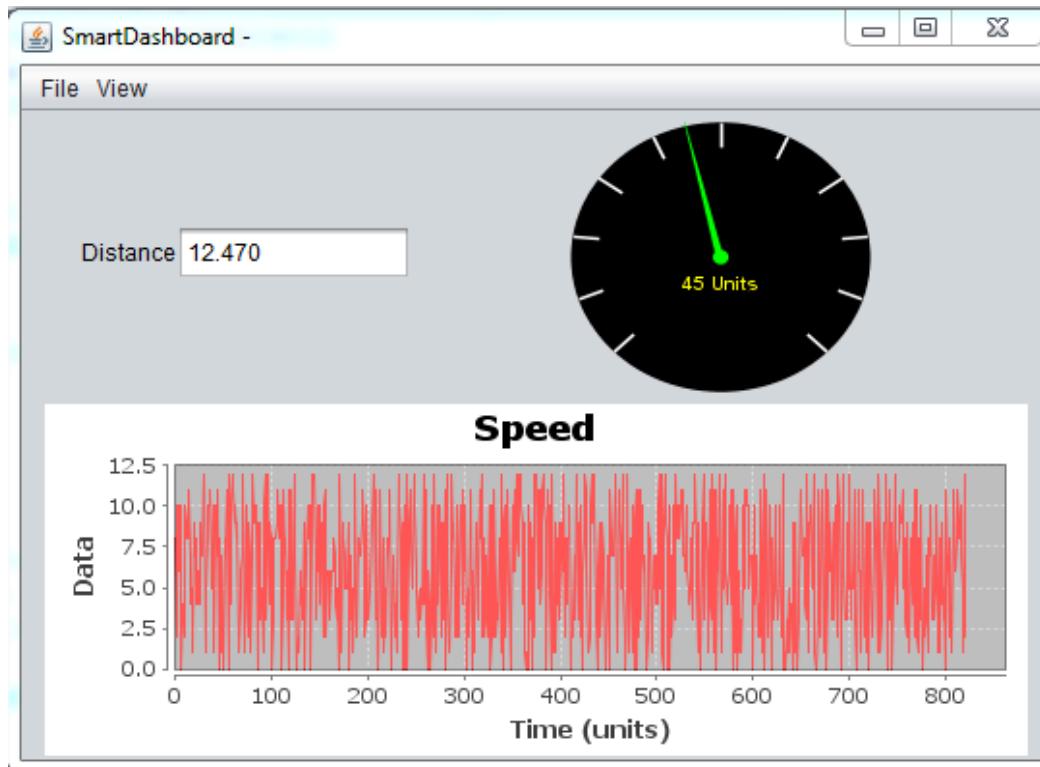
## FRC LabVIEW Dashboard



The FRC LabVIEW Dashboard is the default dashboard program installed with, and automatically launched by, the FRC Driver Station. The purpose of the Dashboard is to provide feedback about the operation of the robot. The FRC Default Dashboard serves as a an example of the types of feedback teams may want from their robot. It includes a tabbed display that can switch between viewing an image from a camera on the robot or a display of NetworkTables variables, a display of information regarding the joysticks and drive motors, an indicator of the robot IP and battery voltage, and a second tabbed display that can switch between examples of custom indicators and controls, a test tab for use with the Driver Station Test Mode and a Checklist tab that teams can use to enter a custom checklist to complete before each match. The FRC Default Dashboard is included in the FRC Update Suite. Installation instructions can be found [here](#). More information about the FRC Default Dashboard software can be found [here](#).

# Getting Started with the Control System

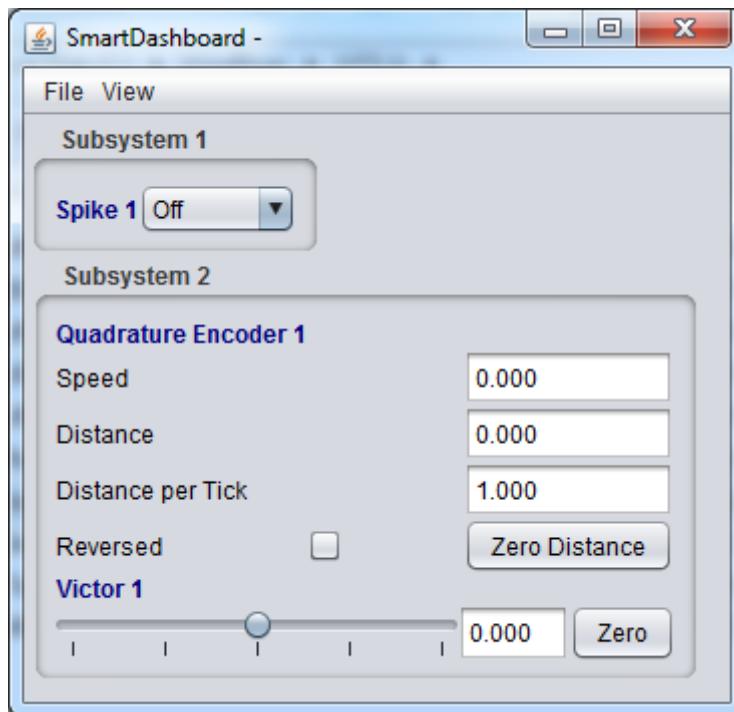
## SmartDashboard\*



The SmartDashboard is an alternate dashboard application written in Java. The SmartDashboard automatically creates a widget for each variable sent from the Robot sent using the SmartDashboard class or VIs. These widgets can be configured to a number of preset display types, or users can create custom extensions in Java. Vision extensions are available for the SmartDashboard which allow it to display images from the Axis camera on the robot. The SmartDashboard is included in the [C++ and Java](#) language updates (enabled by clicking the C++ or Java buttons respectively on the Setup tab of the Driver Station). Additional documentation on the SmartDashboard can be found [here](#).

# Getting Started with the Control System

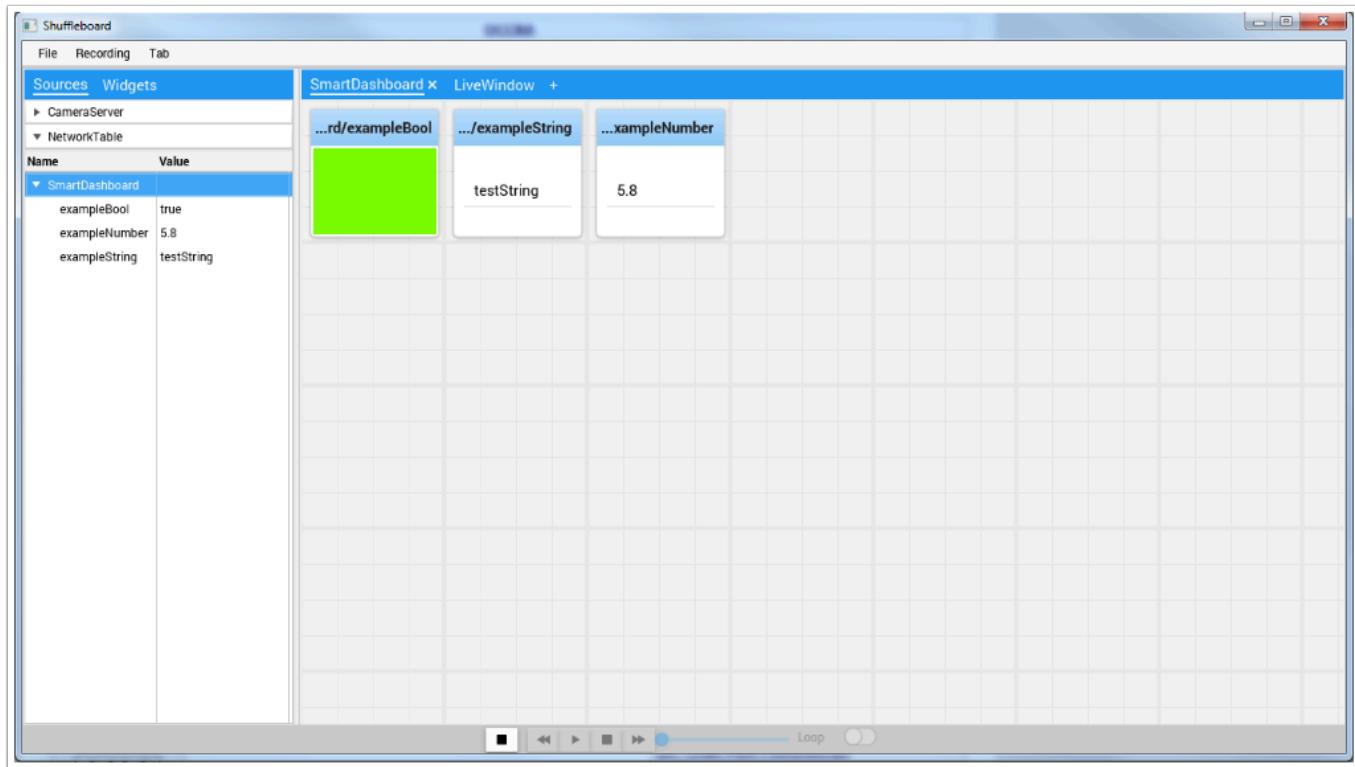
## LiveWindow\*



LiveWindow is a new mode of the SmartDashboard for 2013, designed for use with the new Test Mode of the Driver Station. LiveWindow allows the user to see feedback from sensors on the robot and control actuators independent of the written user code. More information about LiveWindow can be found [here](#).

# Getting Started with the Control System

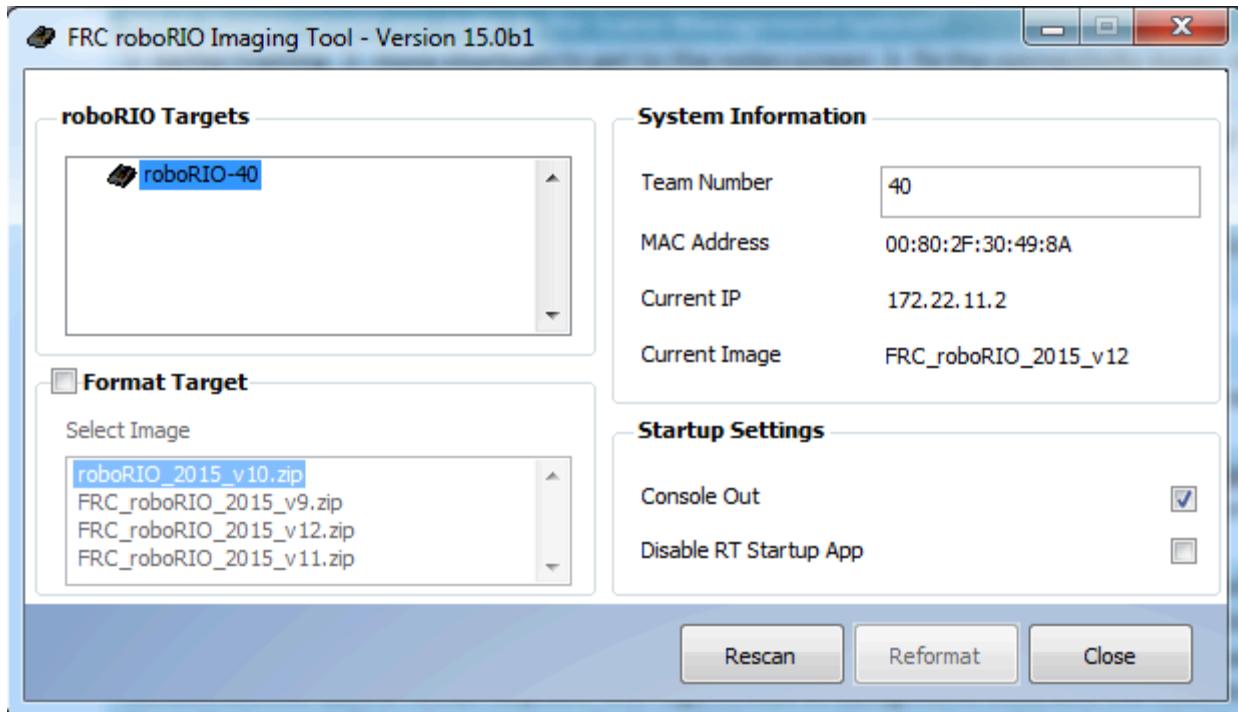
## Shuffleboard\*



Shuffleboard is an alternative dashboard application written in Java. It takes many of the concepts from SmartDashboard such as automatic adding of widgets and adds new features including better layout control and record/playback functionality. Shuffleboard contains all of the basic widget types found in the SmartDashboard as well as a number of new ones intended to make visualizing specific robot components even easier. It has full integration with WPILib's "cscore" for displaying, recording, and playing back camera streams. Shuffleboard is included in the [C++ and Java](#) language updates (enabled by clicking the Shuffleboard button on the Setup tab of the Driver Station or by launching it from the WPILib menu in Eclipse). Additional documentation on Shuffleboard can be found [here](#).

# Getting Started with the Control System

## FRC roboRIO Imaging Tool



The FRC roboRIO Imaging Tool is a software tool used to format and setup an roboRIO-FRC device for use in FRC. The tool detects any roboRIO device on the network, reports the current MAC, name, IP and Image version. The tool allows the user to configure the team number, set options including Console Out and whether an applications runs on Startup, and install the latest software image on the device. The FRC roboRIO Imaging Tool is installed as part of the FRC Update Suite. Installation instructions can be found [here](#). Additional instructions on imaging your roboRIO using this tool can be found [here](#).

# Getting Started with the Control System

## CTRE Toolsuite

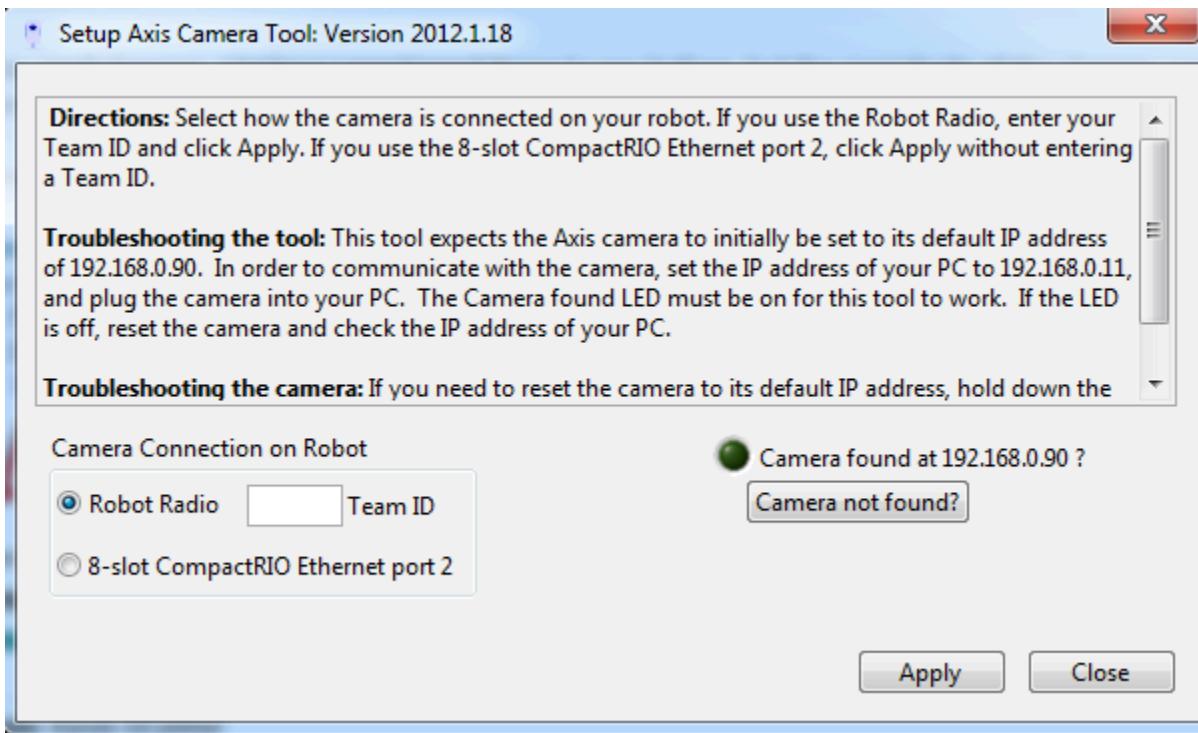


The CTRE Toolsuite installs the software libraries for Talon SRX (C++\Java\LabVIEW) as well as the HERO Lifeboat software which can be used to update the roboRIO web based CAN configuration with the latest CTRE-specific features. The installer can be found here: <http://www.ctre-electronics.com/control-system/hro.html>

Note on non-Windows: A separate package (zip) is provided to get the Talon SRX and Pigeon libraries on non-Windows systems. Users should unzip this file and place the contents into USER\wpilib\user folder

# Getting Started with the Control System

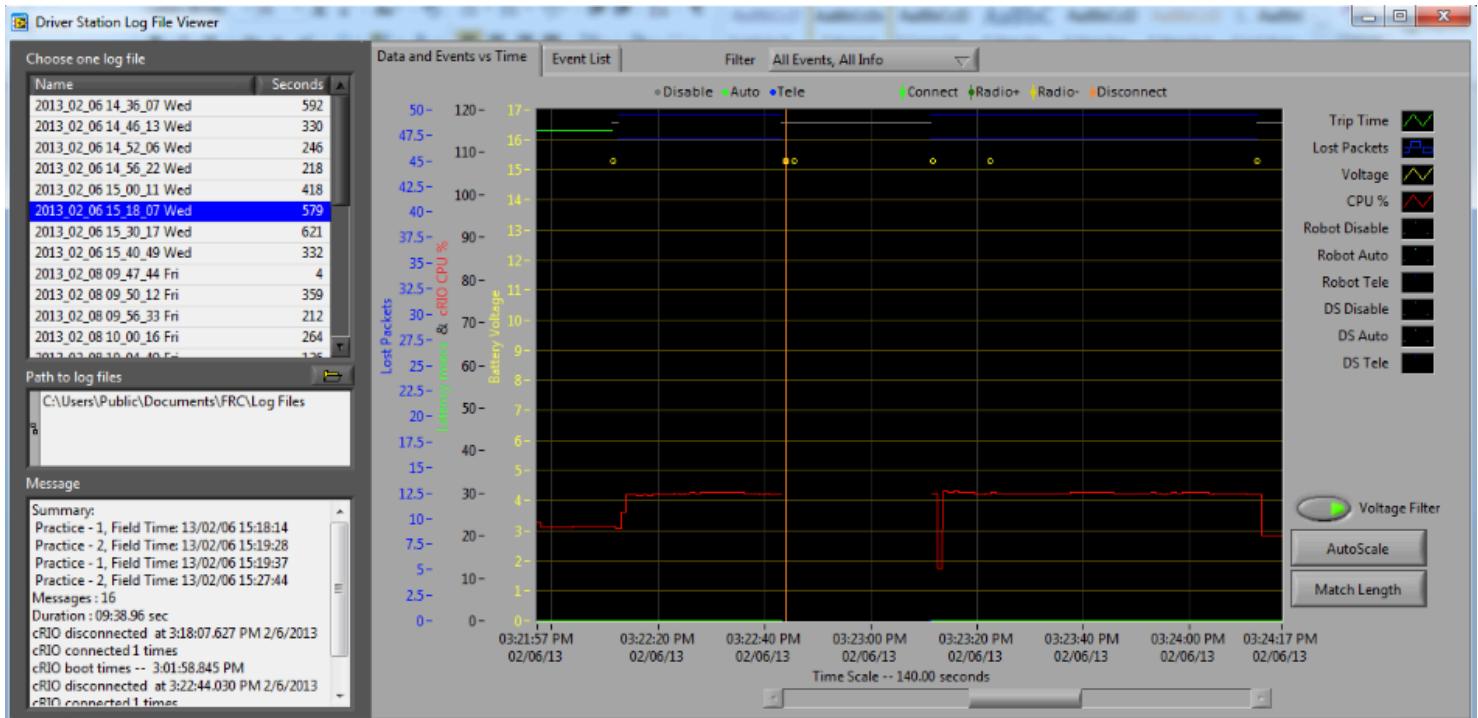
## Setup Axis Camera



The Setup Axis Camera utility is a LabVIEW program used to configure an Axis 206, M1011 or M1013 camera for use on the robot. The tool takes a factory reset camera connected directly to the computer and configures the IP, username and password, anonymous access, and default framerate and compression (for use with the SmartDashboard or other access methods). The Setup Axis Camera tool is installed as part of the FRC Update Suite. Installation instructions can be found [here](#). Instructions for using the tool to configure the camera are located [here](#).

# Getting Started with the Control System

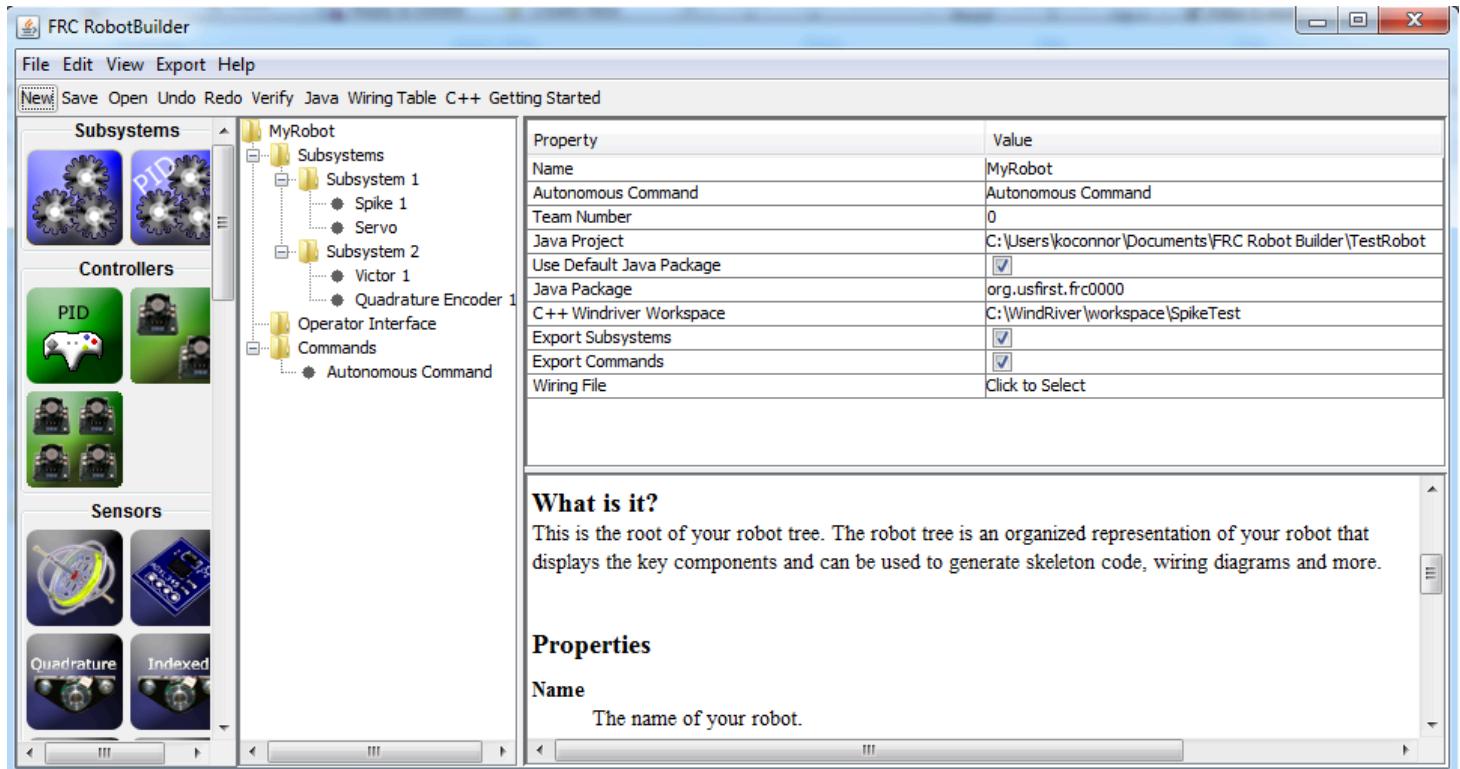
## FRC Driver Station Log Viewer



The FRC Driver Station Log Viewer is a LabVIEW program used to view logs created by the FRC Driver Station. These logs contain information such as battery voltage, trip time, CPU% and robot mode, as well as events such as joystick removal. The FRC Driver Station Log Viewer is included in the FRC Update Suite. Installation instructions can be found [here](#). More information about the FRC Driver Station Log Viewer and understanding the logs can be found [here](#).

# Getting Started with the Control System

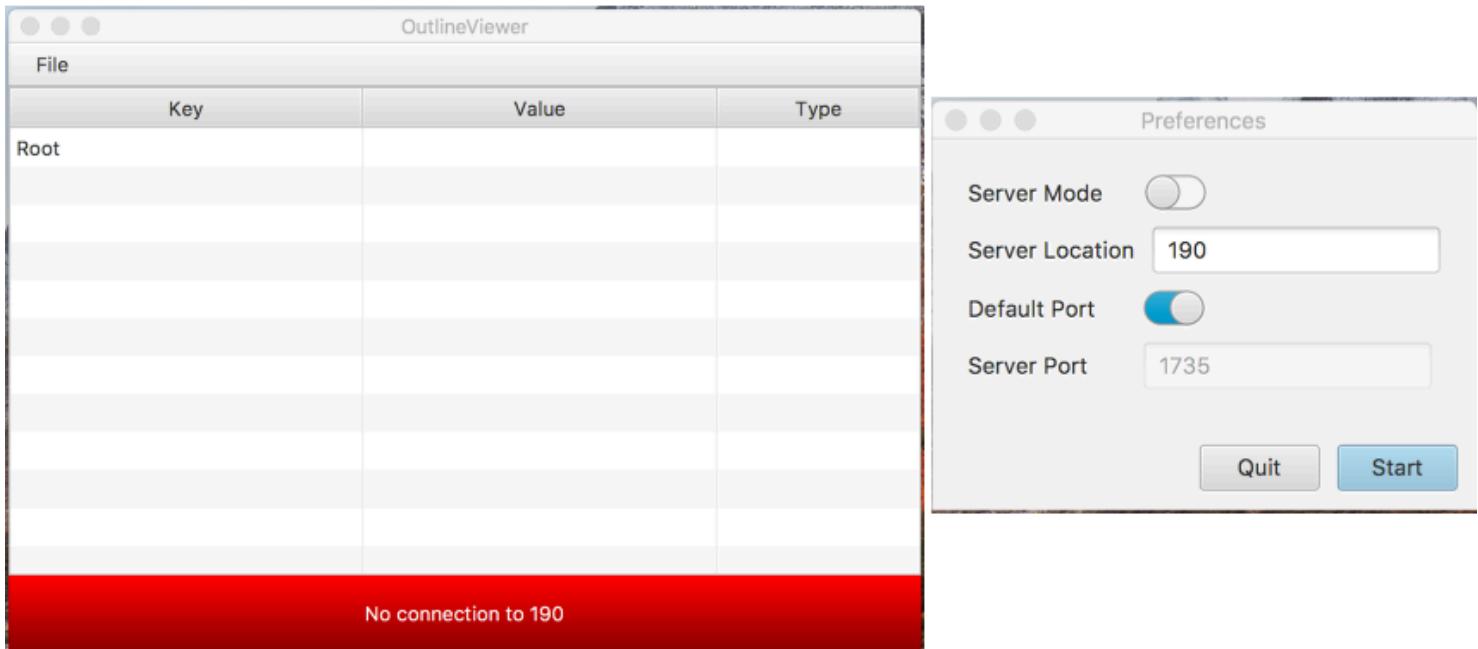
## RobotBuilder\*



RobotBuilder is a tool designed to aid in setup and structuring of a Command Based robot project for C++ or Java. RobotBuilder allows you to enter in the various components of your robot subsystems and operator interface and define what your commands are in a graphical tree structure. RobotBuilder will then verify that you have no port allocation conflicts and can generate a wiring table indicating what is connected to each port as well as C++ or Java code. The code created generates the appropriate files, constructs the appropriate objects and adds LiveWindow code for each sensor and actuator, but does not write any of the actual Subsystem or Command methods. The user must write the appropriate code for these methods for the robot to function. More information about Robot Builder can be found [here](#). More information about the Command Based programming architecture can be found in the [C++](#) and [Java](#) manuals.

# Getting Started with the Control System

## OutlineViewer\*



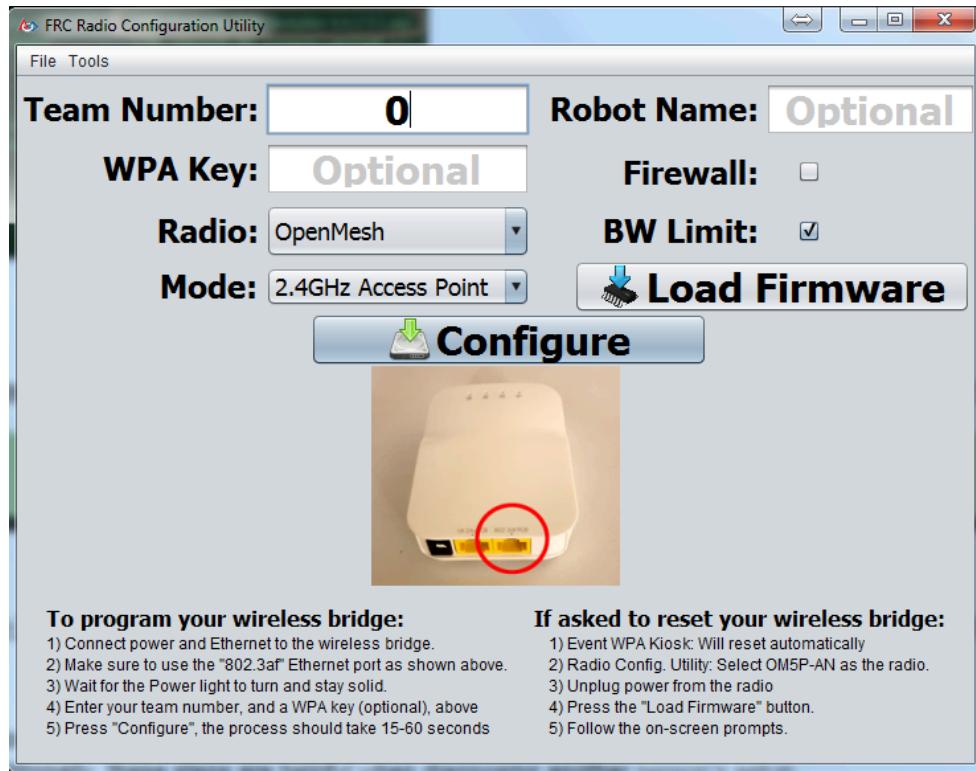
OutlineViewer is a utility used to view, modify and add to the contents of the NetworkTables for debugging purposes. It displays all key value pairs currently in the NetworkTables and can be used to modify the value of existing keys or add new keys to the table. OutlineViewer is included in the [C++ and Java](#) language updates (found in USER\tools\wpilib). Teams may need to install the Java Runtime Environment to use the OutlineViewer on computers not set up for Java programming.

To connect to your robot, open OutlineViewer and set the "Server Location" to be your team number. After you click start, OutlineViewer will connect.

LabVIEW teams can use the Variables tab of the LabVIEW Dashboard to accomplish this functionality.

# Getting Started with the Control System

## FRC Radio Configuration Utility



The FRC Bridge Configuration Utility is a tool used to configure the the OpenMesh OM5P-AN or OM5P-AC radio for practice use at home. This tool sets the appropriate IP, and network settings for proper network connection, as well as the QOS settings required to mimic the bandwidth limiting and packet prioritization experience on the FRC playing field. The FRC Bridge Configuration Utility is installed by a standalone installer, instructions on installing and using the FRC Bridge Configuration Utility to configure your radio can be found [here](#).

# Getting Started with the Control System

## 3rd Party Libraries

A number of software components were broken out of WPILib starting in 2017 and are now maintained by the third parties who produce the hardware. See [this blog](#) for more details.

### Libraries

**⚠ Note:** These are not links to directly plug in to the VS Code -> Install New Libraries (online) feature. Click these links to visit the vendor site to see whether they offer online installers, offline installers, or both.

Most vendors have not yet published final releases for 2019, keep an eye on the vendor pages for 2019 software releases.

[Analog Devices ADIS16448 IMU](#) - Driver for ADIS16448 IMU. More info on <https://wiki.analog.com/first>

[Analog Devices ADIS16470 IMU](#) - Driver for ADIS16470 IMU. More info on <https://wiki.analog.com/first>

[CTRE Phoenix Toolsuite](#) - Contains TalonSRX/Victor SPX Libraries and Phoenix Tuner program for configuring CTRE CAN devices

[Digilent](#) - DMC-60C library

[Kauai Labs](#) - Libraries for NavX-MXP, NavX-Micro, and Sensor Fusion

[Mindsensors Libraries](#) - Contains libraries for SD540C and CANLight

[Rev Robotics](#) - SPARK MAX Library

[Scanse Sweep](#) - C++\Java Libraries for Scansense Sweep LIDAR (packaged by Peter Johnson)

# Getting Started with the Control System

## The Mechanism

In support of this effort NI (for LabVIEW) and FIRST\WPI (for C++\Java) have developed mechanisms that should make it easy for vendors to plug their code into the WPILib software and for teams to use that code once it has been installed. A brief description of how the system works for each language can be found below.

### The Mechanism - LabVIEW

For LabVIEW teams, you may notice a few new Third Party items on various palettes (specifically, one in Actuators, one in Actuators->Motor Control labeled "CAN Motor", and one in "Sensors"). These correspond to folders in Program Files\National Instruments\LabVIEW 2016\vi.lib\Rock Robotics\WPILib\Third Party. For a library to insert VI's in these palettes, they simply make a subfolder in one of these three Third Party folders containing their VI's and they will be added automatically. To control the appearance of the palette (have some VI's not show up, set the Icon for the folder, etc.) there is a process to create a dir.mnu file for your directory. We will be working on documenting that process shortly.

To use installed Third Party libraries, simply locate the VI's in one of these 3 locations and drag them into your project as you would with any other VI.

### The Mechanism - C++\Java

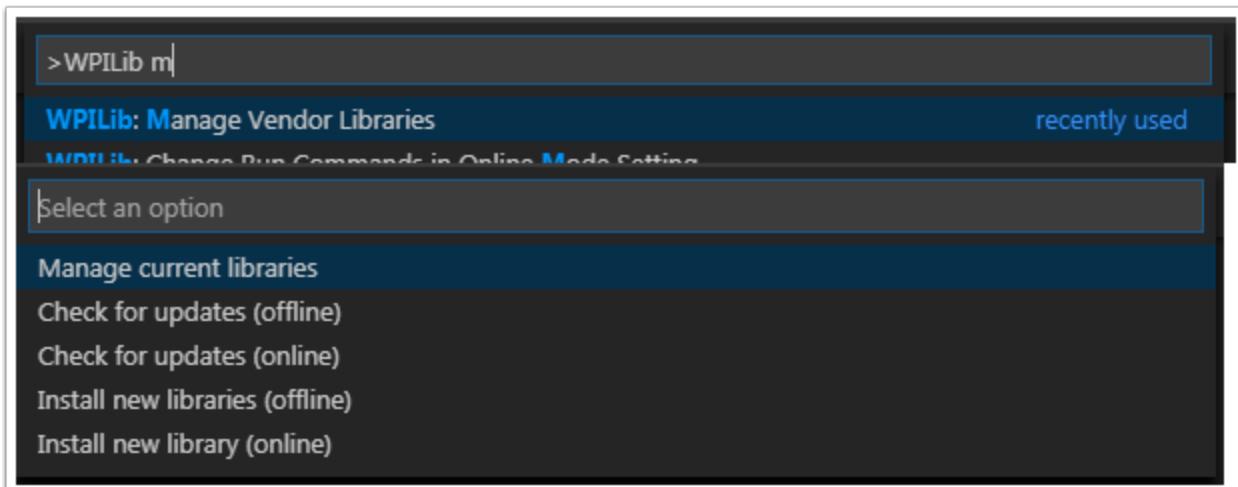
For C++ and Java a JSON file describing the vendor library is installed on your system to ~home\frcYYYY\vendordeps (~home = C:\Users\Public on Windows). This can either be done by an offline installer or the file can be fetched from an online location using the menu item in VSCode. This file is then used from VS Code to add to the library to each individual project. Vendor library information is managed on a per-project basis to make sure that a project is always pointing to a consistent version of a given vendor library. The libraries themselves are placed in the Maven cache at C:\Users\Public\frcYYYY\maven. Vendors can place a local copy here with an offline installer (recommended) or require users to be online for an initial build to fetch the library from a remote Maven location.

The JSON file allows specification of complex libraries with multiple components (C++, Java, JNI, etc.) and also helps handle some complexities related to simulation. Vendors choosing to provide a remote URL in the JSON also enable users to check for updates from within VS Code.

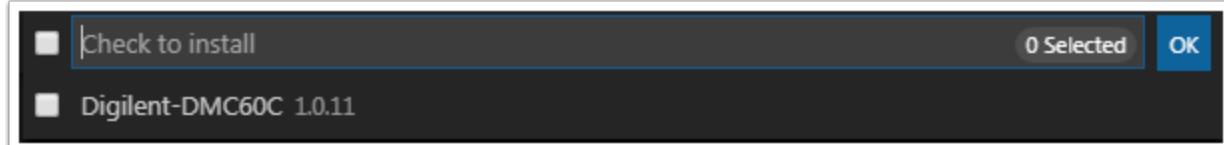
# Getting Started with the Control System

- 💡 Note:** The vendor JSON files are actually processed by GradleRIO once they are in your project's vendordeps folder. If you are using another IDE, you will need to manually create a "vendordeps" folder in your project and copy any desired vendor JSON files from the "frcYYYY" folder (where they should be placed by an offline installer) or download them directly from the vendor and place them into the folder in the project.

## Adding an offline-installed Library



To add a vendor library that has been installed by an offline installer, press Ctrl+Shift+P and type WPILib or click on the WPILib icon in the top right to open the WPILib Command Palette and begin typing **Manage Vendor Libraries**, then select it from the menu. Select the option to **Install new libraries (offline)**.



# Getting Started with the Control System

Select the desired libraries to add to the project by checking the box next to each, then click OK. The JSON file will be copied to the **vendordeps** folder in the project, adding the library as a dependency to the project.

## Checking for Updates (offline)

Remember: Dependencies are now version managed and done on a per-project bases. Even if you have installed an updated library using an offline installer, you will need to **Manage Vendor Libraries** and select **Check for updates (offline)** for each project you wish to update.

## Checking for Updates (online)

Part of the JSON file that vendors may optionally populate is an online update location. If a library has an appropriate location specified, running **Check for updates (online)** will check if a newer version of the library is available from the remote location.

## Removing a library dependency

To remove a library dependency from a project, select **Manage Current Libraries** from the **Manage Vendor Libraries** menu, check the box for any libraries to uninstall and click OK. These libraries will be removed as dependencies from the project.

Getting Started with the Control System

# Getting Started: Step-By-Step

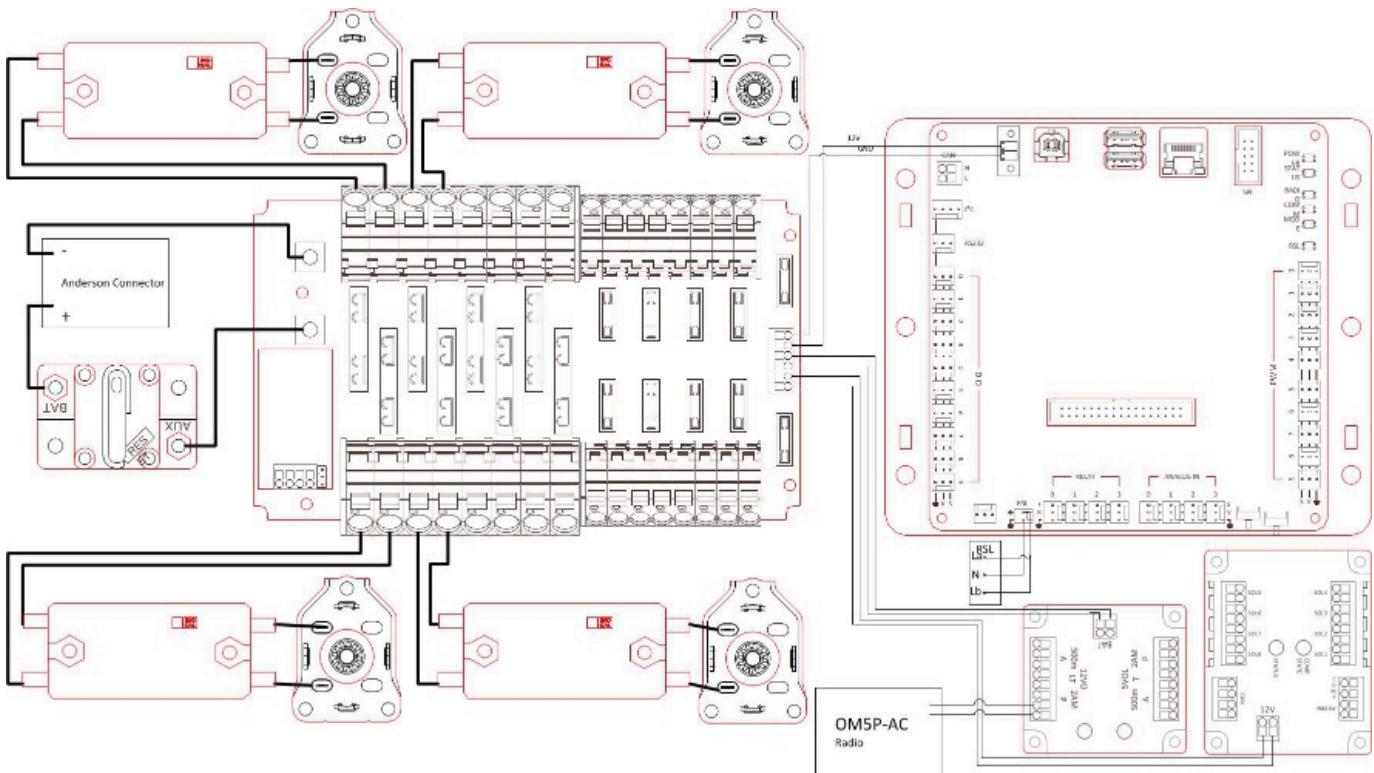
# Getting Started with the Control System

## Wiring the FRC Control System

This document details the wiring of a basic electronics board for bench-top testing.

Some images shown in this section reflect the setup for a Robot Control System using Spark motor controllers. Wiring diagram and layout should be similar for other motor controllers. Where appropriate, a second set of images shows the wiring steps for using PWM controllers with integrated wires.

## Gather Materials



Locate the following control system components and tools

- Kit Materials:
  - Power Distribution Panel (PDP)

# Getting Started with the Control System

- roboRIO
  - Pneumatics Control Module (PCM)
  - Voltage Regulator Module (VRM)
  - OpenMesh radio (with power cable and Ethernet cable)
  - Robot Signal Light (RSL)
  - 4x Victor SPX or other speed controllers
  - 2x PWM y-cables
  - 120A Circuit breaker
  - 4x 40A Circuit breaker
  - 6 AWG Red wire
  - 10 AWG Red/Black wire
  - 18 AWG Red/Black wire
  - 22AWG yellow/green twisted CAN cable
  - 16x 10-12 AWG (yellow) ring terminals (8x quick disconnect pairs if using integrated wire controllers)
  - 2x Andersen SB50 battery connectors
  - 6AWG Terminal lugs
  - 12V Battery
  - Red/Black Electrical tape
  - Dual Lock material or fasteners
  - Zip ties
  - 1/4" or 1/2" plywood
- Tools Required:
- Wago Tool or small flat-head screwdriver
  - Very small flat head screwdriver (eyeglass repair size)
  - Philips head screw driver
  - 5mm Hex key (3/16" may work if metric is unavailable)
  - 1/16" Hex key
  - Wire cutters, strippers, and crimpers
  - 7/16" box end wrench or nut driver

## Create the Base for the Control System

For a benchtop test board, cut piece of 1/4" or 1/2" material (wood or plastic) approximately 24" x 16". For a Robot Quick Build control board see the supporting documentation for the proper size board for the chosen chassis configuration.

# Getting Started with the Control System

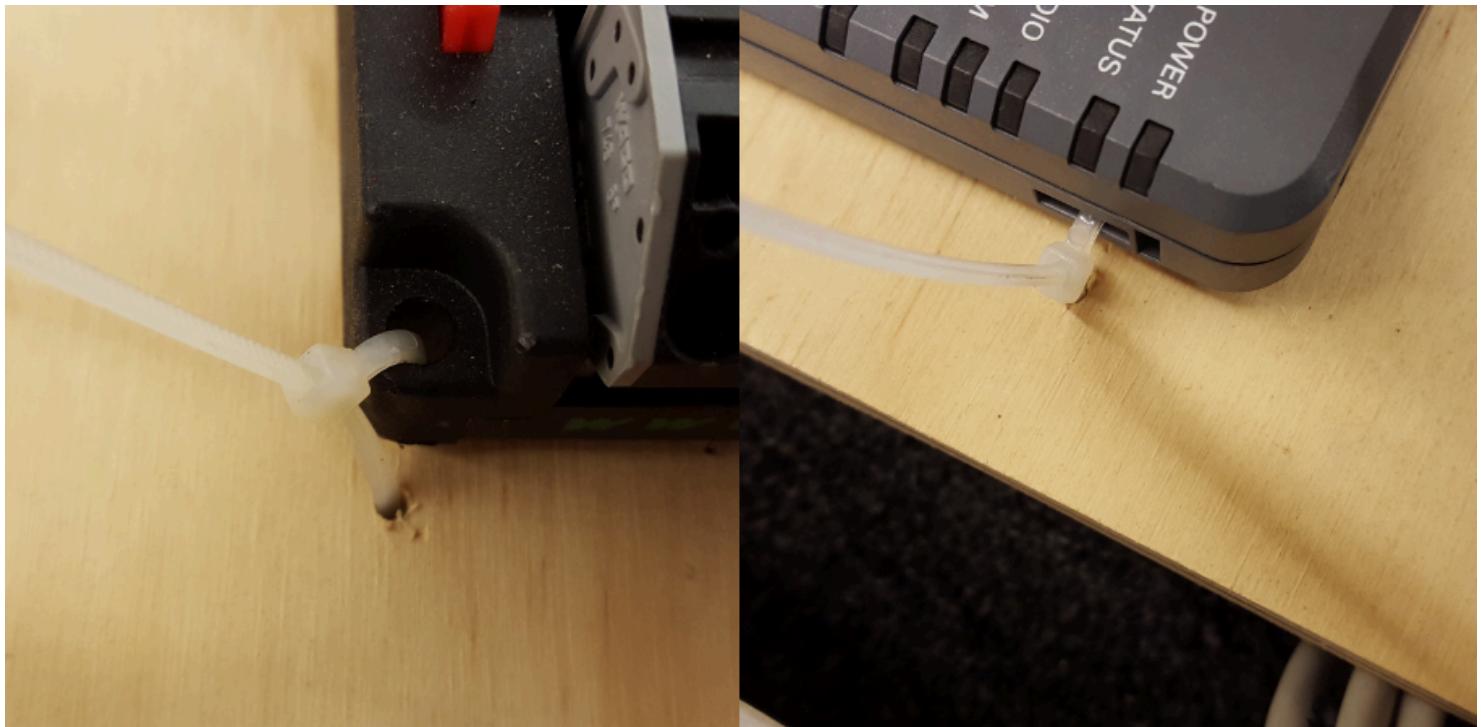
## Layout the Core Control System Components



Layout the components on the board. One layout that should work is shown in the images above.

# Getting Started with the Control System

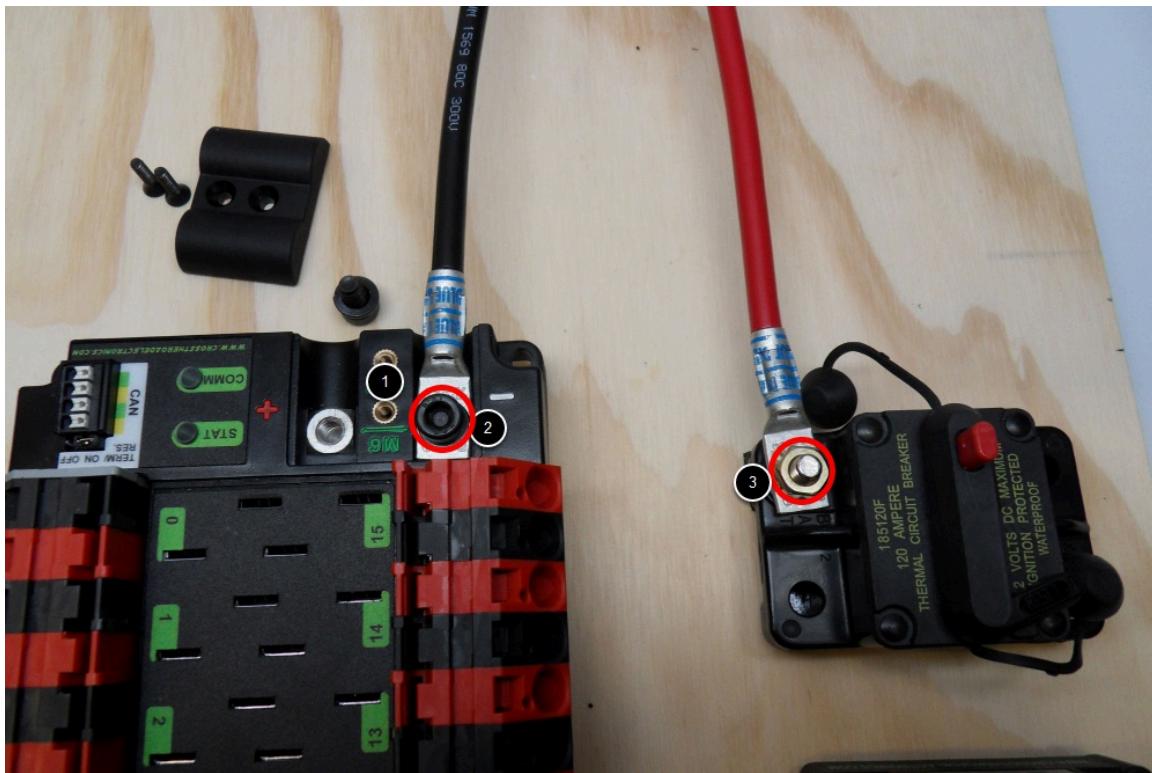
## Fasten components



Using the Dual Lock or hardware, fasten all components to the board. Note that in many FRC games robot-to-robot contact may be substantial and Dual Lock alone is unlikely to stand up as a fastener for many electronic components. Teams may wish to use nut and bolt fasteners or (as shown in the image above) cable ties, with or without Dual Lock to secure devices to the board.

# Getting Started with the Control System

## Attach Battery Connector to PDP

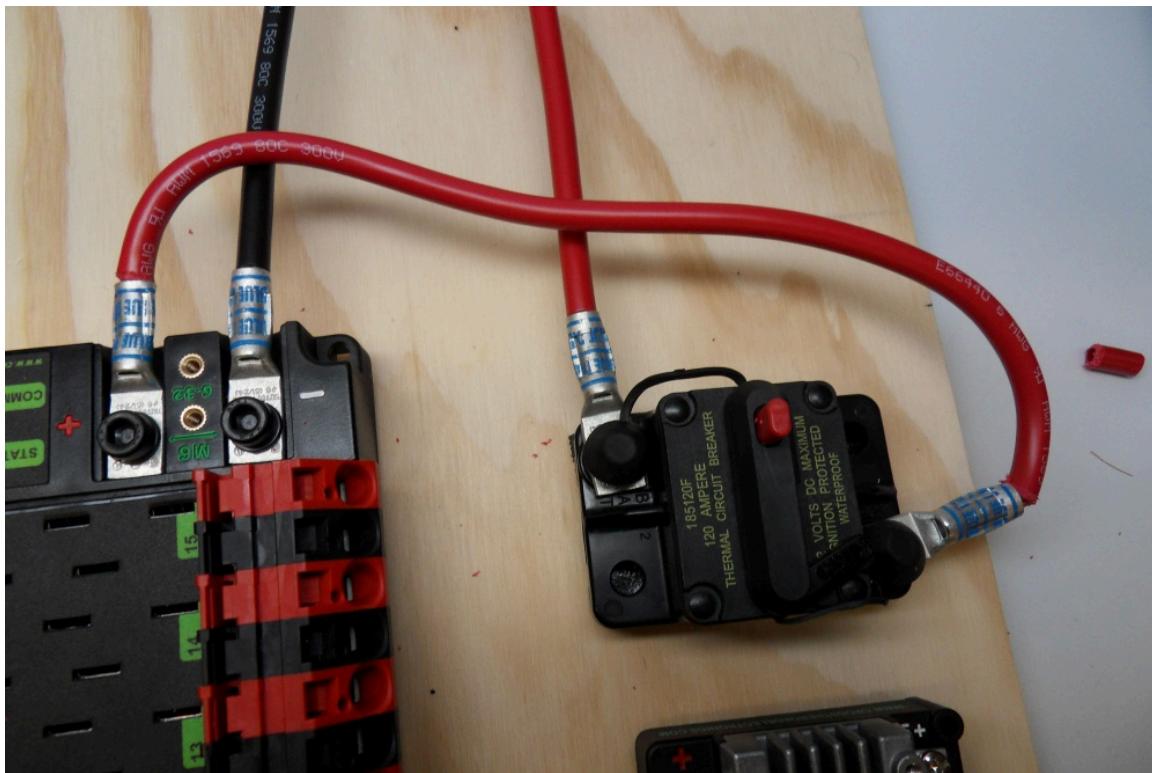


Requires: Battery Connector, 6AWG terminal lugs, 1/16" Allen, 5mm Allen, 7/16" Box end

1. Attach terminal lugs to battery connector.
2. Using a 1/16" Allen wrench, remove the two screws securing the PDP terminal cover.
3. Using a 5mm Allen wrench (3/16" will work if metric is not available), remove the negative (-) bolt and washer from the PDP and fasten the negative terminal of the battery connector.
4. Using a 7/16" box end wrench, remove the nut on the "Batt" side of the main breaker and secure the positive terminal of the battery conenector

# Getting Started with the Control System

## Wire Breaker to PDP



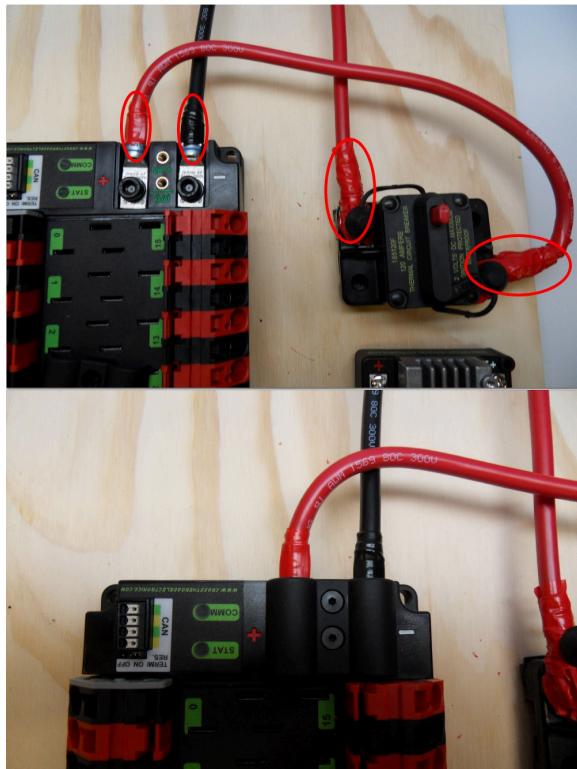
Requires: 6AWG red wire, 2x 6AWG terminal lugs, 5mm Allen, 7/16" box end

Secure one terminal lug to the end of the 6AWG red wire. Using the 7/16" box end, remove the nut from the "AUX" side of the 120A main breaker and place the terminal over the stud. Loosely secure the nut (you may wish to remove it shortly to cut, strip, and crimp the other end of the wire). Measure out the length of wire required to reach the positive terminal of the PDP.

1. Cut, strip, and crimp the terminal to the 2nd end of the red 6AWG wire.
2. Using the 7/16" box end, secure the wire to the "AUX" side of the 120A main breaker.
3. Using the 5mm, secure the other end to the PDP positive terminal.

# Getting Started with the Control System

## Insulate PDP connections



Requires: 1/16" Allen, Electrical tape

1. Using electrical tape, insulate the two connections to the 120A breaker. Also insulate any part of the PDP terminals which will be exposed when the cover is replaced. One method for insulating the main breaker connections is to wrap the stud and nut first, then use the tape wrapped around the terminal and wire to secure the tape.
2. Using the 1/16" Allen wrench, replace the PDP terminal cover

## Wago connectors

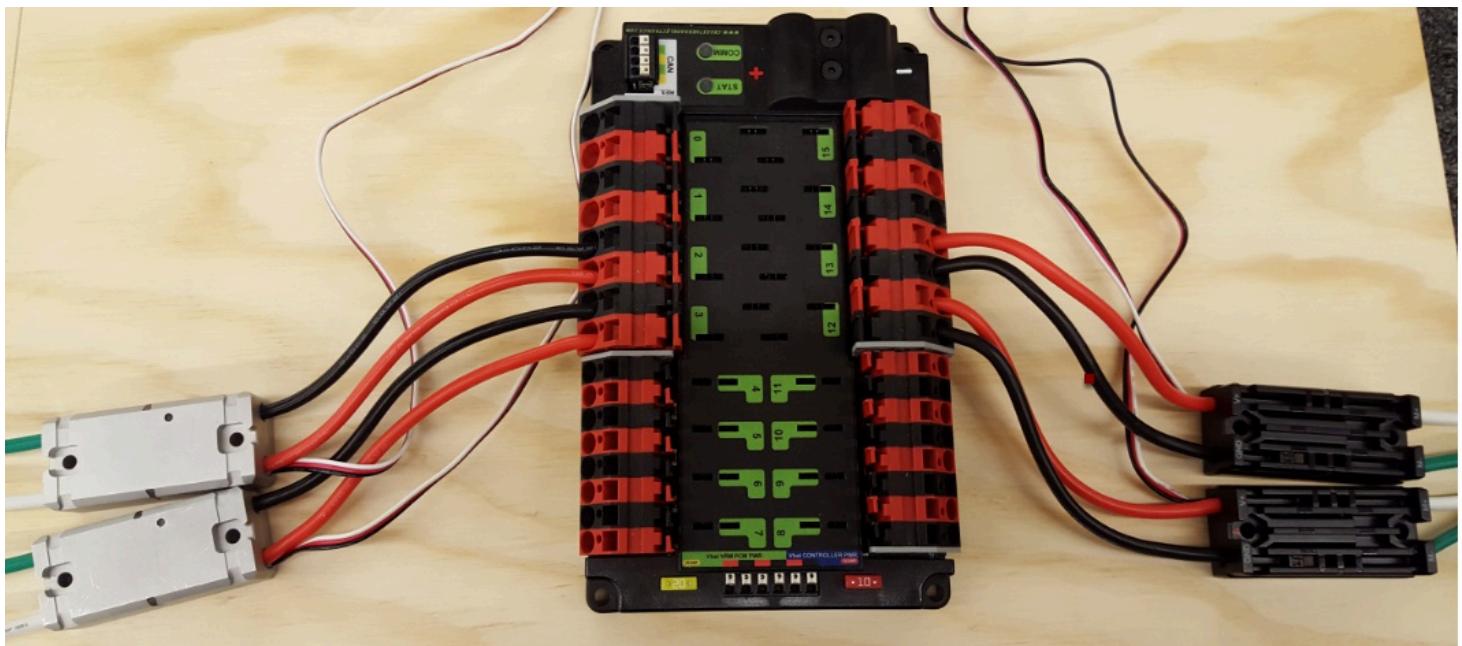
The next step will involve using the Wago connectors on the PDP. To use the Wago connectors, insert a small flat blade screwdriver into the rectangular hole at a shallow angle then angle the screwdriver upwards as you continue to press in to actuate the lever, opening the terminal. Two sizes of Wago connector are found on the PDP:

# Getting Started with the Control System

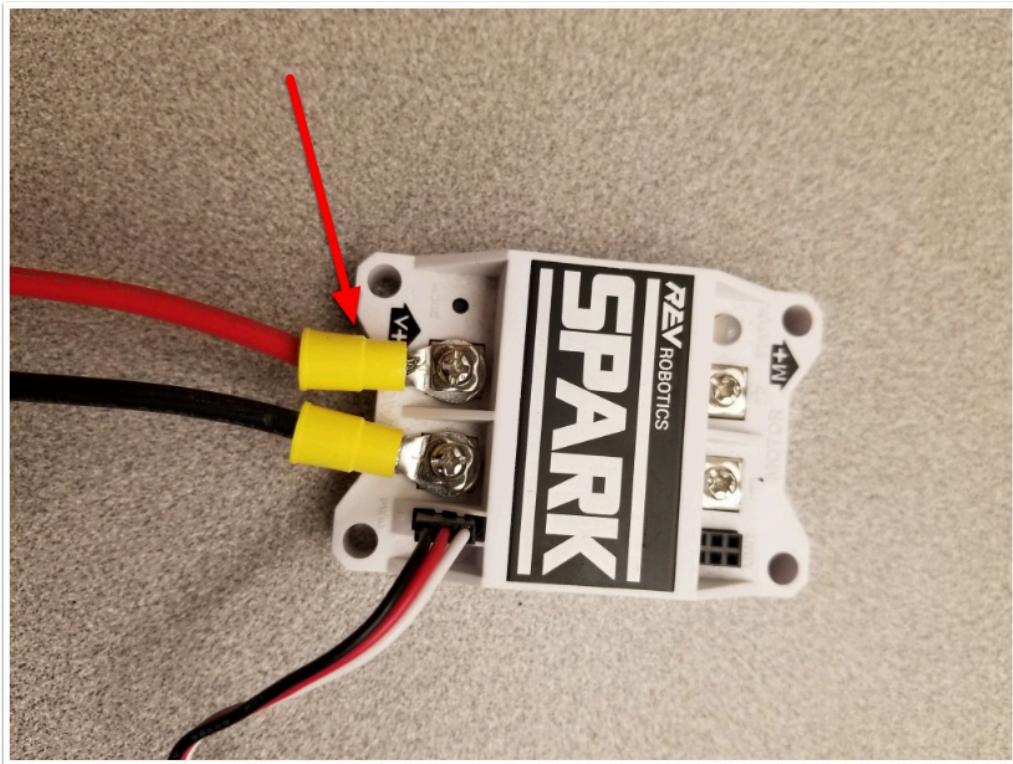
- Small Wago connector: Accepts 10AWG-24AWG, strip 11-12mm (~7/16")
- Large Wago connector: Accepts 6AWG-12AWG, strip 12-13mm(~1/2")

To maximize pullout force and minimize connection resistance wires should not be tinned (and ideally not twisted) before inserting into the Wago connector.

## Motor Controller Power



# Getting Started with the Control System



Requires: Wire Stripper, Small Flat Screwdriver, 10 or 12 AWG wire, 10 or 12 AWG fork/ring terminals (terminal controllers only), wire crimper

For Victor SPX or other wire integrated motor controllers (top image):

1. Cut and strip the red and black power input wires wire, then insert into one of the 40A (larger) Wago terminal pairs.

For terminal motor controllers (bottom image):

1. Cut red and black wire to appropriate length to reach from one of the 40A (larger) Wago terminal pairs to the input side of the speed controller (with a little extra for the length that will be inserted into the terminals on each end)
2. Strip one end of each of the wires, then insert into the Wago terminals.
3. Strip the other end of each wire, and crimp on a ring or fork terminal
4. Attach the terminal to the speed controller input terminals (red to +, black to -)

## Weidmuller Connectors

# Getting Started with the Control System

**The correct strip length is ~5/16" (8mm), not the 5/8" mentioned in the video.**

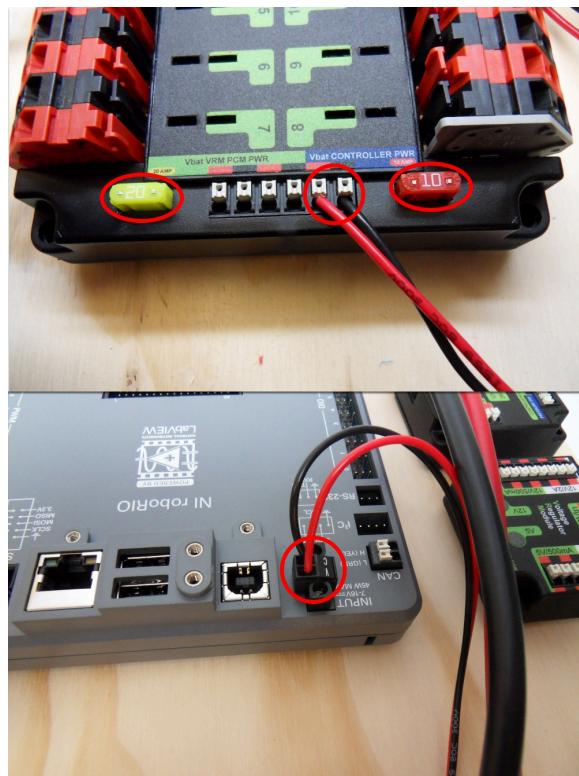
A number of the CAN and power connectors in the system use a Weidmuller LSF series wire-to-board connector. There are a few things to keep in mind when using this connector for best results:

- Wire should be 16AWG to 24AWG (consult rules to verify required gauge for power wiring)
- Wire ends should be stripped approximately 5/16"
- To insert or remove the wire, press down on the corresponding "button" to open the terminal

After making the connection check to be sure that it is clean and secure:

- Verify that there are no "whiskers" outside the connector that may cause a short circuit
- Tug on the wire to verify that it is seated fully. If the wire comes out and is the correct gauge it needs to be inserted further and/or stripped back further.

## roboRIO Power

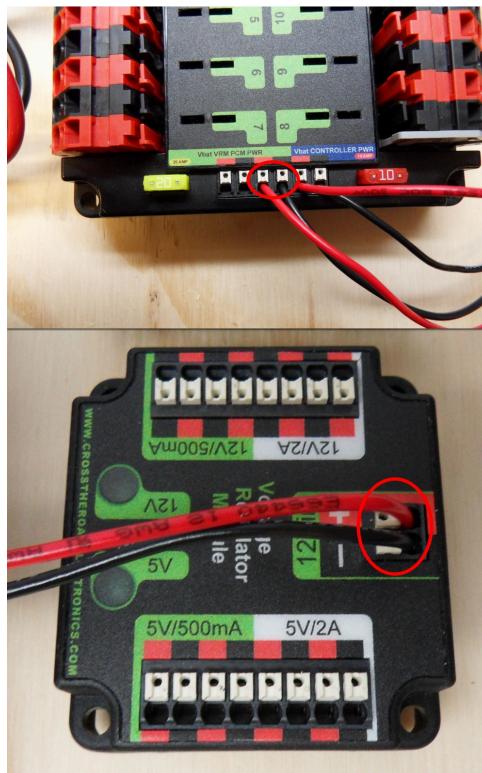


Requires: 10A/20A mini fuses, Wire stripper, very small flat screwdriver, 18AWG Red and Black

# Getting Started with the Control System

1. Insert the 10A and 20A mini fuses in the PDP in the locations shown on the silk screen (and in the image above)
2. Strip ~5/16" on both the red and black 18AWG wire and connect to the "Vbat Controller PWR" terminals on the PDB
3. Measure the required length to reach the power input on the roboRIO. Take care to leave enough length to route the wires around any other components such as the battery and to allow for any strain relief or cable management.
4. Cut and strip the wire.
5. Using a very small flat screwdriver connect the wires to the power input connector of the roboRIO (red to V, black to C). **Also make sure that the power connector is screwed down securely to the roboRIO.**

## Voltage Regulator Module Power



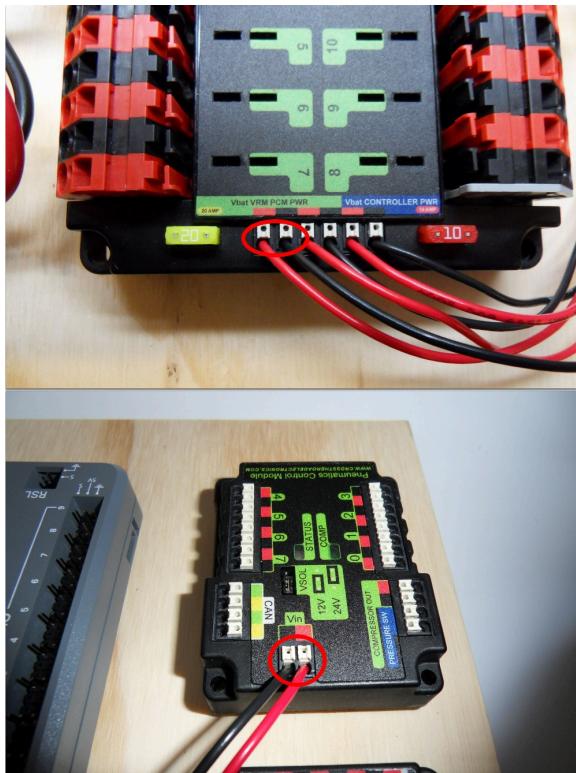
Requires: Wire stripper, small flat screwdriver (optional), 18AWG red and black wire

1. Strip ~5/16" on the end of the red and black 18AWG wire.
2. Connect the wire to one of the two terminal pairs labeled "Vbat VRM PCM PWR" on the PDP.

# Getting Started with the Control System

3. Measure the length required to reach the "12Vin" terminals on the VRM. Take care to leave enough length to route the wires around any other components such as the battery and to allow for any strain relief or cable management.
4. Cut and strip ~5/16" from the end of the wire.
5. Connect the wire to the VRM 12Vin terminals.

## Pneumatics Control Module Power (Optional)



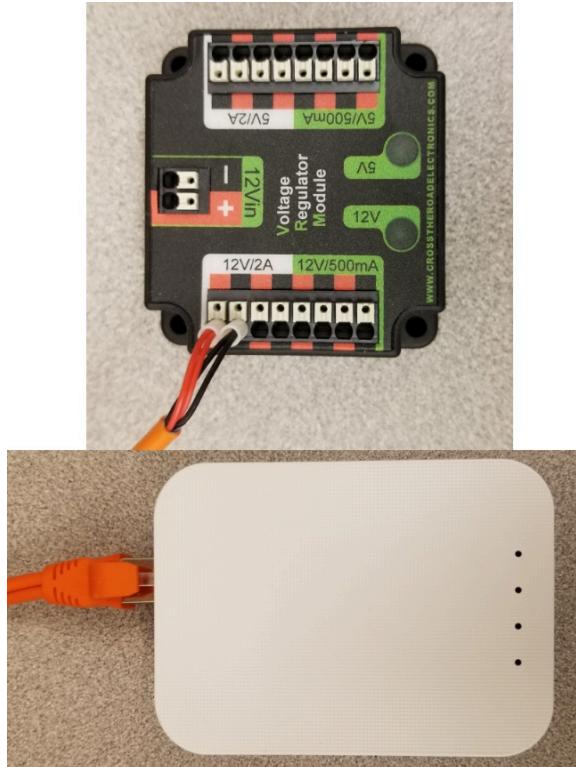
Requires: Wire stripper, small flat screwdriver (optional), 18AWG red and black wire

Note: The PCM is an optional component used for controlling pneumatics on the robot.

1. Strip ~5/16" on the end of the red and black 18AWG wire.
2. Connect the wire to one of the two terminal pairs labeled "Vbat VRM PCM PWR" on the PDP.
3. Measure the length required to reach the "Vin" terminals on the VRM. Take care to leave enough length to route the wires around any other components such as the battery and to allow for any strain relief or cable management.
4. Cut and strip ~5/16" from the end of the wire.
5. Connect the wire to the VRM 12Vin terminals.

# Getting Started with the Control System

## Radio Power and Ethernet



**!** **DO NOT connect the Rev passive POE injector cable directly to the roboRIO. The roboRIO MUST connect to the female end of the cable using an additional Ethernet cable as shown in the next step.**

Requires: Small flat screwdriver (optional), Rev radio PoE cable

1. Insert the ferrules of the passive PoE injector cable into the corresponding colored terminals on the 12V/2A section of the VRM.
2. Connect the male RJ45 (Ethernet) end of the cable into the Ethernet port on the radio closest to the barrel connector (labeled 18-24v POE)

# Getting Started with the Control System

## RoboRIO to Radio Ethernet

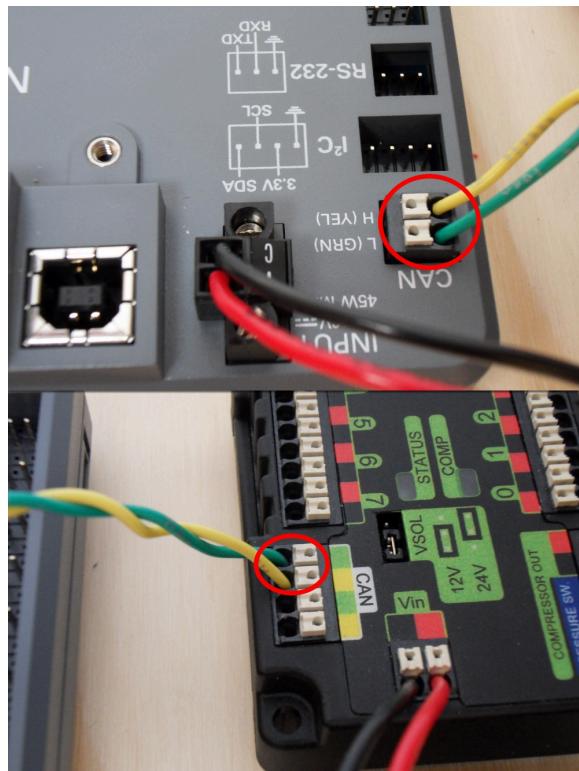


Requires: Ethernet cable

Connect an Ethernet cable from the female RJ45 (Ethernet) port of the Rev Passive POE cable to the RJ45 (Ethernet) port on the roboRIO.

# Getting Started with the Control System

## RoboRIO to PCM CAN



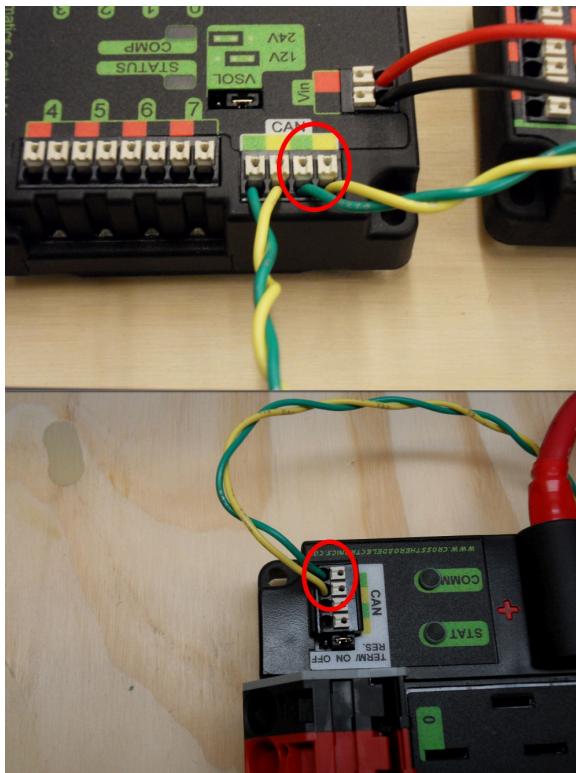
Requires: Wire stripper, small flat screwdriver (optional), yellow/green twisted CAN cable

Note: The PCM is an optional component used for controlling pneumatics on the robot. If you are not using the PCM, wire the CAN connection directly from the roboRIO (shown in this step) to the PDP (show in the next step).

1. Strip ~5/16" off of each of the CAN wires.
2. Insert the wires into the appropriate CAN terminals on the roboRIO (Yellow->YEL, Green->GRN).
3. Measure the length required to reach the CAN terminals of the PCM (either of the two available pairs). Cut and strip ~5/16" off this end of the wires.
4. Insert the wires into the appropriate color coded CAN terminals on the PCM. You may use either of the Yellow/Green terminal pairs on the PCM, there is no defined in or out.

# Getting Started with the Control System

## PCM to PDP CAN



Requires: Wire stripper, small flat screwdriver (optional), yellow/green twisted CAN cable

Note: The PCM is an optional component used for controlling pneumatics on the robot. If you are not using the PCM, wire the CAN connection directly from the roboRIO (shown in the above step) to the PDP (show in this step).

1. Strip ~5/16" off of each of the CAN wires.
2. Insert the wires into the appropriate CAN terminals on the PCM.
3. Measure the length required to reach the CAN terminals of the PDP (either of the two available pairs). Cut and strip ~5/16" off this end of the wires.
4. Insert the wires into the appropriate color coded CAN terminals on the PDP. You may use either of the Yellow/Green terminal pairs on the PDP, there is no defined in or out.

Note: The PDP ships with the CAN bus terminating resistor jumper in the "ON" position. It is recommended to leave the jumper in this position and place any additional CAN nodes between the roboRIO and the PDP (leaving the PDP as the end of the bus). If you wish to place the PDP in

# Getting Started with the Control System

the middle of the bus (utilizing both pairs of PDP CAN terminals) move the jumper to the "OFF" position and place your own 120 ohm terminating resistor at the end of your CAN bus chain.

## PWM Cables



Requires: 4x PWM cables (if using non-integrated wire controllers), 2x PWM Y-cable (Optional)

Option 1 (Direct connect):

1. Connect the PWM cables from each controller directly to the roboRIO. For Victor SPX's and other PWM/CAN controllers, the green wire (black wire for non-integrated controllers) should be towards the outside of the roboRIO. For controllers without integrated wires, make sure the controller side of the black wire is located according to the markings on the controller. It is recommended to connect the left side to PWM 0 and 1 and the right side to PWM 2 and 3 for the most straightforward programming experience, but any channel will work as long as you note which side goes to which channel and adjust the code accordingly.

Option 2 (Y-cable):

# Getting Started with the Control System

1. Connect 1 PWM Y-cable to the PWM cables for the controllers controlling one side of the robot. The brown wire on the Y-cable should match the green/black wire on the PWM cable.
2. Connect the PWM Y-cables to the PWM ports on the roboRIO. The brown wire should be towards the outside of the roboRIO. It is recommended to connect the left side to PWM 0 and the right side to PWM 1 for the most straightforward programming experience, but any channel will work as long as you note which side goes to which channel and adjust the code accordingly.

## Robot Signal Light



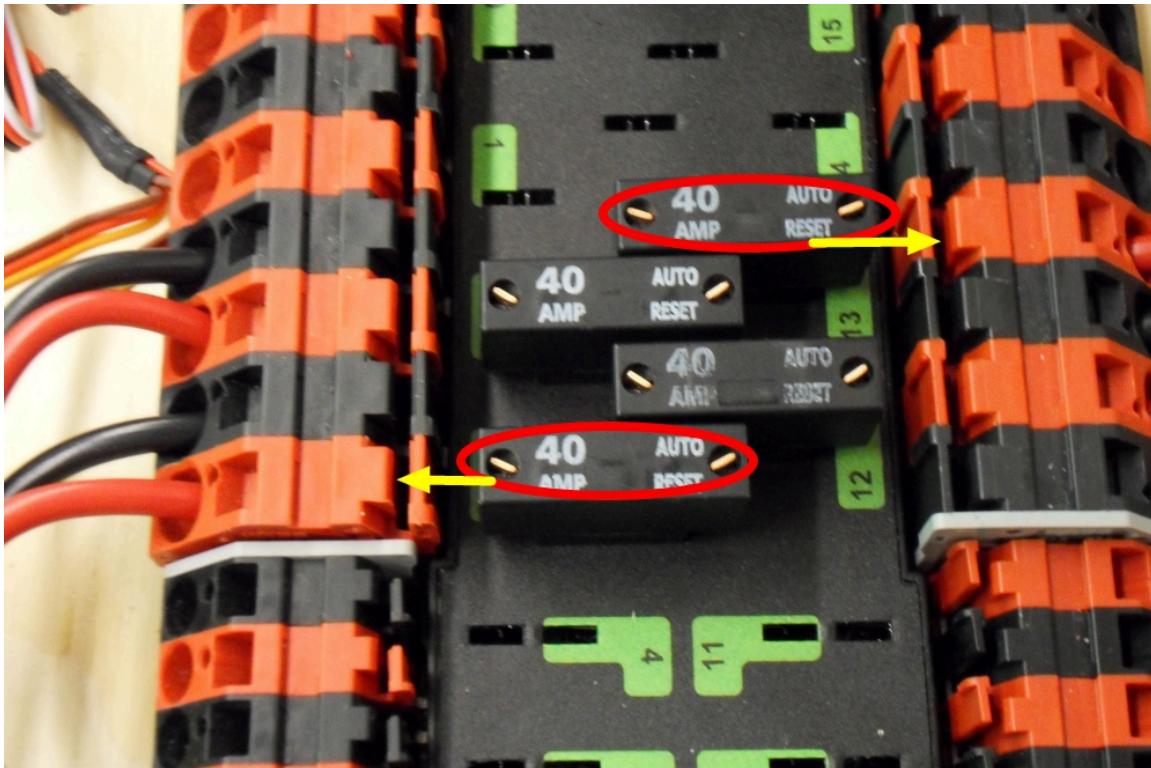
Requires: Wire stripper, 2 pin cable, Robot Signal Light, 18AWG red wire, very small flat screwdriver

1. Cut one end off of the 2 pin cable and strip both wires
2. Insert the black wire into the center, "N" terminal and tighten the terminal.
3. Strip the 18AWG red wire and insert into the "La" terminal and tighten the terminal.
4. Cut and strip the other end of the 18AWG wire to insert into the "Lb" terminal
5. Insert the red wire from the two pin cable into the "Lb" terminal with the 18AWG red wire and tighten the terminal.
6. Connect the two-pin connector to the RSL port on the roboRIO. The black wire should be closest to the outside of the roboRIO.

# Getting Started with the Control System

You may wish to temporarily secure the RSL to the control board using zip ties or Dual Lock (it is recommended to move the RSL to a more visible location as the robot is being constructed)

## Circuit Breakers



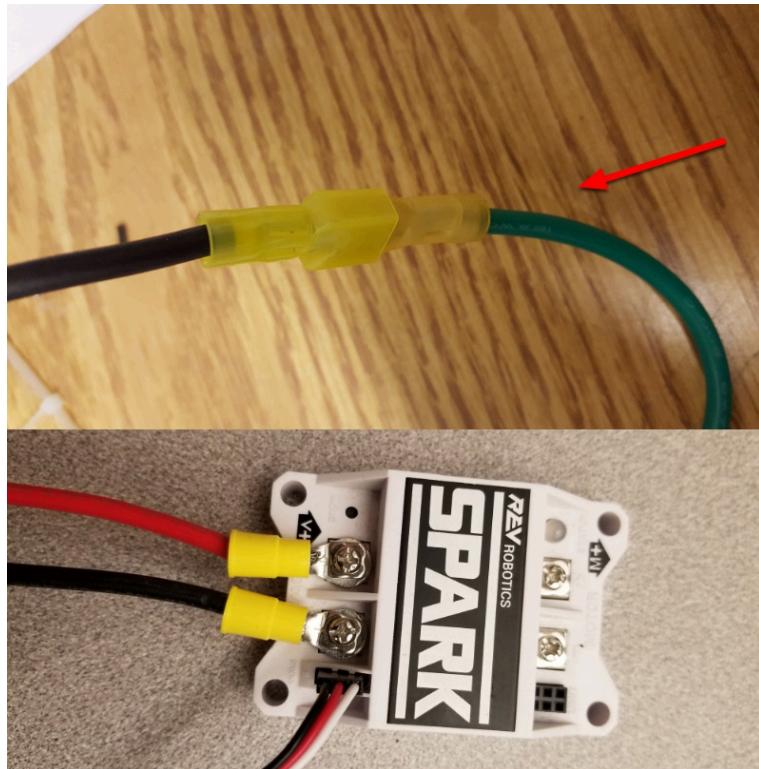
Requires: 4x 40A circuit breakers

Insert 40-amp Circuit Breakers into the positions on the PDP corresponding with the Wago connectors the Talons are connected to. Note that, for all breakers, the breaker corresponds with the nearest positive (red) terminal (see graphic above). All negative terminals on the board are directly connected internally.

**If working on a Robot Quick Build, stop here and insert the board into the robot chassis before continuing.**

# Getting Started with the Control System

## Motor Power



Requires: Wire stripper, wire crimper, phillips head screwdriver, wire connecting hardware

For each CIM motor:

1. Strip the ends of the red and black wires from the CIM

For integrated wire controllers (including Victor SPX):

1. Strip the white and green wires from the controller
2. Connect the motor wires to the controller output wires (it is recommended to connect the red wire to the white M+ output). The images above show examples using quick disconnect terminals.

For Sparks or other non-integrated-wire controllers:

1. Crimp a ring/fork terminal on each of the motor wires.
2. Attach the wires to the output side of the motor controller (red to +, black to -)

# Getting Started with the Control System

**STOP**



**STOP!!**

Before plugging in the battery, make sure all connections have been made with the proper polarity. Ideally have someone that did not wire the robot check to make sure all connections are correct.

- Start with the battery and verify that the red wire is connected to the positive terminal
- Check that the red wire passes through the main breaker and to the + terminal of the PDP and that the black wire travels directly to the - terminal.
- For each motor controller, verify that the red wire goes from the red PDP terminal to the Talon input labeled with the red + (not the white M+!!!!)
- For each device on the end of the PDP, verify that the red wire connects to the red terminal on the PDP and the red terminal on the component.
- **Make sure that the orange Passive PoE cable is plugged directly into the radio NOT THE roboRIO! It must be connected to the roboRIO using an additional Ethernet cable.**

# Getting Started with the Control System

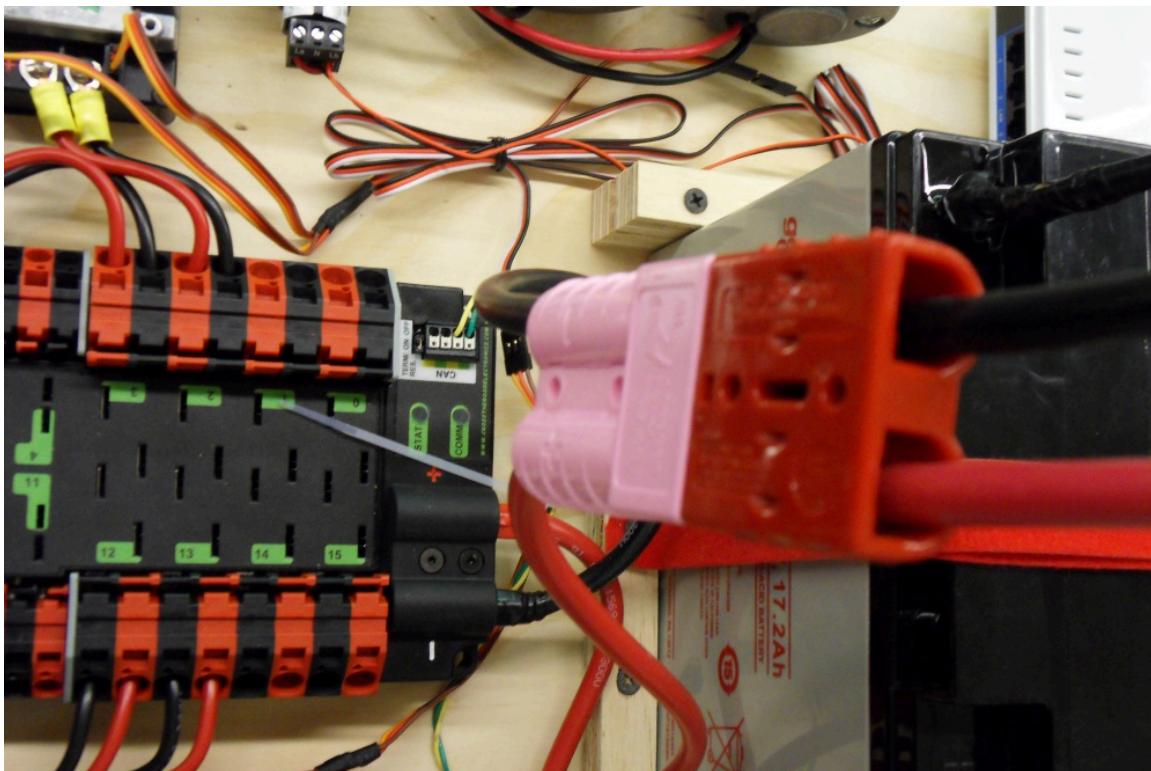
**It is also recommended to put the robot on blocks so the wheels are off the ground before proceeding. This will prevent any unexpected movement from becoming dangerous.**

## Manage Wires

Requires: Zip ties

Now may be a good time to add a few zip ties to manage some of the wires before proceeding. This will help keep the robot wiring neat.

## Connect Battery



Connect the battery to the robot side of the Andersen connector. Power on the robot by moving the lever on the top of the 120A main breaker into the ridge on the top of the housing.

# Imaging your Classmate (Veteran Image Download)

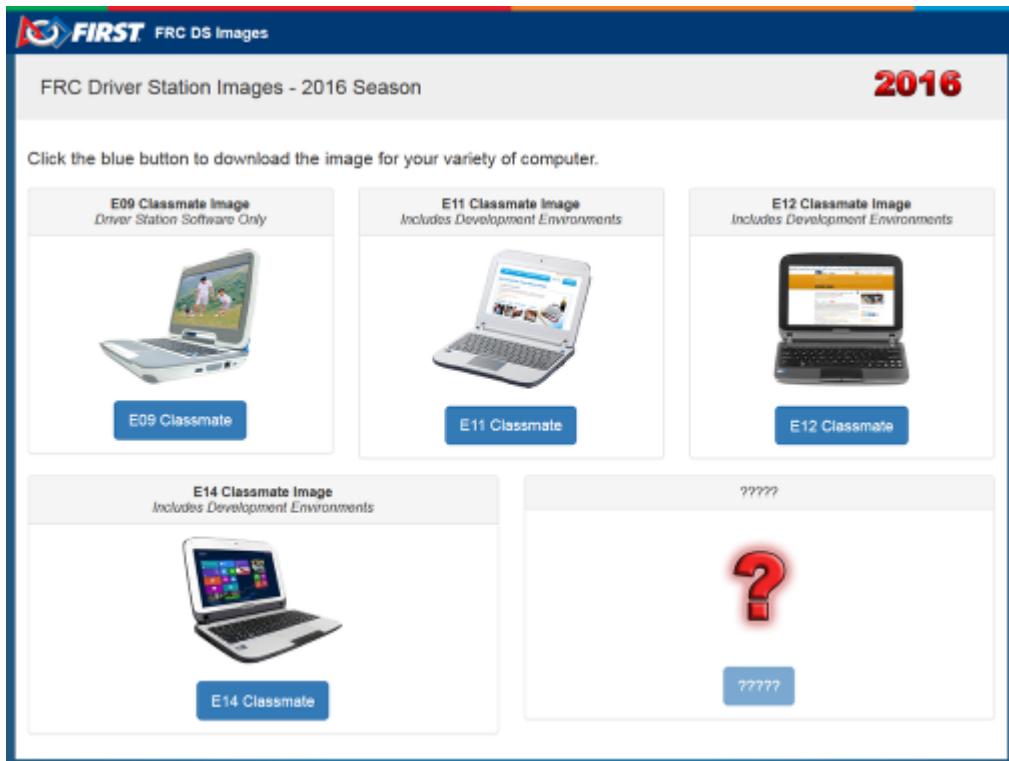
This document describes the procedure for creating a bootable USB drive to restore the 2017 FRC image on a Classmate computer. Note that Veteran teams are not required to re-image their Classmates. If you do not wish to re-image your Classmate then you can start with the appropriate document for [C++\Java](#), [LabVIEW](#), or [DS only](#).

## Prerequisites

1. E09, E11, E12, or E14 Classmate computer or Acer ES1 computer
2. 16GB or larger USB drive
3. 7-Zip software installed. [Download here \(www.7-zip.org\)](http://www.7-zip.org) As of the writing of this document, the current released version is 18.06 (2018-12-30)
4. RMprepUSB software installed. [Download here \(http://www.rmprepusb.com/documents/release-2-0\)](http://www.rmprepusb.com/documents/release-2-0) Scroll down the page and select the stable (Full) version's download link. As of the writing of this document, the current stable version is 2.1.741a

# Getting Started with the Control System

## Download the Computer Image



Download the image from the [FIRST FRC Driver Station System Image Portal](#). There are several computer images available, one for each model. On the download site, select the option that matches your computer by clicking the button below the image. Due to the limited size of hard drive in the E09, it is supported with a DS/Utilities image only and does not have the IDEs for LabVIEW or C++/Java installed. All other images have the LabVIEW base installation already present.

**NOTE: These images only install the prerequisite core FRC software, it is still necessary to install the FRC specific updates. See the [Update Software](#) step for more information.**

- ! Note: Due to computer availability, the E14 image provided is the 2018 image. This image will still load and activate, however the wrong LabVIEW version will be installed and it will

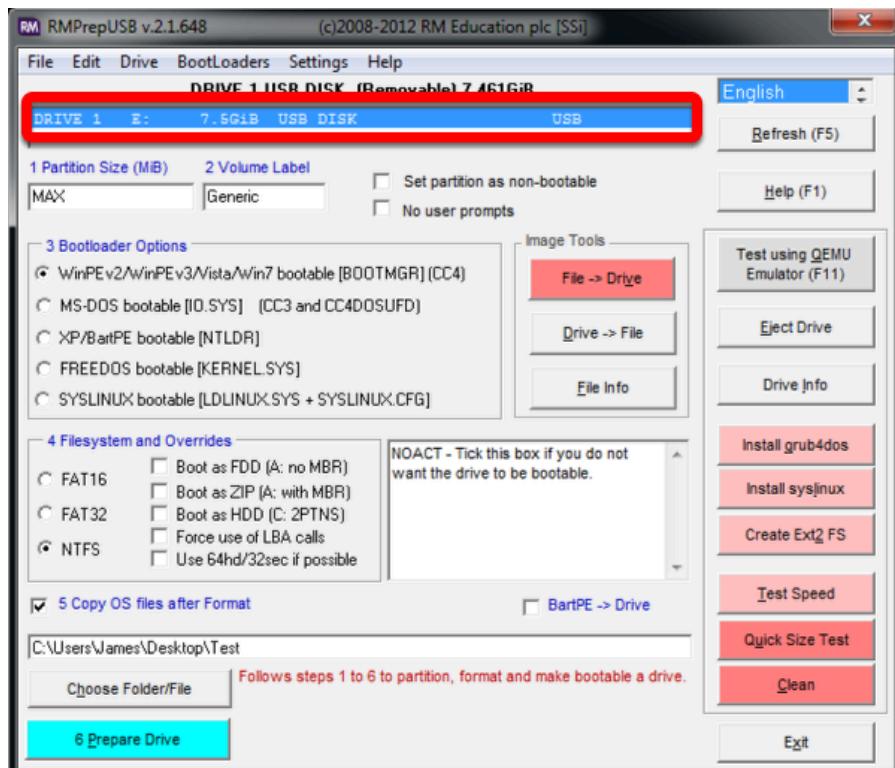
# Getting Started with the Control System

contain Eclipse for C++/Java. If you wish to use this computer for FRC 2019 development you will need to load the appropriate base software in addition to any updates.

## Preparation

1. Place the image file downloaded from the site to a folder on your root drive (e.g. C:\2016\_Image)
2. Connect 16GB or larger USB Flash drive to the PC to use as the new restoration drive.

## RMPrep

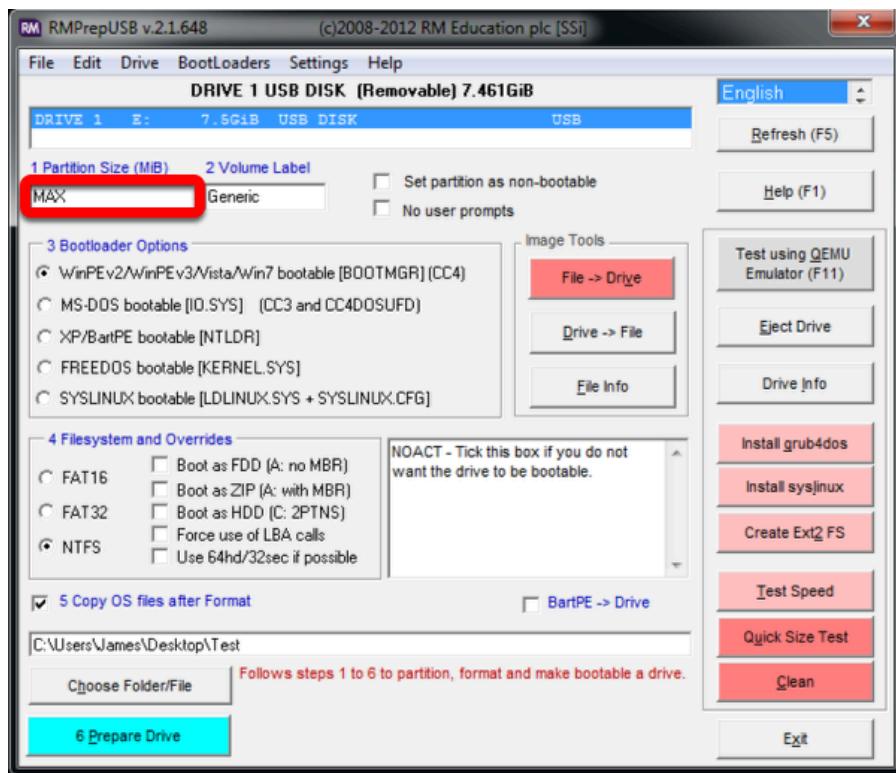


Start/Run RMprepUSB

Select USB Drive

# Getting Started with the Control System

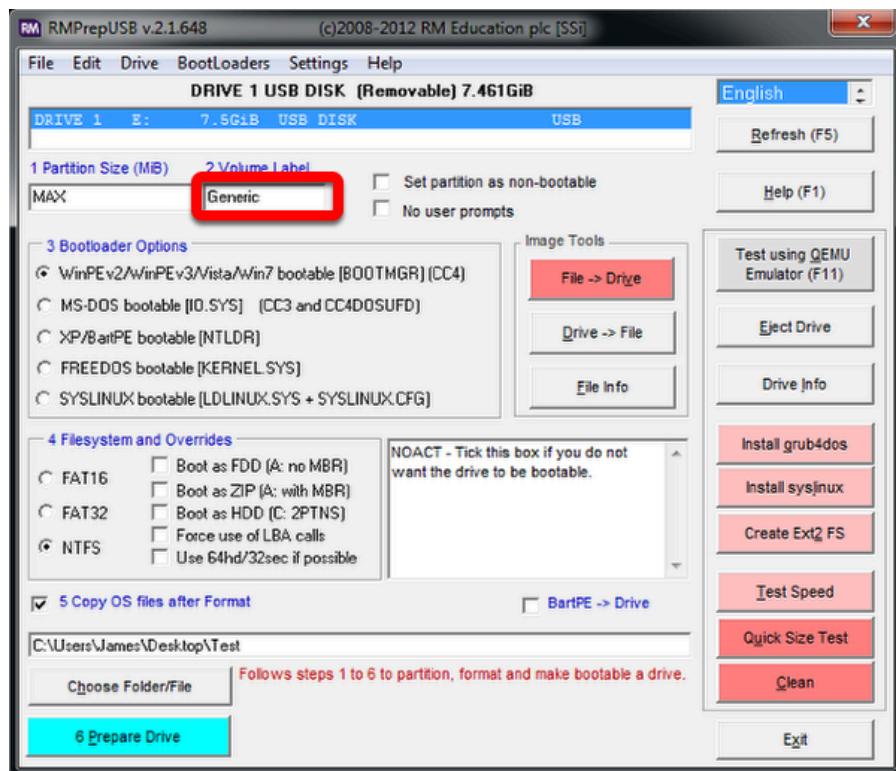
## Set Partition Size



Set Partition Size to MAX

# Getting Started with the Control System

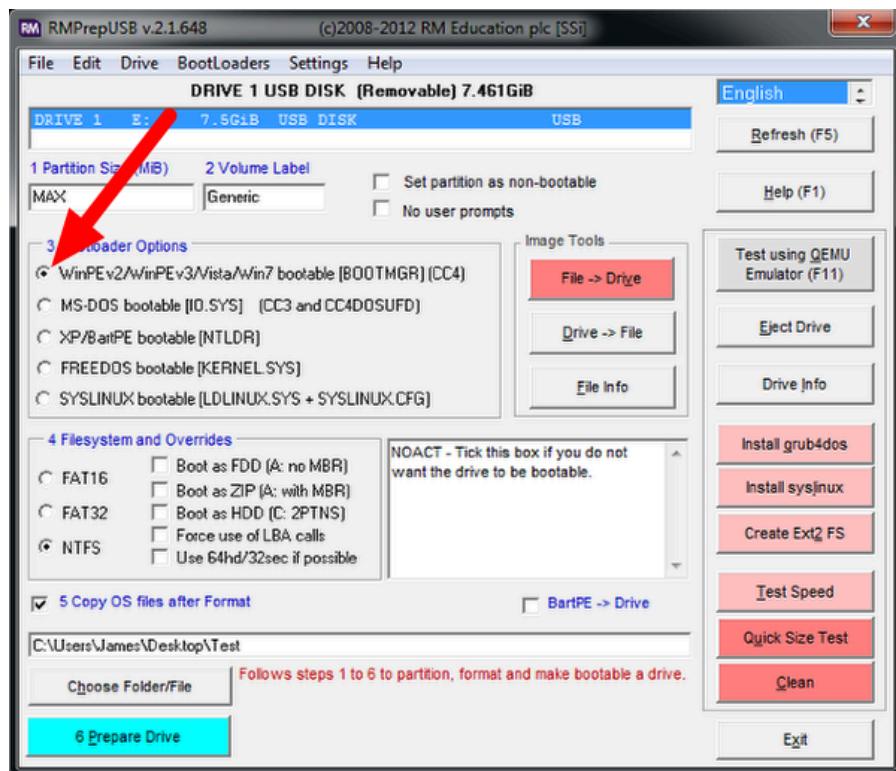
## Set Volume Label



Set Volume Label to Generic

# Getting Started with the Control System

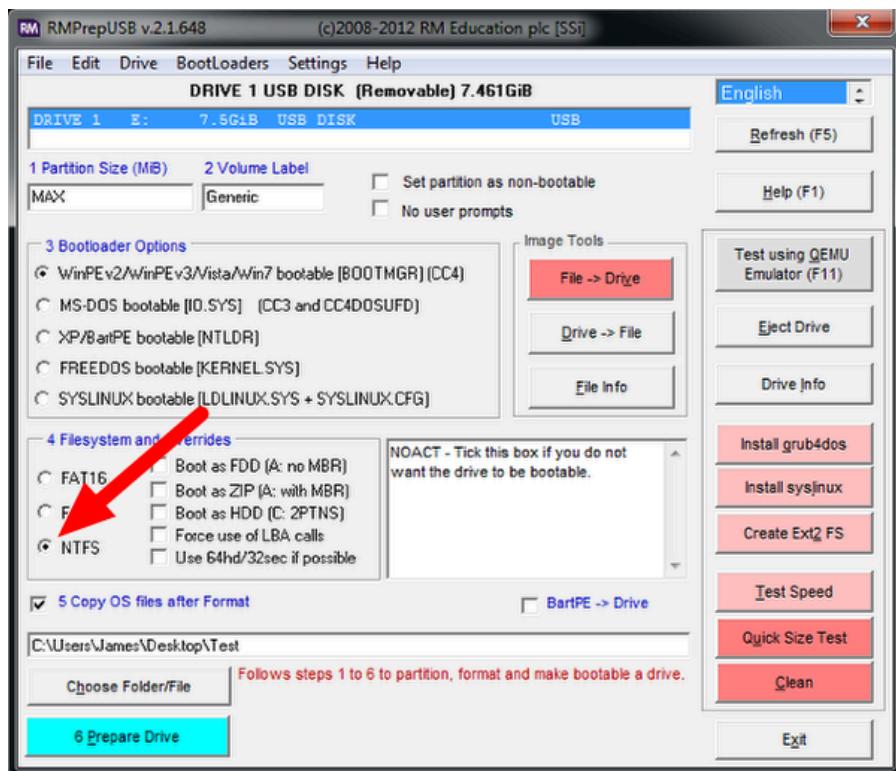
## Set Bootloader Option



Select Bootloader Option "WinPE v2/WinPE v3/Vista/Win7 bootable"

# Getting Started with the Control System

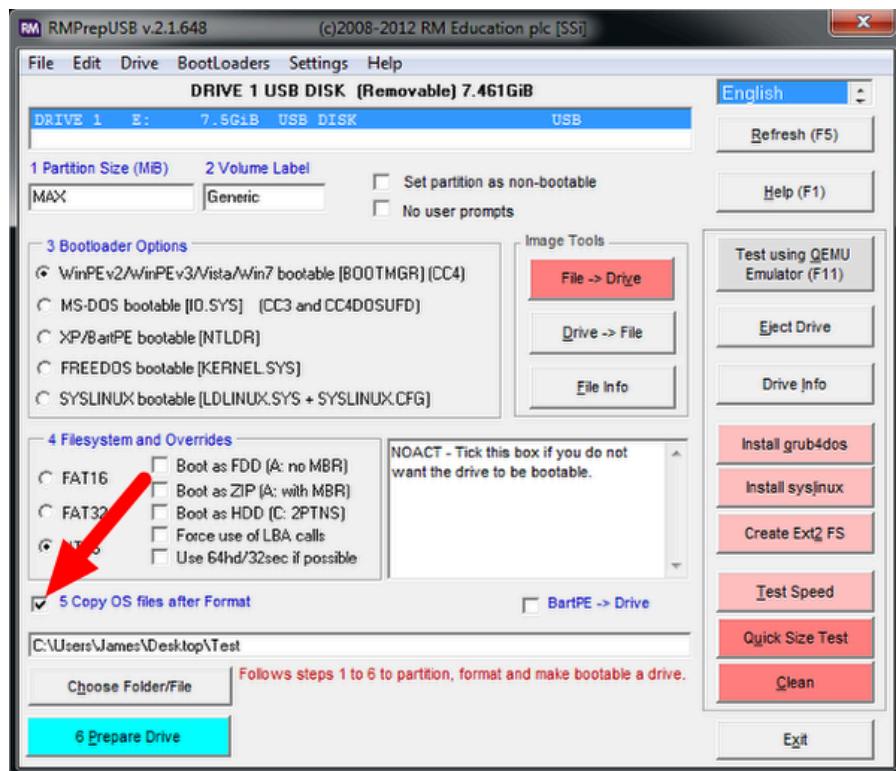
## Select Filesystem



Select NTFS Filesystem

# Getting Started with the Control System

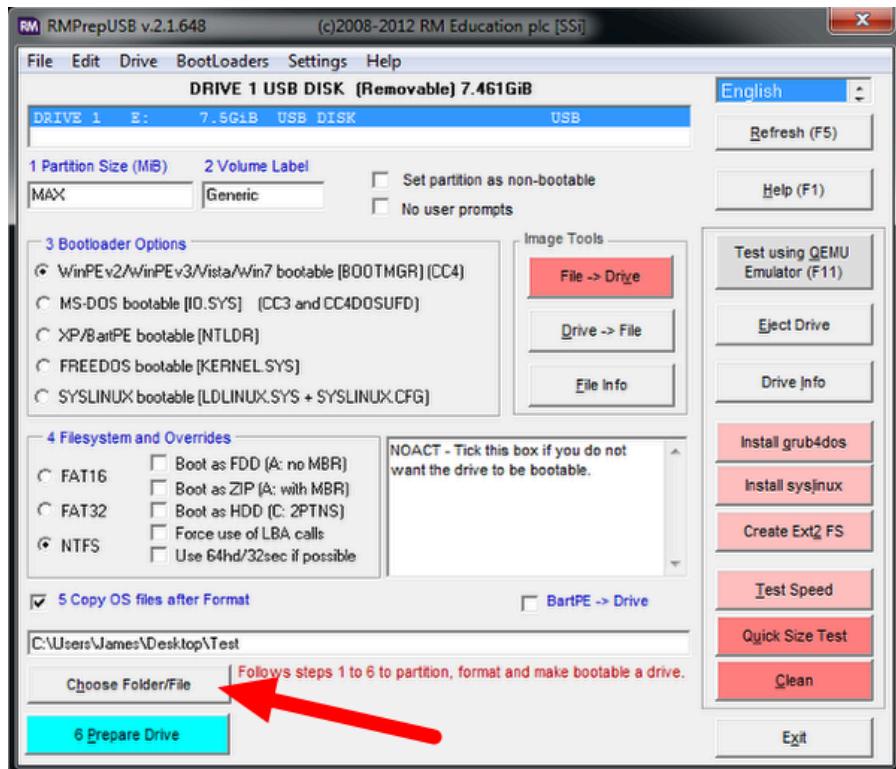
## Copy OS Files Option



Ensure the “Copy OS files after Format” box is checked

# Getting Started with the Control System

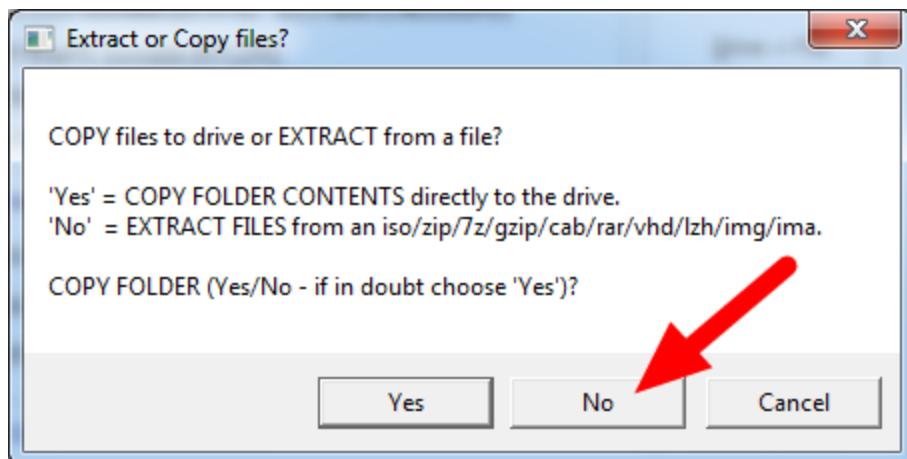
## Locate Image



Select the “Choose Folder/File” button

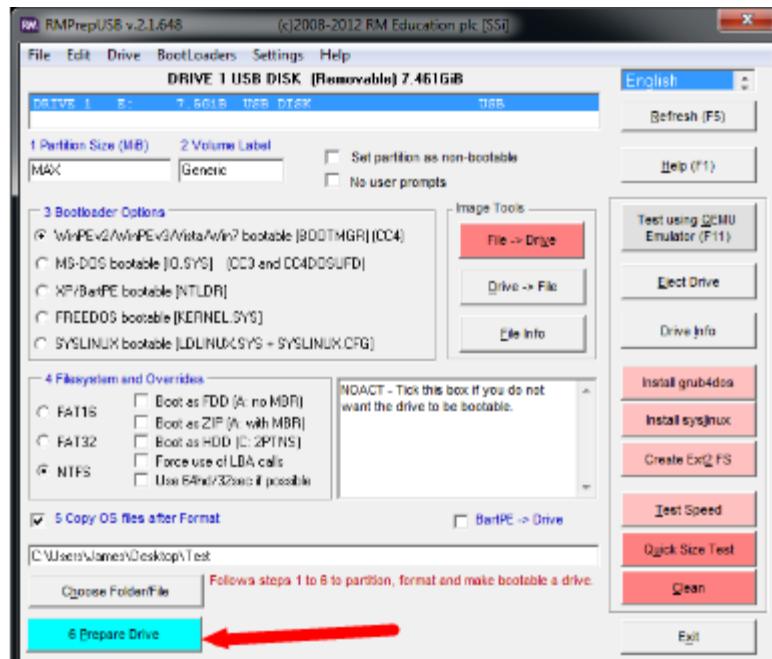
# Getting Started with the Control System

## Copy Files Dialog



Choose "No" and select your .7z image

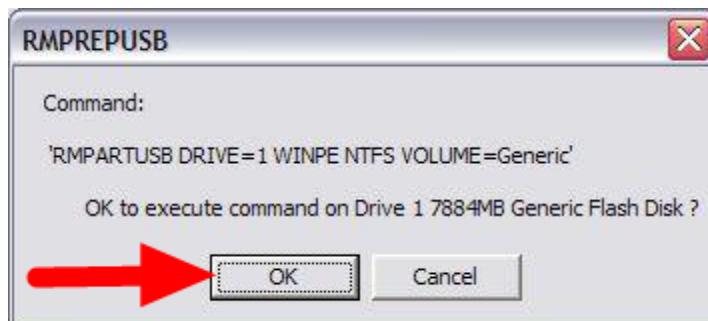
## Prepare Drive



# Getting Started with the Control System

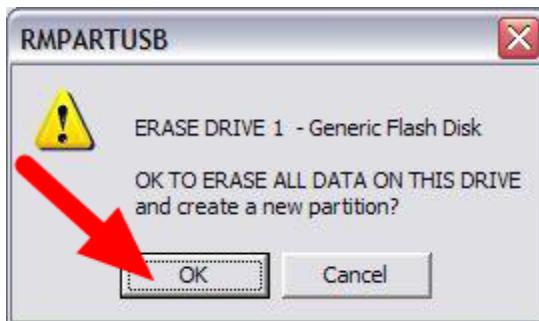
All configuration settings are now complete. Select “Prepare Drive” to begin the process

## Confirmation Dialog 1



Click “OK” to execute the command on the selected USB Flash drive. A Command Prompt will open showing the progress.

## Confirmation Dialog 2



Click “OK” to format the USB drive

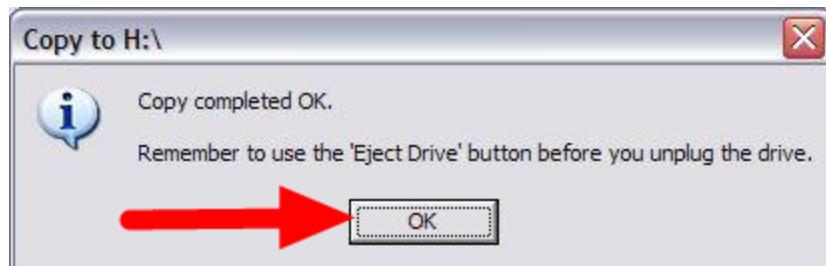
**NOTE: ALL DATA ON THE DRIVE WILL BE ERASED!**

## Decryption

**Note: If you are using an encrypted version of the image downloaded before kickoff you will be prompted to enter the decryption key found at the end of the Kickoff video.**

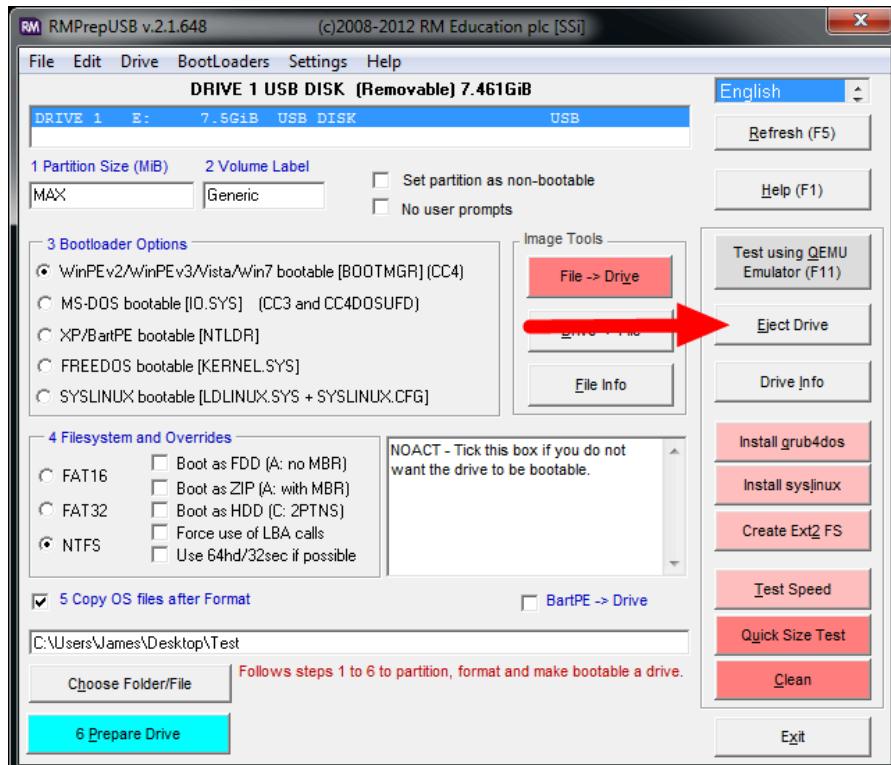
# Getting Started with the Control System

## Copy Complete



Once formatting is complete, the restoration files will be extracted and copied to the USB drive. This process should take ~15 minutes when connected to a USB 2.0 port. When all files have been copied, this message will appear, press OK to continue.

## Eject Drive



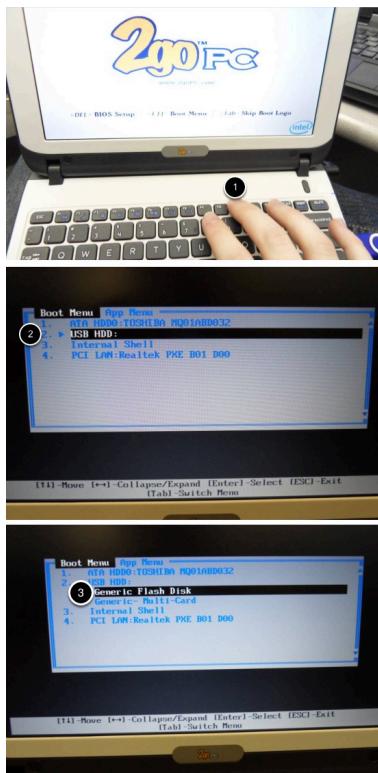
# Getting Started with the Control System

Press the “Eject Drive” button to safely remove the USB drive. The USB drive is now ready to be used to restore the image onto the PC.

## Hardware Setup

1. Make sure the computer is turned off, but plugged in.
2. Insert the USB Thumb Drive into a USB port on the Driver Station computer.

## Boot to USB



### Classmate:

1. Power on the Classmate and tap the F11 key on the keyboard. Tapping the F11 key during boot will bring up the boot menu.
2. Use the up/down keys to select the **USB HDD:** entry on the menu, then press the right arrow to expand the listing
3. Use the up/down arrow keys on the keyboard to select the USB device (it will be called “Generic Flash Disk”). Press the ENTER key when the USB device is highlighted.

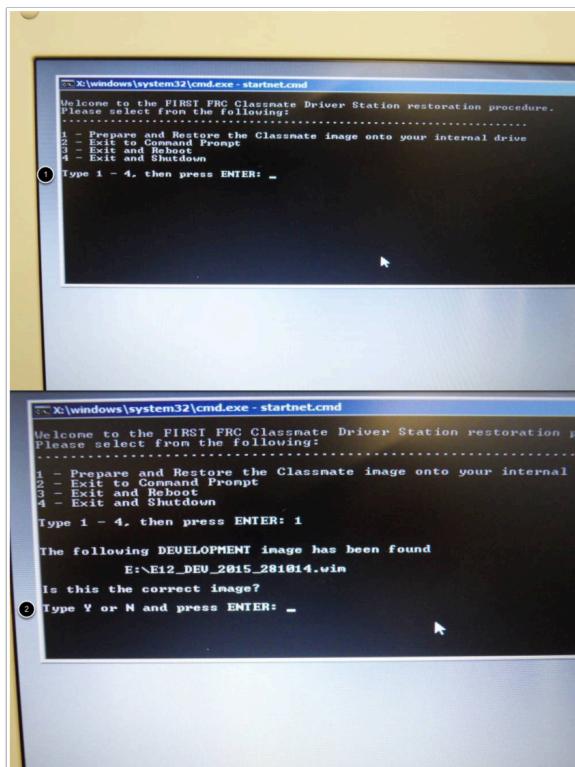
# Getting Started with the Control System

## Acer ES1

1. Power on the computer and tap the F12 key on the keyboard. Tapping the F12 key during boot will bring up the boot menu.
2. Use the up/down keys to select the **USB HDD: Generic** entry on the menu, then press the ENTER key when the USB device is highlighted.

**Acer ES1: If pressing F12 does not pull up the boot menu or if the USB device is not listed in the boot menu, see "Checking BIOS Settings" at the bottom of this article.**

## Image the Classmate



1. To confirm that you want to reimagine the Classmate, type "1" and press ENTER.
2. Then, type "Y" and press ENTER. The Classmate will begin re-imaging. The installation will take 15-30 minutes.
3. When the installation is complete, remove the USB drive.
4. Restart the Classmate. The Classmate will boot into Windows.

# Getting Started with the Control System

## Initial Driver Station Boot

The first time the Classmate is turned on, there are some unique steps, listed below, that you'll need to take. The initial boot may take several minutes; make sure you do not cycle power during the process.

Please note that these steps are only required during original startup.

## Enter Setup

1. Log into the Developer account.
2. Click "Ask me later".
3. Click "OK". The computer now enters a Set Up that may take a few minutes.

## Activate Windows

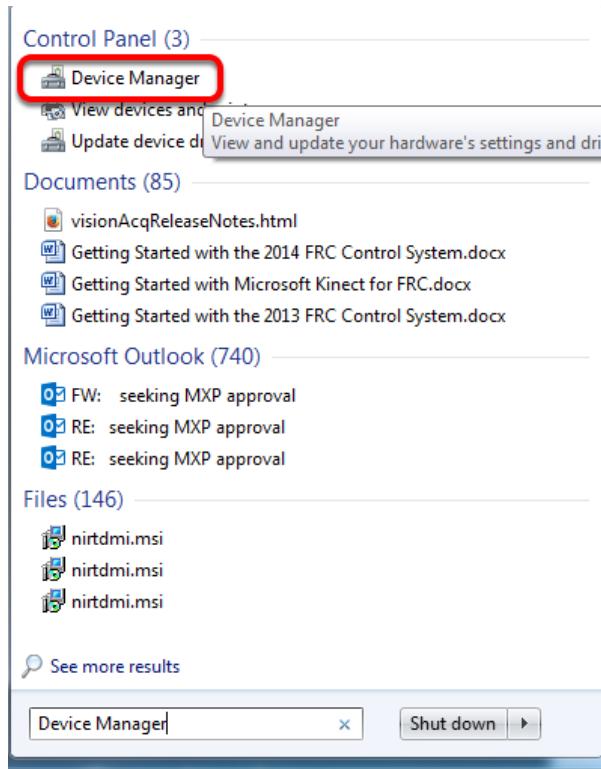
1. Establish an Internet connection.
2. Once you have an Internet connection, click the Start menu, right click "Computer" and click "Properties".
3. Scroll to the bottom section, "Windows activation", and Click "Activate Windows now"
4. Click "Activate Windows online now". The activation may take a few minutes.
5. When the activation is complete, close all of the windows.

## Microsoft Security Essentials

Navigate through the Microsoft Security Essentials Setup Wizard. Once it is complete, close all of the windows.

# Getting Started with the Control System

## Acer ES1: Fix Wireless Driver

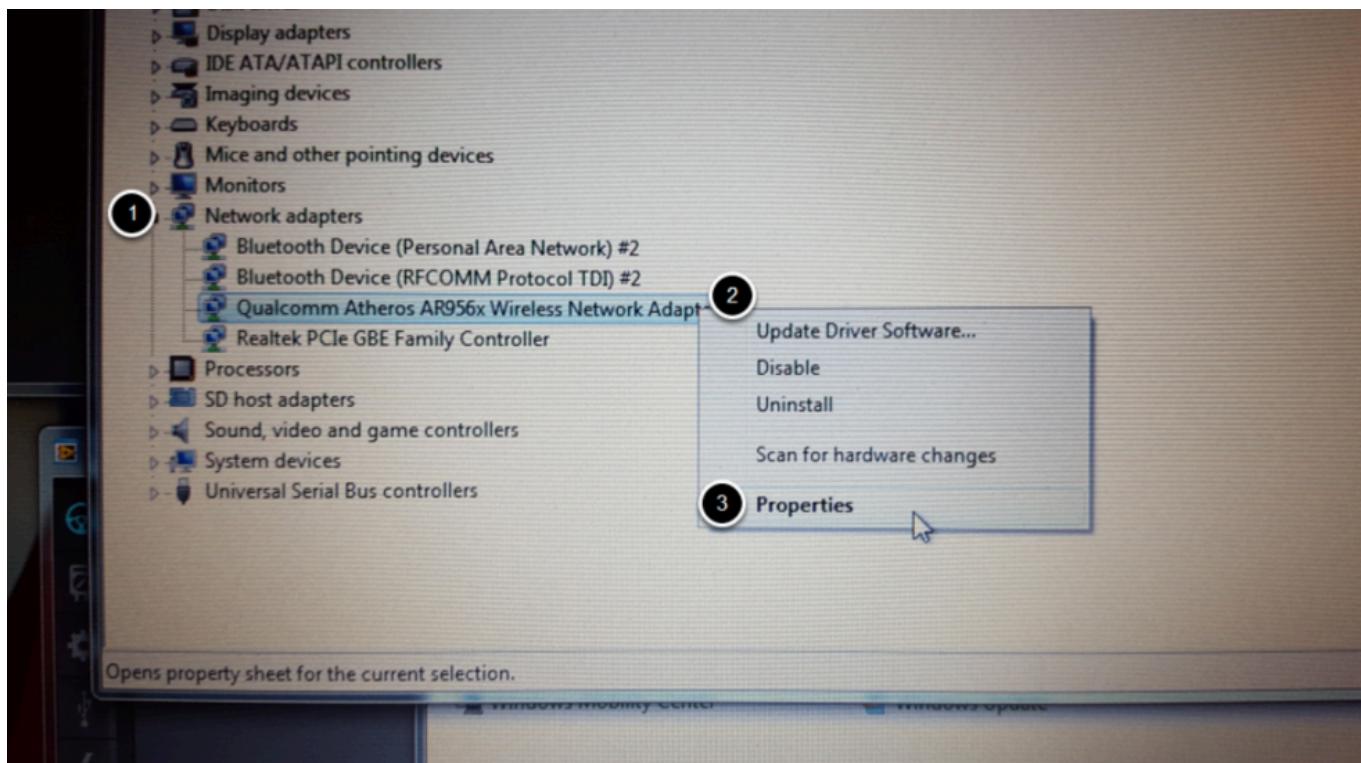


### Acer ES1 PC only!

The default wireless driver in the image may have issues with intermittent communication with the robot radio. The correct driver is in the image, but could not be set to load by default. To load the correct driver, open the Device Manager by clicking start, typing "Device Manager" in the box and clicking Device Manager

# Getting Started with the Control System

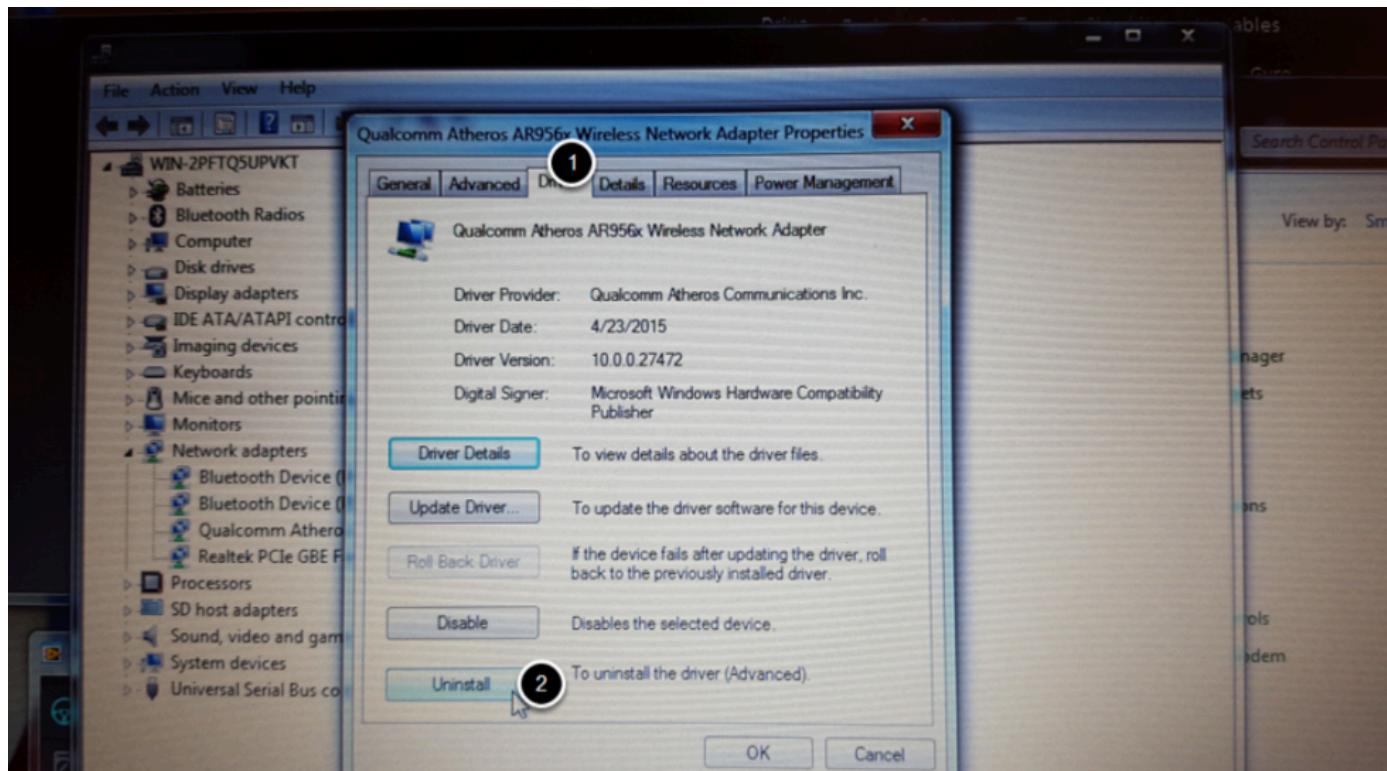
## Open Wireless Device Properties



Click on the arrow next to Network Adapters to expand it and locate the Wireless Network Adapter. Right click the adapter and select Properties.

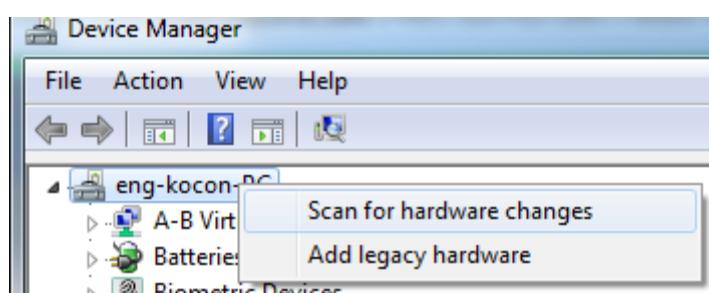
# Getting Started with the Control System

## Uninstall Driver



Click on the Driver tab, then click the Uninstall button. Click Yes at any prompts.

## Scan for New Hardware



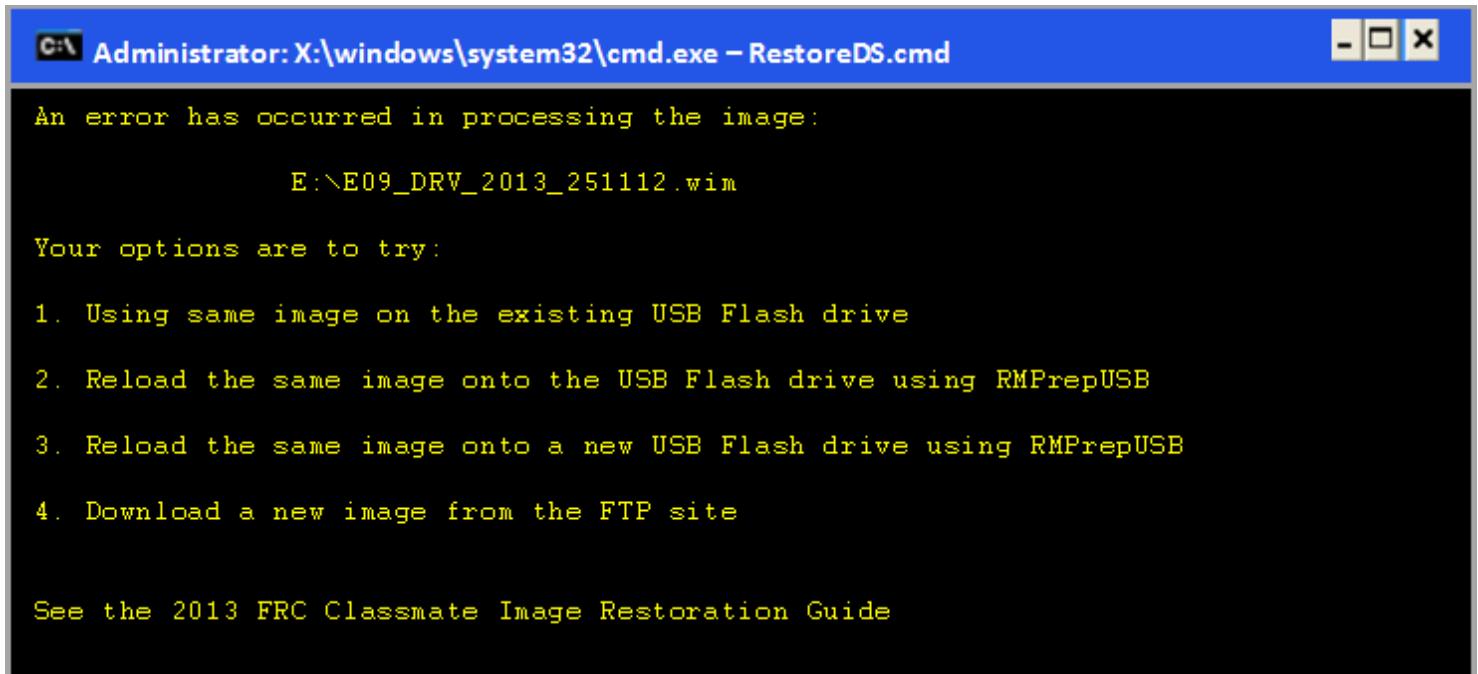
Right click on the top entry of the tree and click "Scan for hardware changes". The wireless adapter should automatically be re-detected and the correct driver should be installed.

# Getting Started with the Control System

## Update Software

In order for the Classmate images to be prepared on time, they are created before the final versions of the software were ready. To use the software for FRC some additional components will need to be installed. LabVIEW teams should continue with the [Installing the FRC Update Suite \(All Languages\)](#) article. C++ or Java teams should continue with [Installing C++ and Java Development Tools for FRC](#).

## Errors during Imaging Process



The screenshot shows a Windows Command Prompt window titled "Administrator: X:\windows\system32\cmd.exe – RestoreDS.cmd". The window contains the following text:

```
C:\ Administrator: X:\windows\system32\cmd.exe – RestoreDS.cmd
An error has occurred in processing the image:
E:\E09_DRV_2013_251112.wim

Your options are to try:
1. Using same image on the existing USB Flash drive
2. Reload the same image onto the USB Flash drive using RMPrepUSB
3. Reload the same image onto a new USB Flash drive using RMPrepUSB
4. Download a new image from the FTP site

See the 2013 FRC Classmate Image Restoration Guide
```

If an error is detected during the imaging process, the following screen will appear. Note that the screenshot below shows the error screen for the Driver Station-only image for the E09. The specific image filename shown will vary depending on the image being applied.

The typical reason for the appearance of this message is due to an error with the USB device on which the image is stored. Each option is listed below with further details as to the actions you can take in pursuing a solution. Pressing any key once this error message is shown will return the user to the menu screen shown in [Image the Classmate](#).

# Getting Started with the Control System

## Option 1

*Using same image on the existing USB Flash drive:* To try this option, press any key to return to the main menu and select #1. This will run the imaging process again.

## Option 2

*Reload the same image onto the USB Flash drive using RMPrepUSB:* It's possible the error message was displayed due to an error caused during the creation of the USB Flash drive (e.g. file copy error, data corruption, etc.) Press any key to return to the main menu and select #4 to safely shutdown the Classmate then follow the steps starting with [RMPrep](#) to create a new USB Restoration Key using the same USB Flash drive.

## Option 3

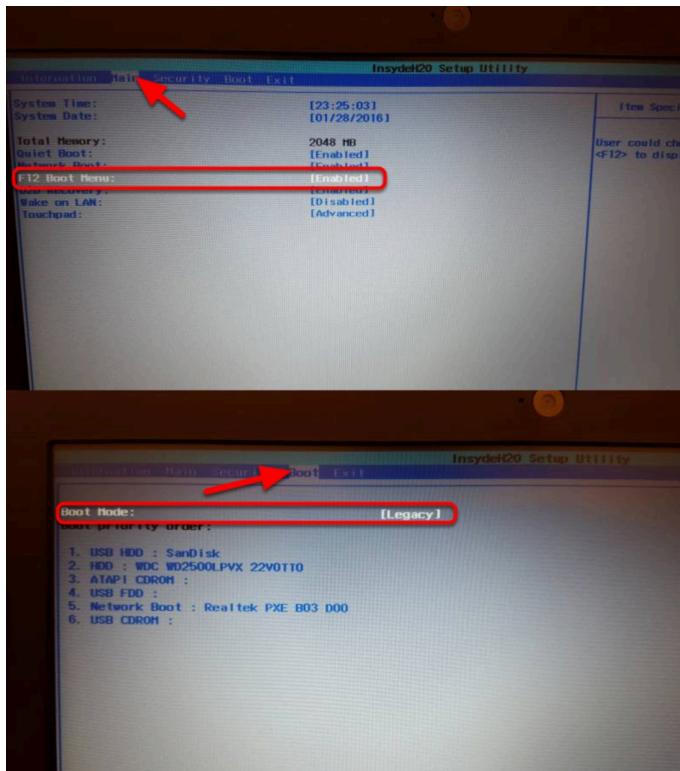
*Reload the same image onto a new USB Flash drive using RMPrepUSB:* The error message displayed may also be caused by an error with the USB Flash drive itself. Press any key to return to the main menu and select #4 to safely shutdown the Classmate. Select a new USB Flash drive and follow the steps starting with [RMPrep](#).

## Option 4

*Download a new image:* An issue with the downloaded image may also cause an error when imaging. Press any key to return to the main menu and select #4 to safely shutdown the Classmate. Starting with [Download the Classmate Image](#) create a new copy of the imaging stick.

# Getting Started with the Control System

## Checking BIOS Settings



If you are having difficulty booting to USB, check the BIOS settings to insure they are correct. To do this:

- Repeatedly tap the F2 key while the computer is booting to enter the BIOS settings
- Once the BIOS settings screen has loaded, use the right and left arrow keys to select the **Main** tab, then check if the line for **F12 Boot Menu** is set to **Enabled**. If it is not, use the Up\Down keys to highlight it, press Enter, use Up\Down to select Enabled and press Enter again.
- Next, use the Left\Right keys to select the **Boot** tab. Make sure that the **Boot Mode** is set to **Legacy**. If it is not, highlight it using Up\Down, press Enter, highlight Legacy and press Enter again. Press Enter to move through any pop-up dialogs you may see.
- Press F10 to save any changes and exit.

## Getting Started with the Control System

# Installing LabVIEW for FRC 2019 (LabVIEW only)



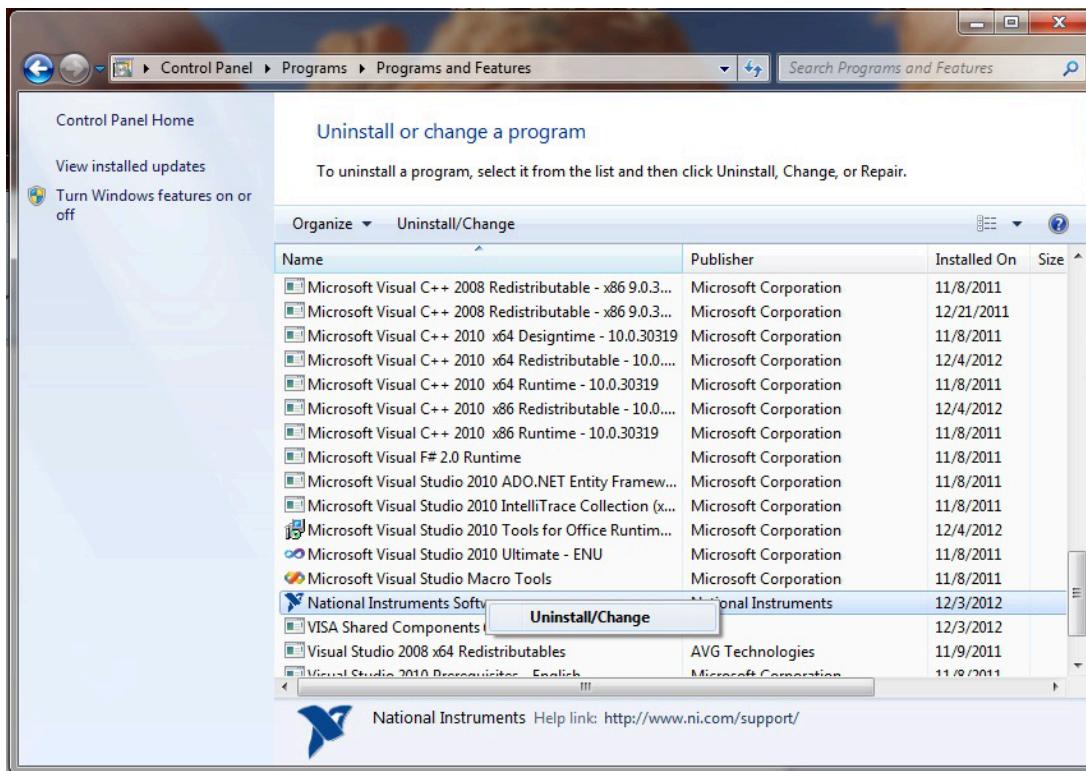
Note: This installation is for teams programming in LabVIEW or using NI Vision Assistant only.

**C++ and Java teams not using these features do not need to install from the DVD.**

**Download and installation times will vary widely with computer and internet connection specifications, however note that this process involves a large file download and installation and will likely take at least an hour to complete.**

# Getting Started with the Control System

## Uninstall Old Versions (Recommended)

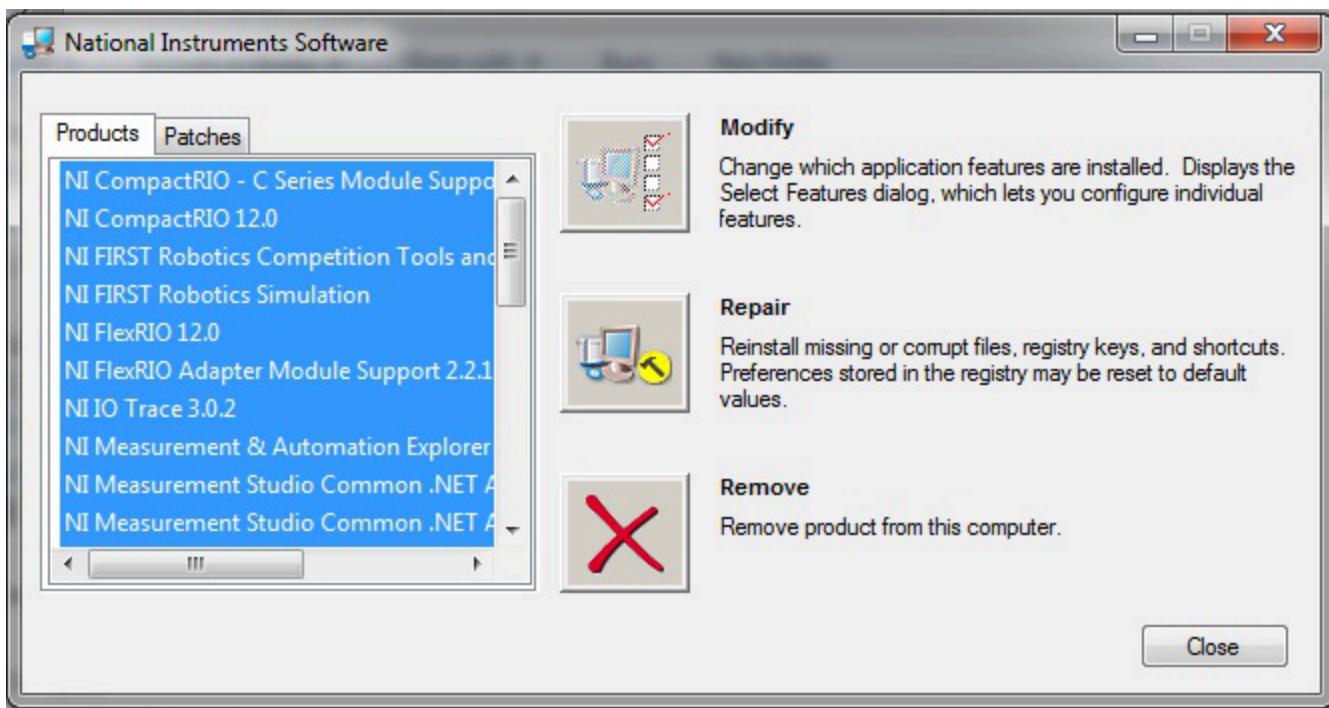


**NOTE: If you wish to keep programming cRIOs you will need to maintain an install of LabVIEW for FRC 2014. The LabVIEW for FRC 2014 license has been extended.** While these versions should be able to co-exist on a single computer, this is not a configuration that has been extensively tested.

Before installing the new version of LabVIEW it is recommended to remove any old versions. The new version will likely co-exist with the old version, but all testing has been done with FRC 2019 only. Make sure to back up any team code located in the "User\LabVIEW Data" directory before un-installing. Then click **Start >> Control Panel >> Uninstall a Program**. Locate the entry labeled "**National Instruments Software**", right-click on it and select **Uninstall/Change**.

# Getting Started with the Control System

## Select Components to Uninstall



In the left pane of the dialog box that appears, **select all entries**. The easiest way to do this is to click the top entry to highlight it, then scroll down to the bottom entry, press and hold shift and click on the last entry then release shift. Click **Remove**. Wait for the uninstaller to complete and reboot if prompted.

## Getting LabVIEW Installer

Either locate and insert the LabVIEW USB Drive or download the LabVIEW 2019 installer from <http://www.ni.com/download/labview-for-frc-18.0/7841/en/>

If downloaded, right click on the downloaded file (NI\_FRC2019.zip) and select Extract All.

**Note: This is a large download (~5GB). It is recommended to use a fast internet connection and to use the NI Downloader to allow the download to resume if interrupted.**

# Getting Started with the Control System

## Installing LabVIEW

National Instruments LabVIEW requires a license. Each season's license is active until January 31st of the following year (e.g. the license for the 2019 season expires on January 31, 2020)

Teams are permitted to install the software on as many team computers as needed, subject to the restrictions and license terms that accompany the applicable software, and provided that only team members or mentors use the software, and solely for FRC. Rights to use LabVIEW are governed solely by the terms of the license agreements that are shown during the installation of the applicable software.

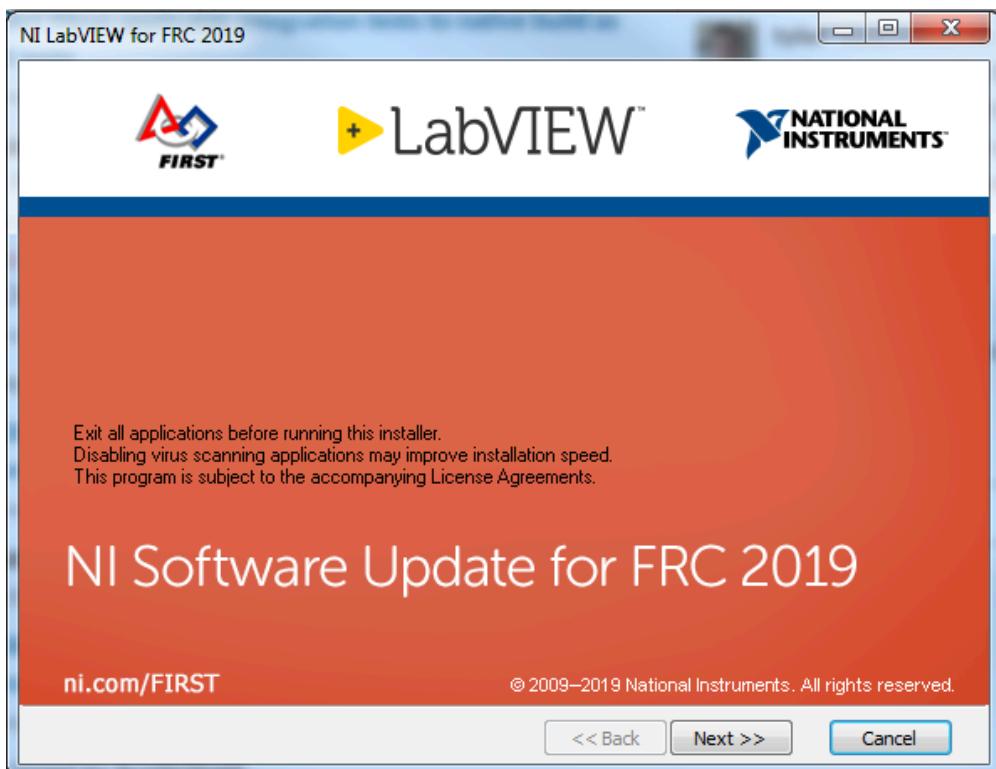
## Welcome



Double click on **autorun.exe** to launch the installer. If prompted to allow changes click Yes. To install LabVIEW to program your FRC robot, click the top option **Install Everything for LabVIEW Development**. To install only NI Vision Assistant for use with C++ or Java, click Install Only NI Vision Development Module. If prompted with any Windows security warnings, click Allow or Yes.

# Getting Started with the Control System

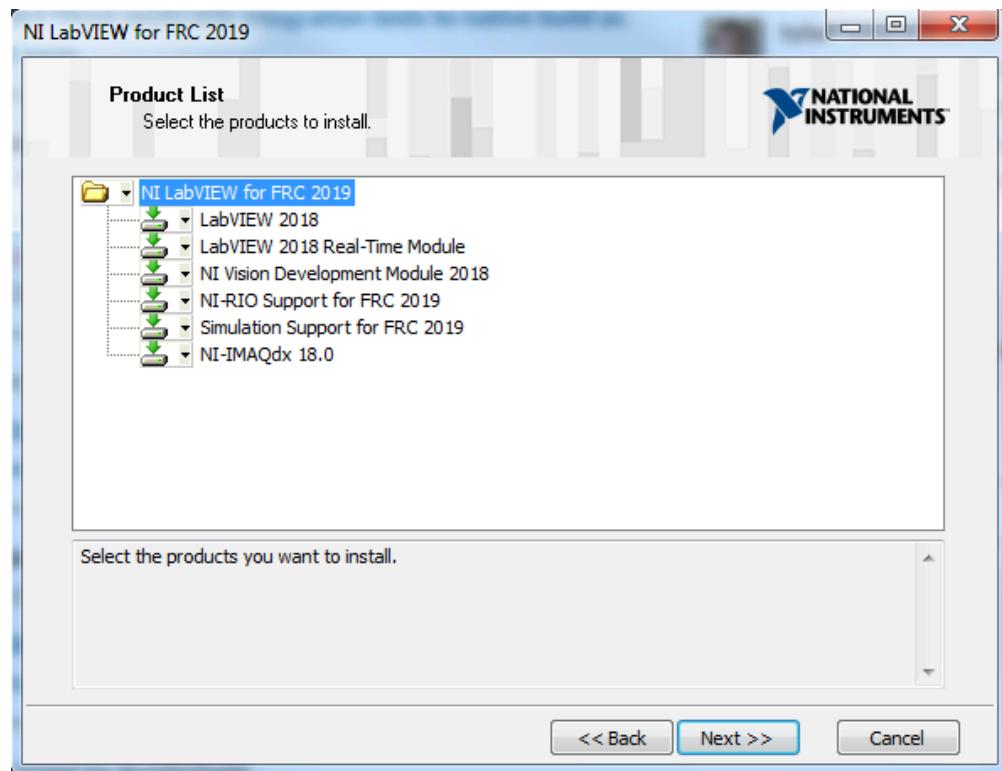
## Warnings



Click "Next"

# Getting Started with the Control System

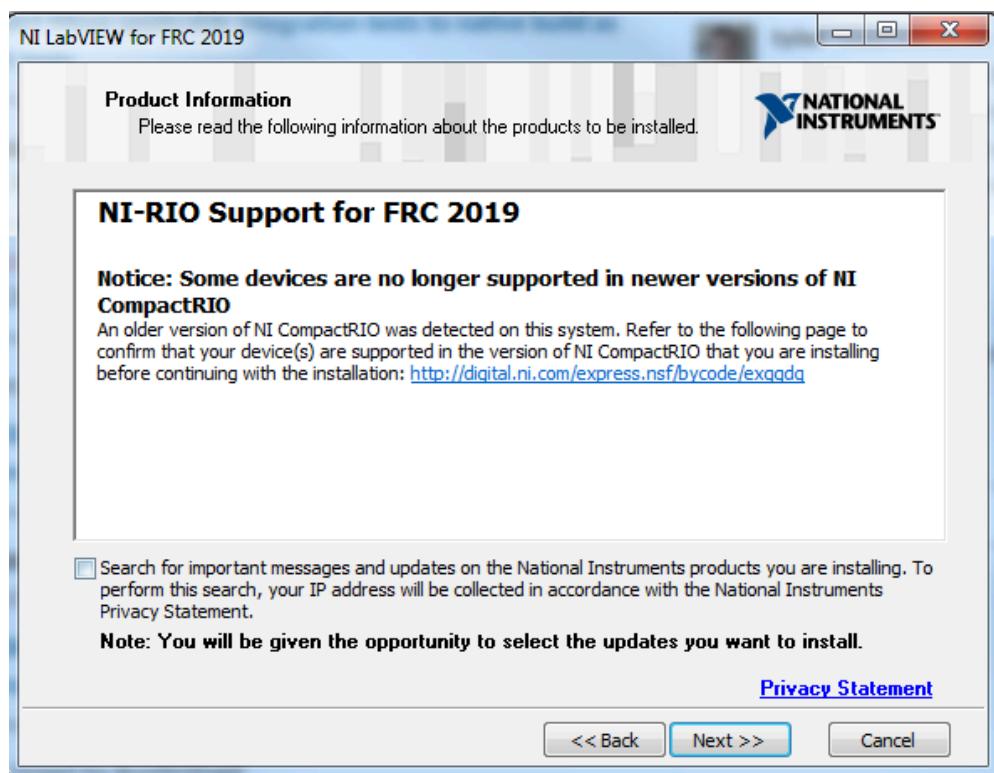
## Product List



Click "Next"

# Getting Started with the Control System

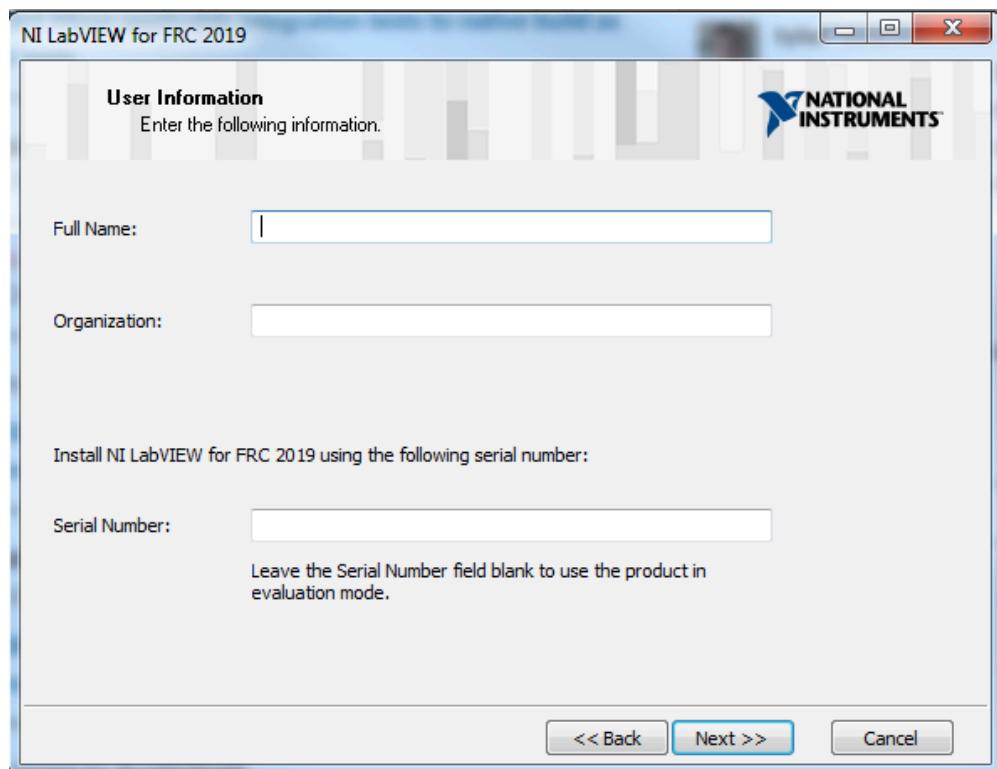
## Product Information



Un-check the box, then click "Next". (Note: you may not see the warning shown in the top portion of the window in this picture).

# Getting Started with the Control System

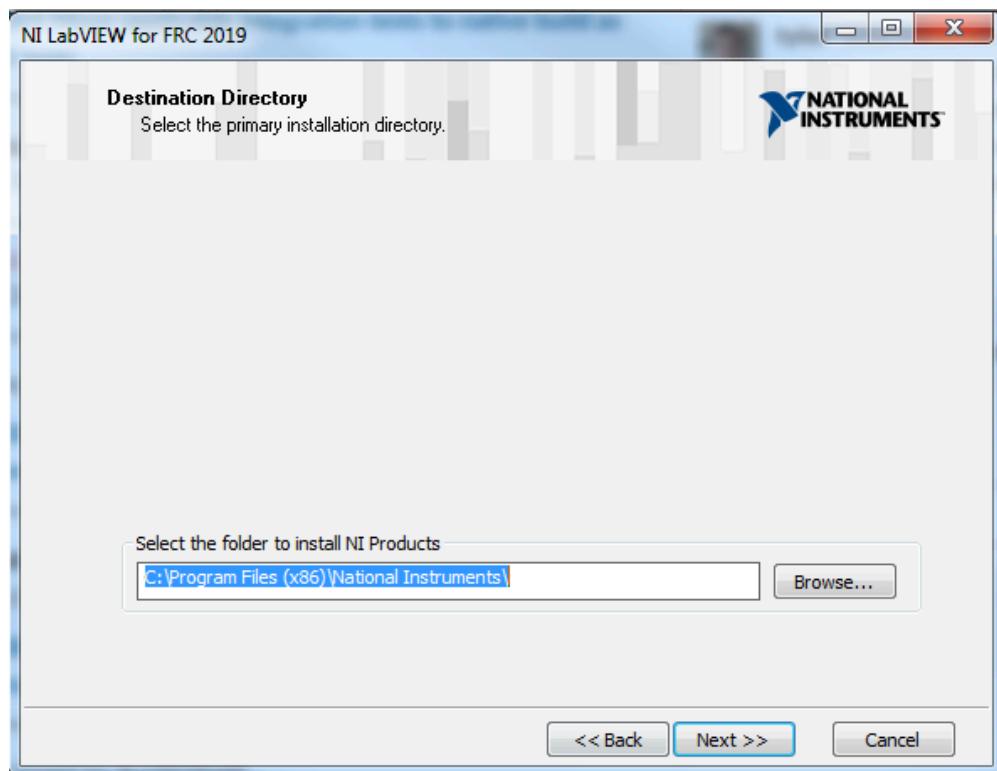
## User Information



Enter name, organization, and the serial number from the **ReadMe in the File Releases on Teamforge**. Click "Next". If you cannot find your serial number, please reach out to National Instruments at [www.ni.com/frc/needhelp](http://www.ni.com/frc/needhelp).

# Getting Started with the Control System

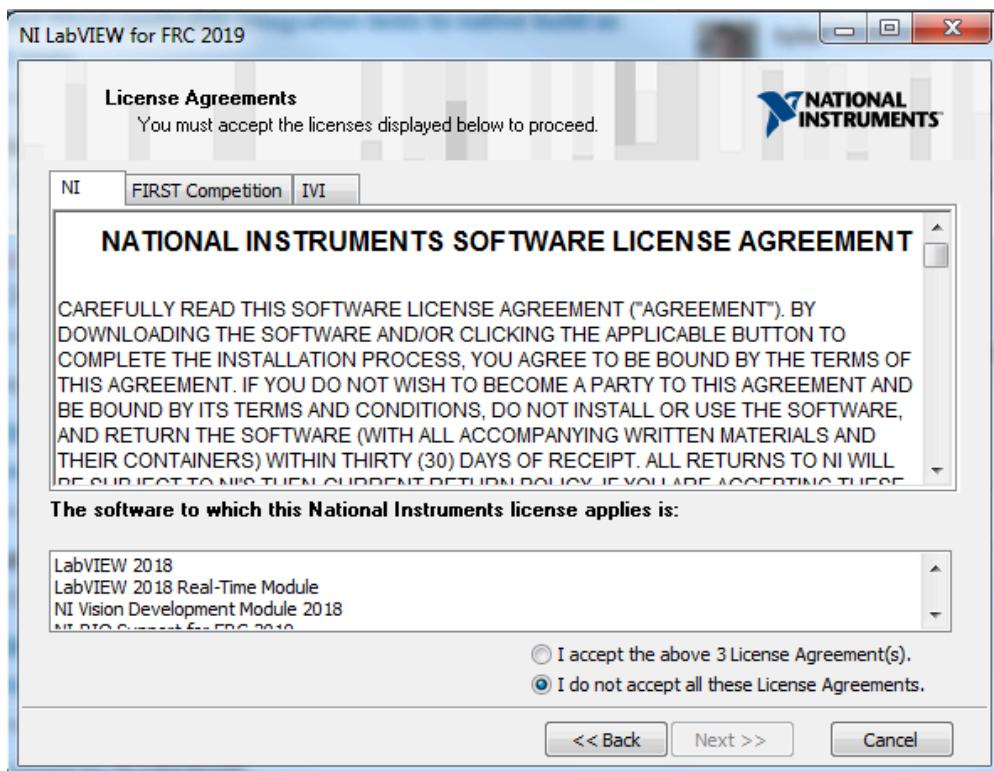
## Destination Directory



Click "Next"

# Getting Started with the Control System

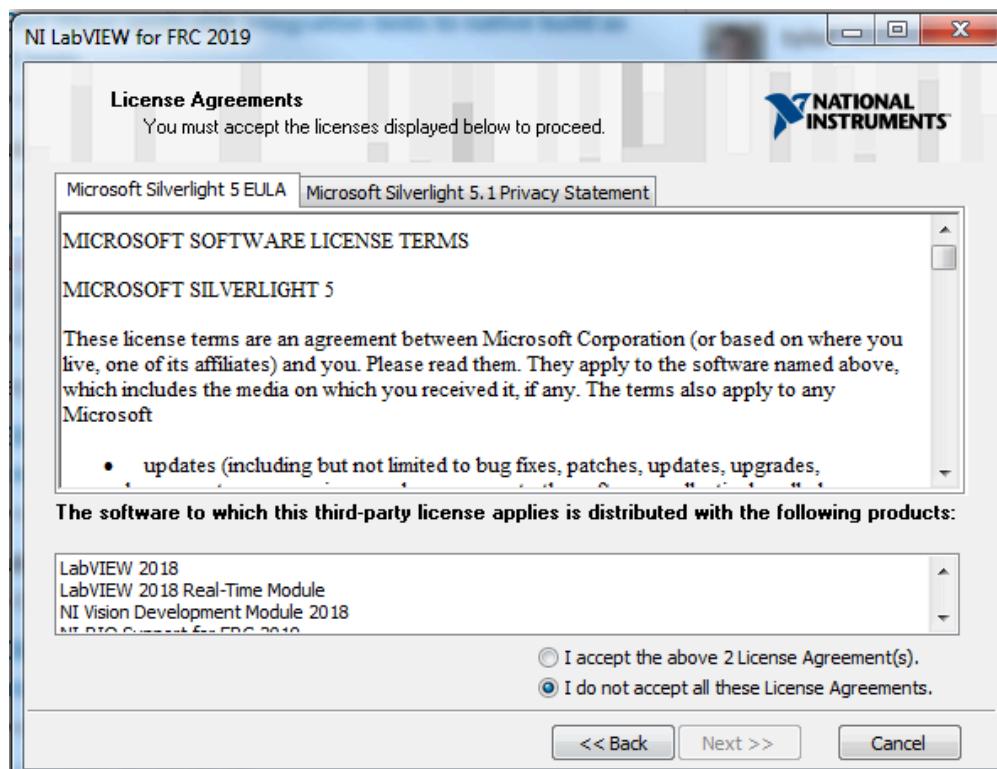
## License Agreements (1)



Check "I accept..." then Click "Next"

# Getting Started with the Control System

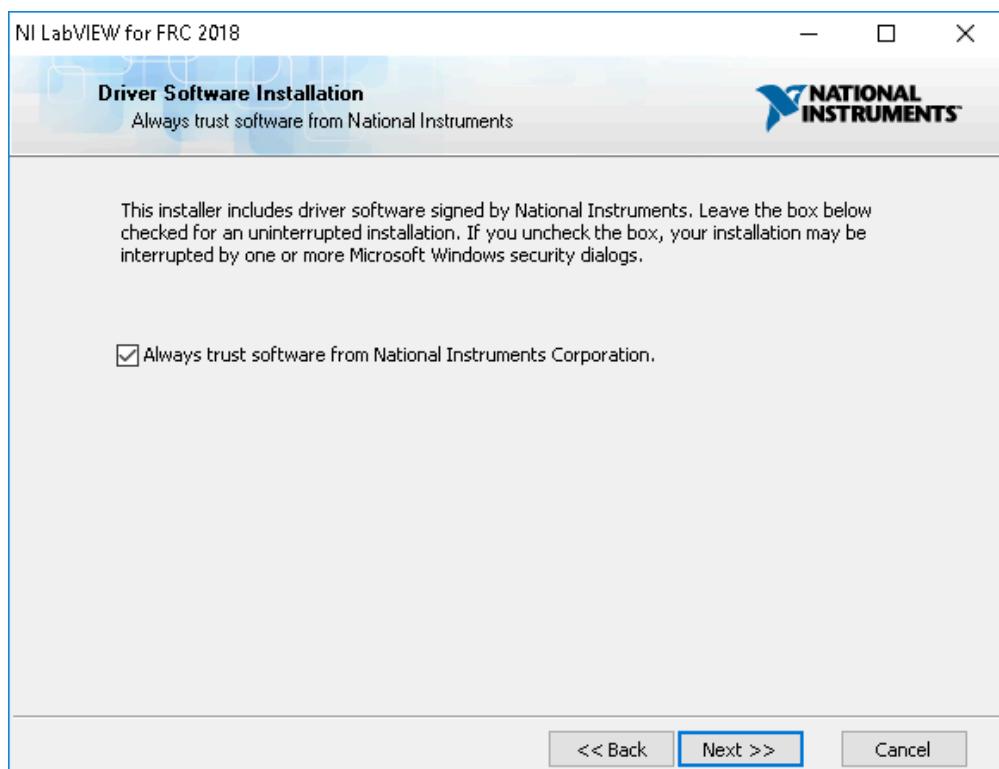
## License Agreements (2)



Check "I accept..." then Click "Next"

# Getting Started with the Control System

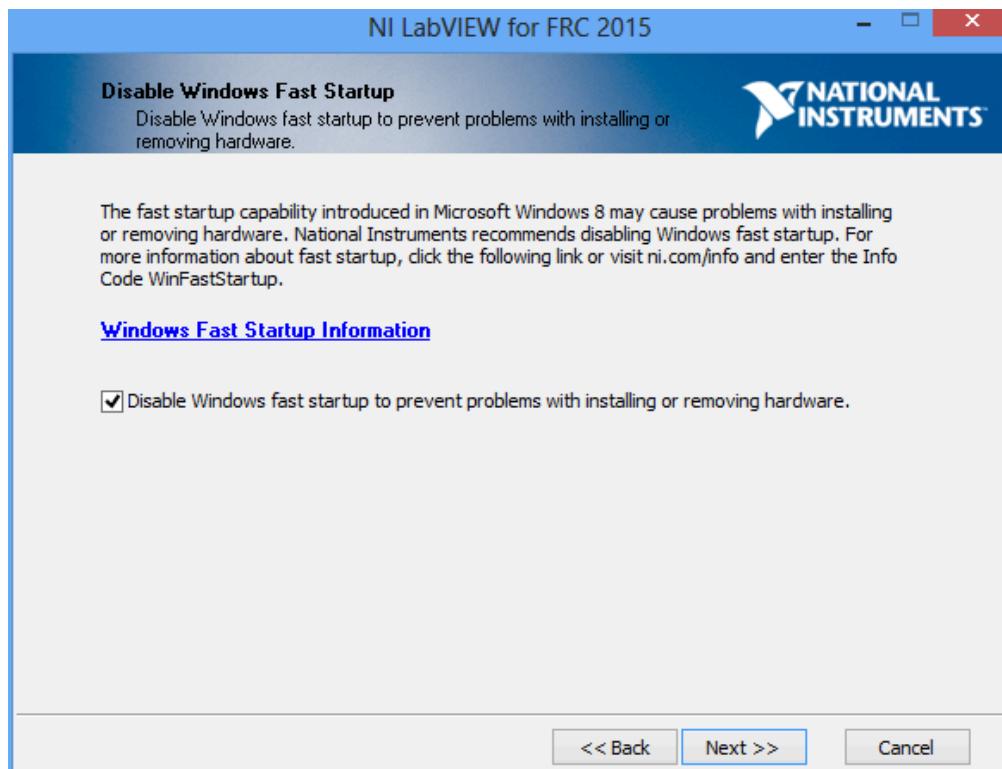
## Driver Software Installation



If you see this screen, Click "Next"

# Getting Started with the Control System

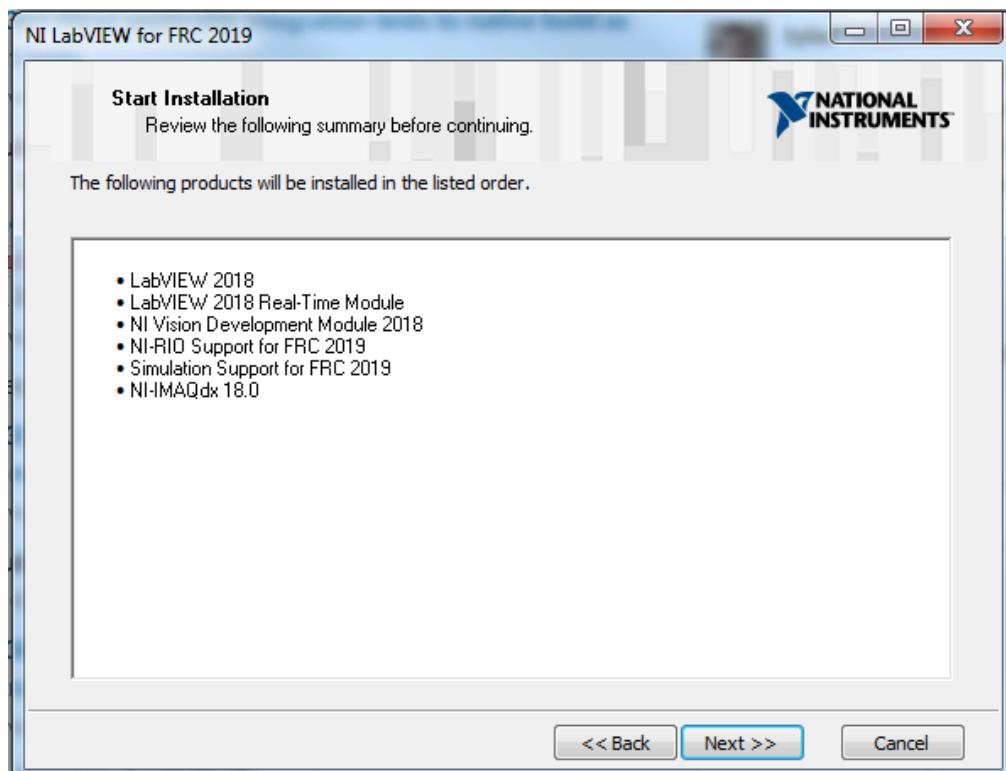
## Disable Windows Fast Startup



If you see this screen, click "Next"

# Getting Started with the Control System

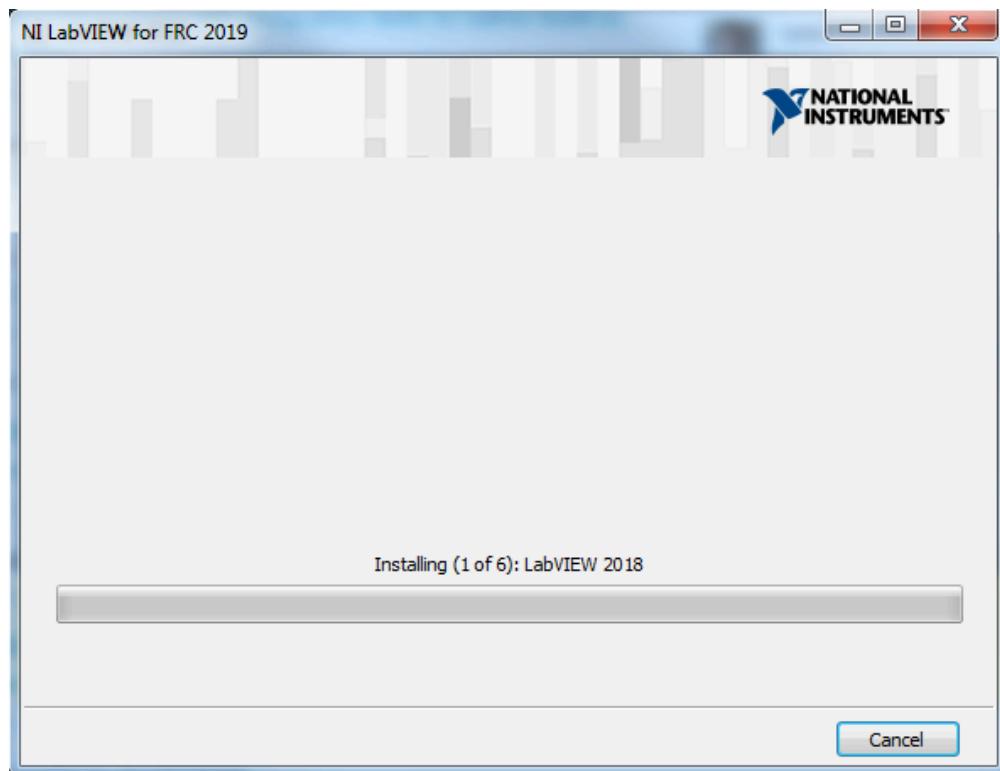
## Start Installation



Click "Next"

# Getting Started with the Control System

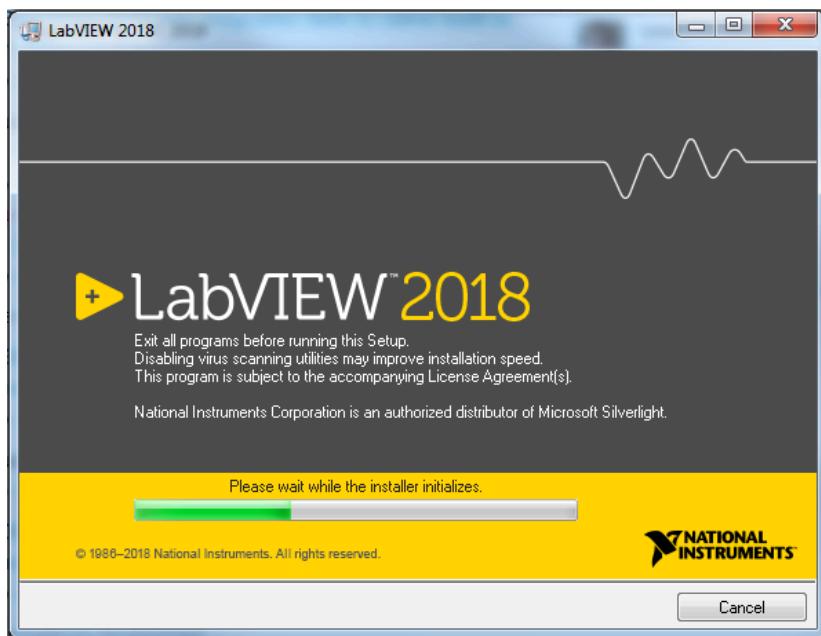
## Overall Progress



Overall installation progress will be tracked in this window

# Getting Started with the Control System

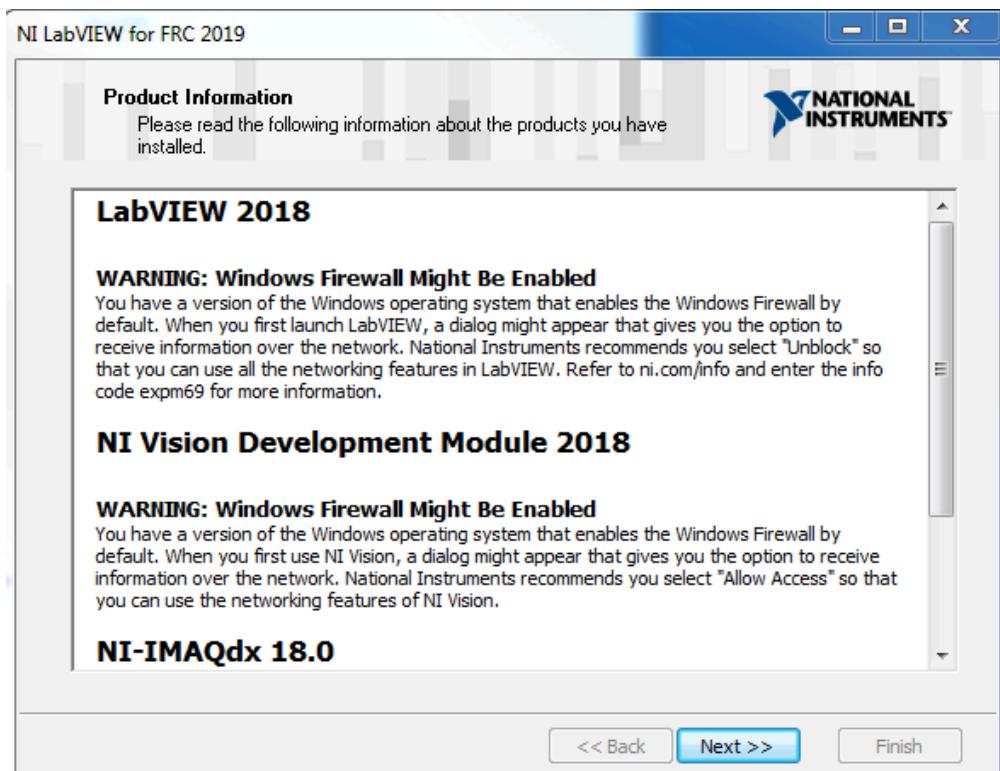
## Individual Product Progress



Each product installed will also create an individual progress window like the one shown above.

# Getting Started with the Control System

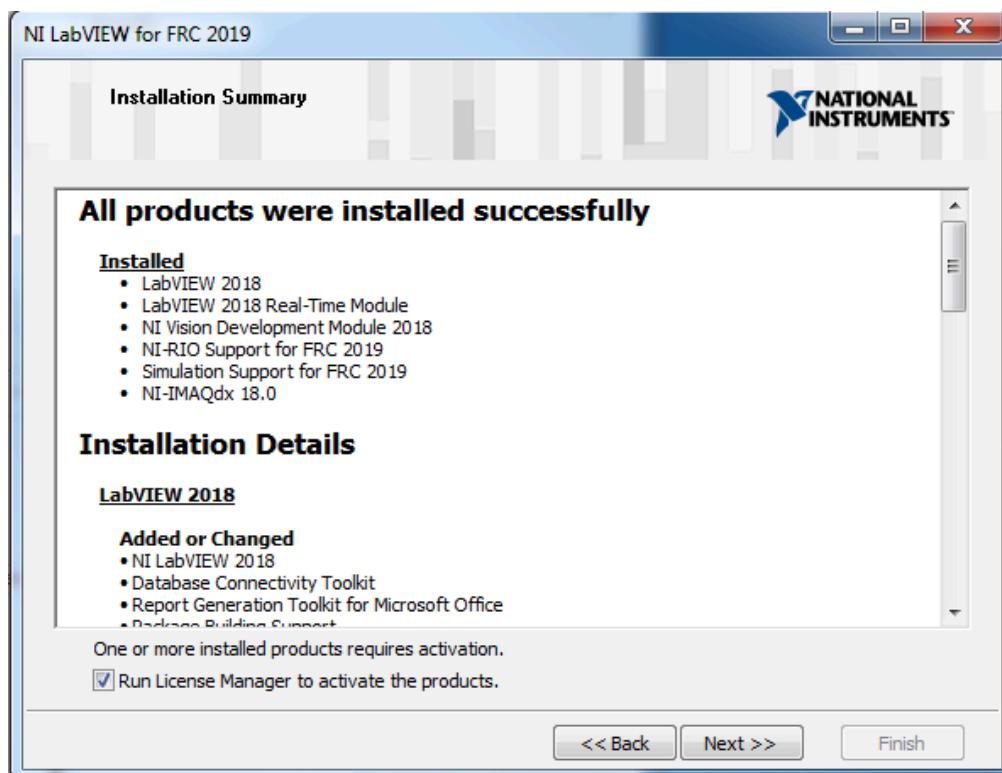
## Product Information



Click "Next"

# Getting Started with the Control System

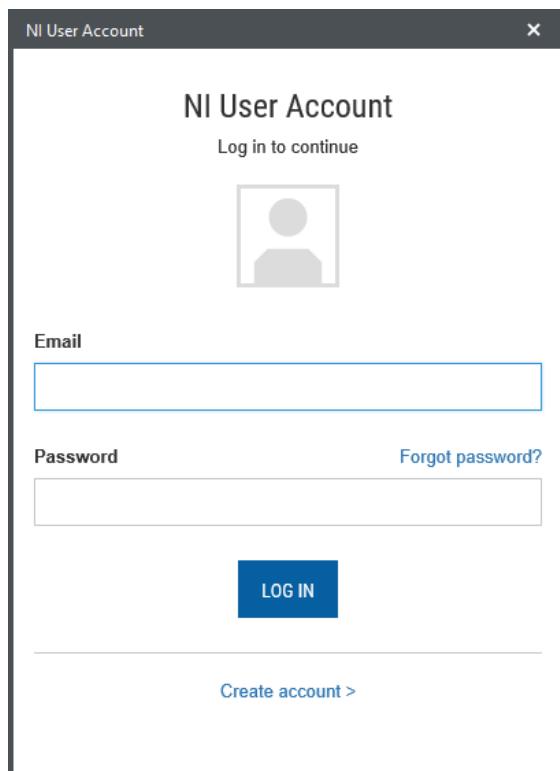
## Installation Summary



If internet access is available and you are ready to activate, click "Next"; otherwise uncheck the "Run License Manager..." and click "Next".

# Getting Started with the Control System

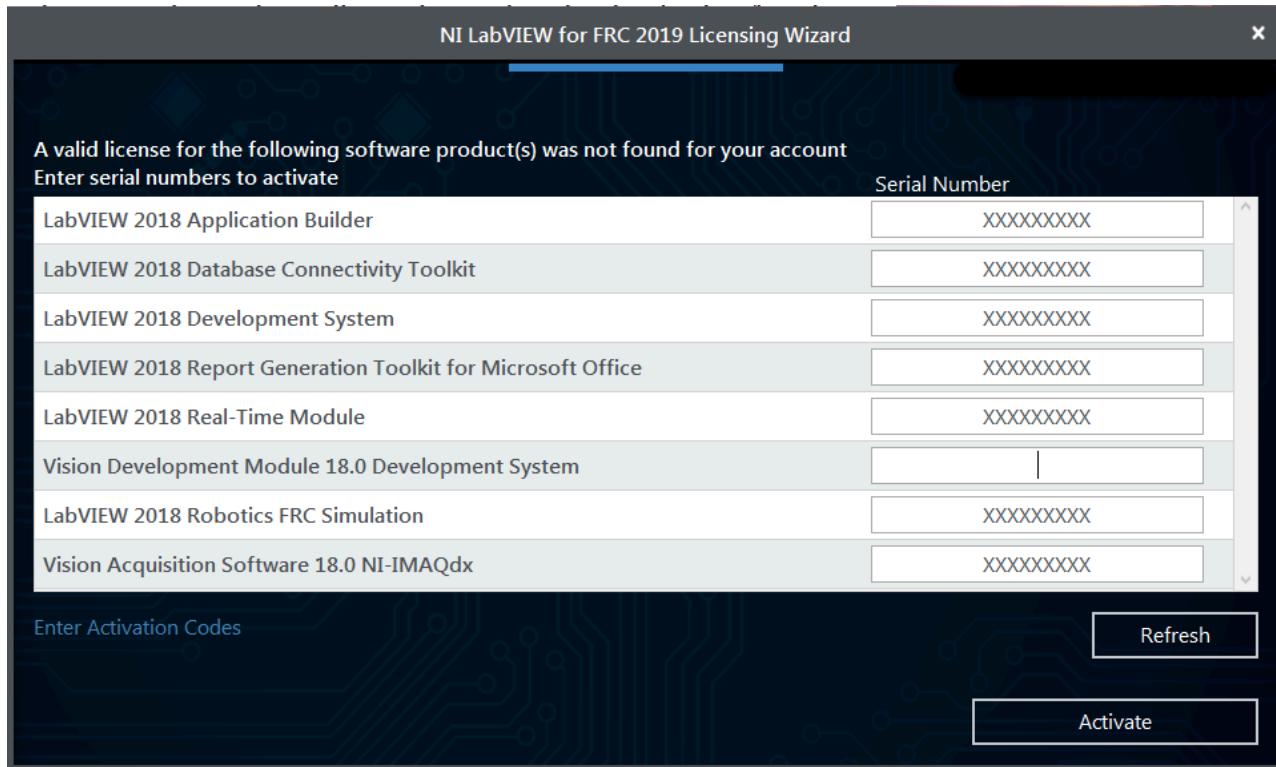
## NI Activation Wizard



Log into your ni.com account. If you don't have an account, select 'Create account' to create a free account.

# Getting Started with the Control System

## NI Activation Wizard (2)

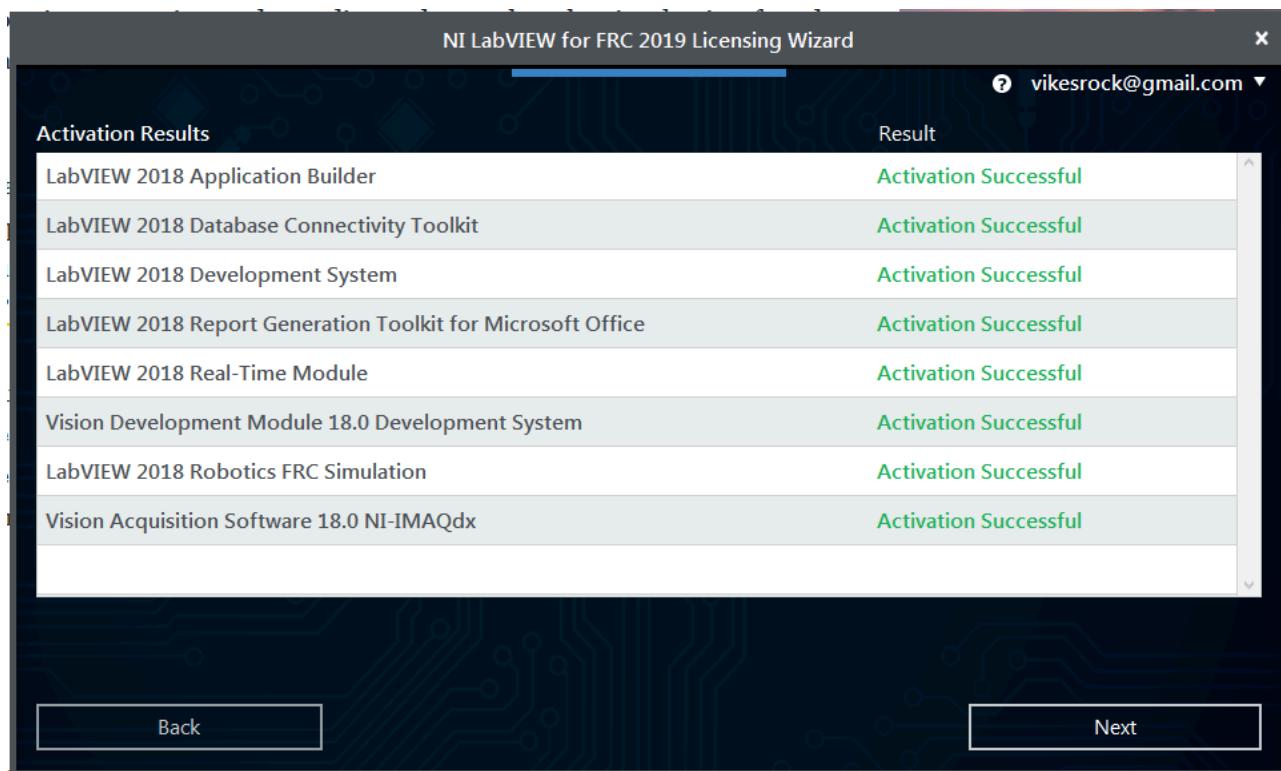


The serial number you entered at the "User Information" screen should appear in all of the text boxes, if it doesn't, enter it now. Click "Activate".

Note: If this is the first time activating the 2019 software on this account, you will see the message shown above about a valid license not being found. You can ignore this.

# Getting Started with the Control System

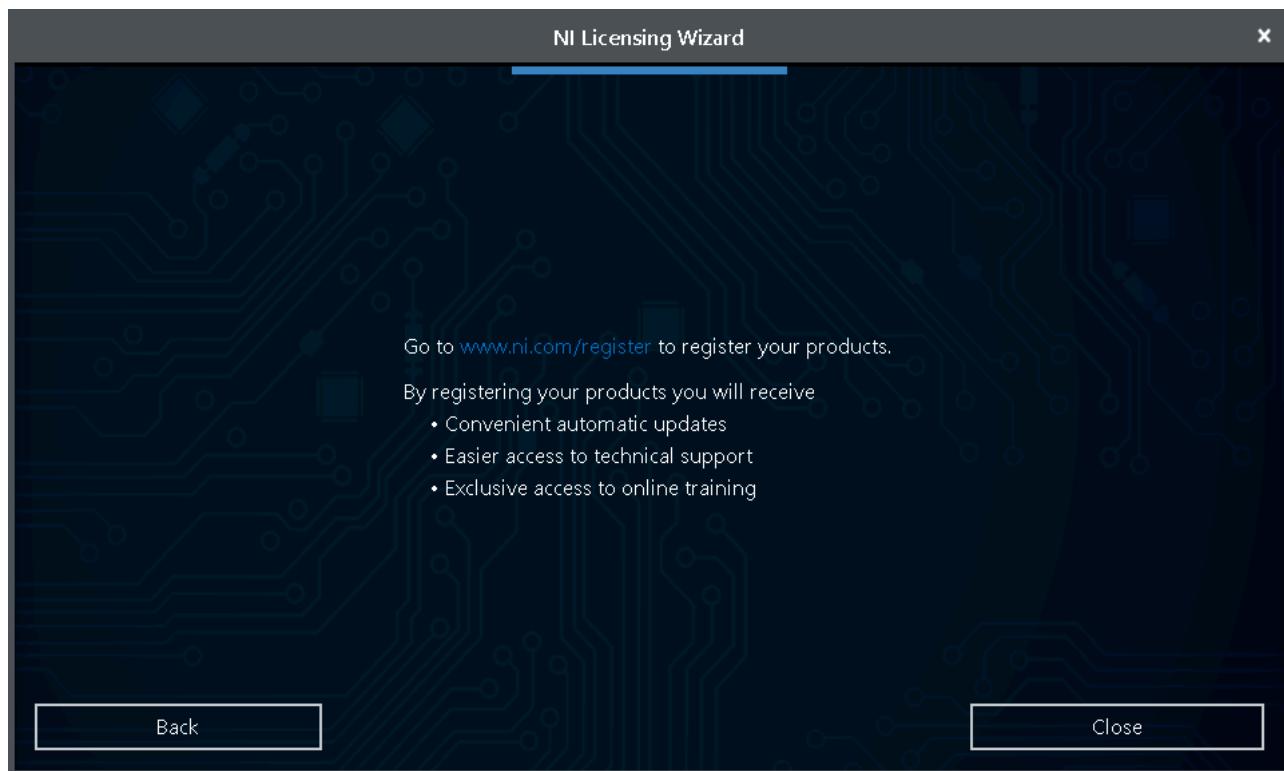
## NI Activation Wizard (3)



If your products activate successfully, an “Activation Successful” message will appear. If the serial number was incorrect, it will give you a text box and you can re-enter the number and select “Try Again”. If everything activated successfully, click “Next”.

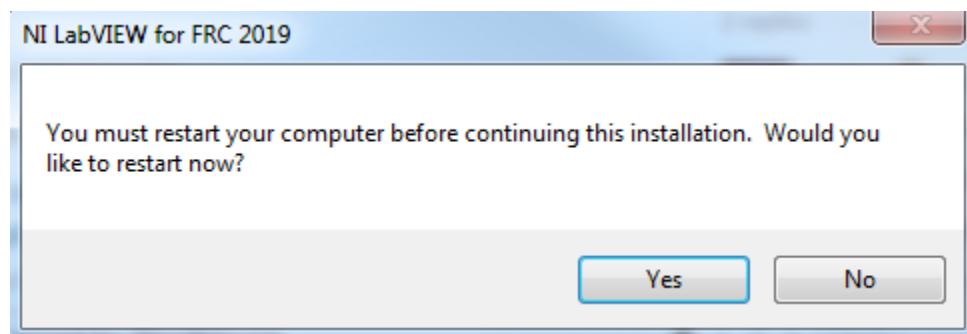
# Getting Started with the Control System

## NI Activation Wizard (4)



Click "Close".

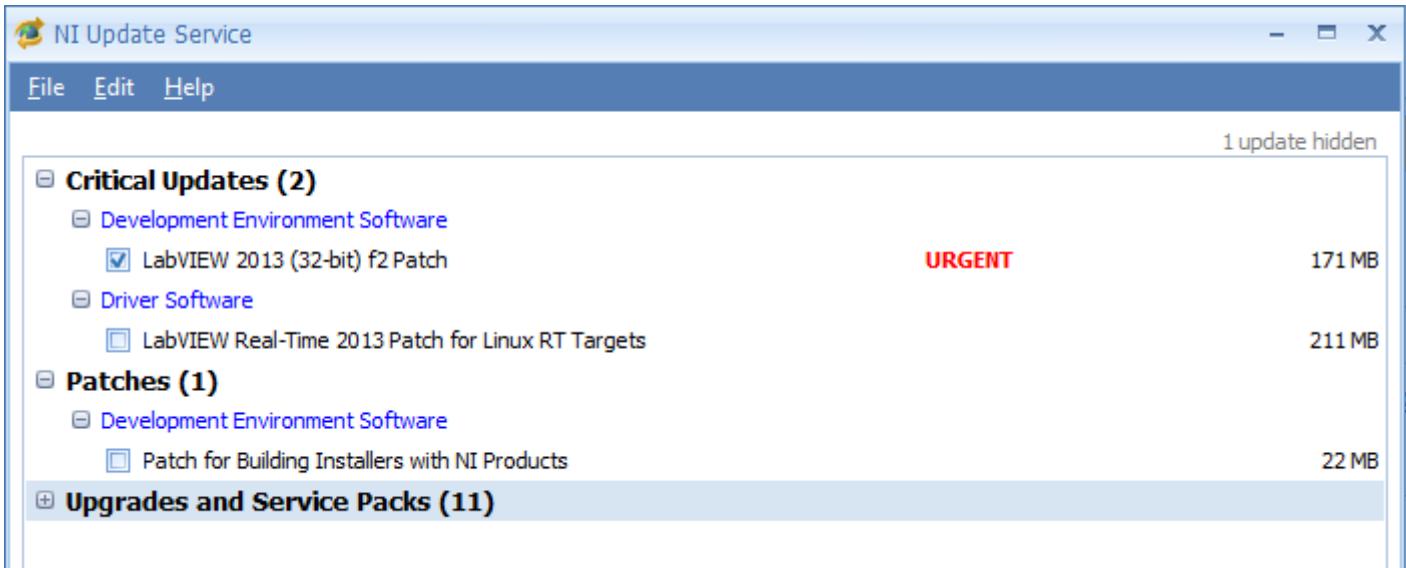
## Restart Message



Select "Yes"

# Getting Started with the Control System

## NI Update Service



On occasion you may see alerts from the NI Update Service about patches to LabVIEW. **It is not recommended to install these updates unless directed by FRC through our usual communication channels (Frank's Blog, Team Updates or E-mail Blasts).**

# Installing the FRC Update Suite (All Languages)



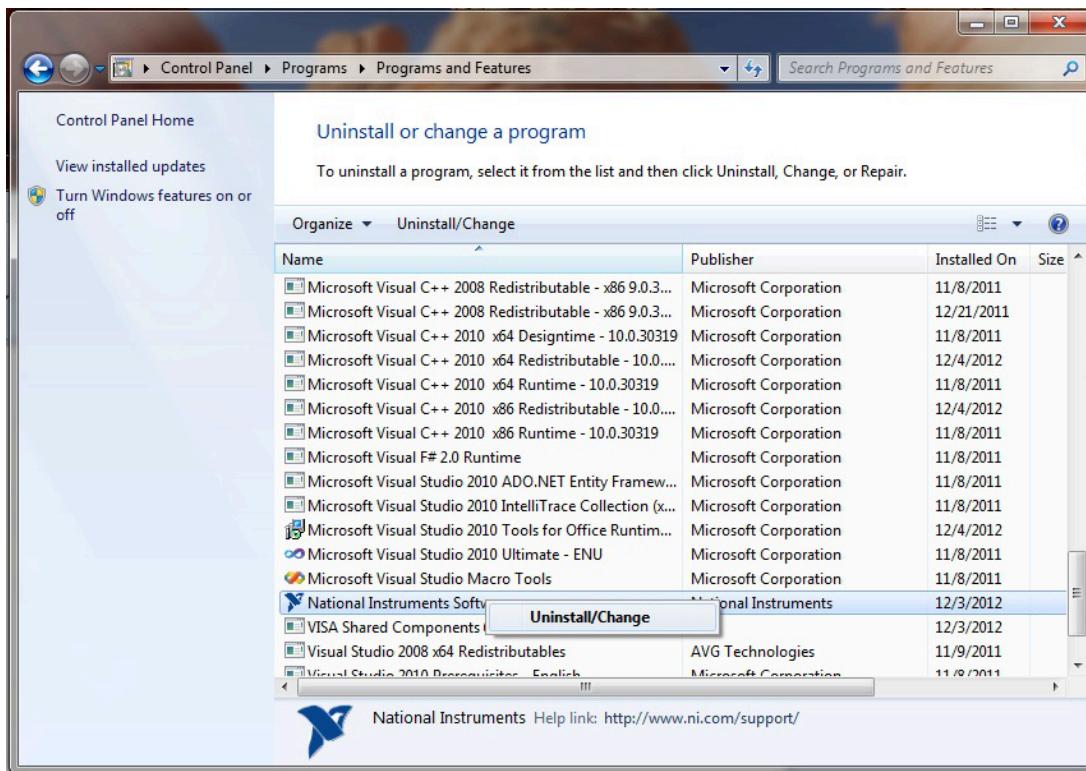
The FRC Update Suite contains the following software components: LabVIEW Update, FRC Driver Station, and FRC Utilities. If an FRC LabVIEW installation is found, the LabVIEW Update will be installed or updated, otherwise this step will be skipped. The FRC Driver Station and FRC Utilities will always be installed or updated. The LabVIEW runtime components required for the driver station and utilities is included in this package. **No components from the LabVIEW Merged Suite are required for running either the Driver Station or Utilities.**

C++ and Java teams wishing to use NI Vision Assistant should run the full Suite installer as described in the article - [Installing LabVIEW for FRC \(LabVIEW only\)](#)

**!** Note: The Driver Station will only work on Windows 7, Windows 8, Windows 8.1, and Windows 10. It will not work on Windows XP.

# Getting Started with the Control System

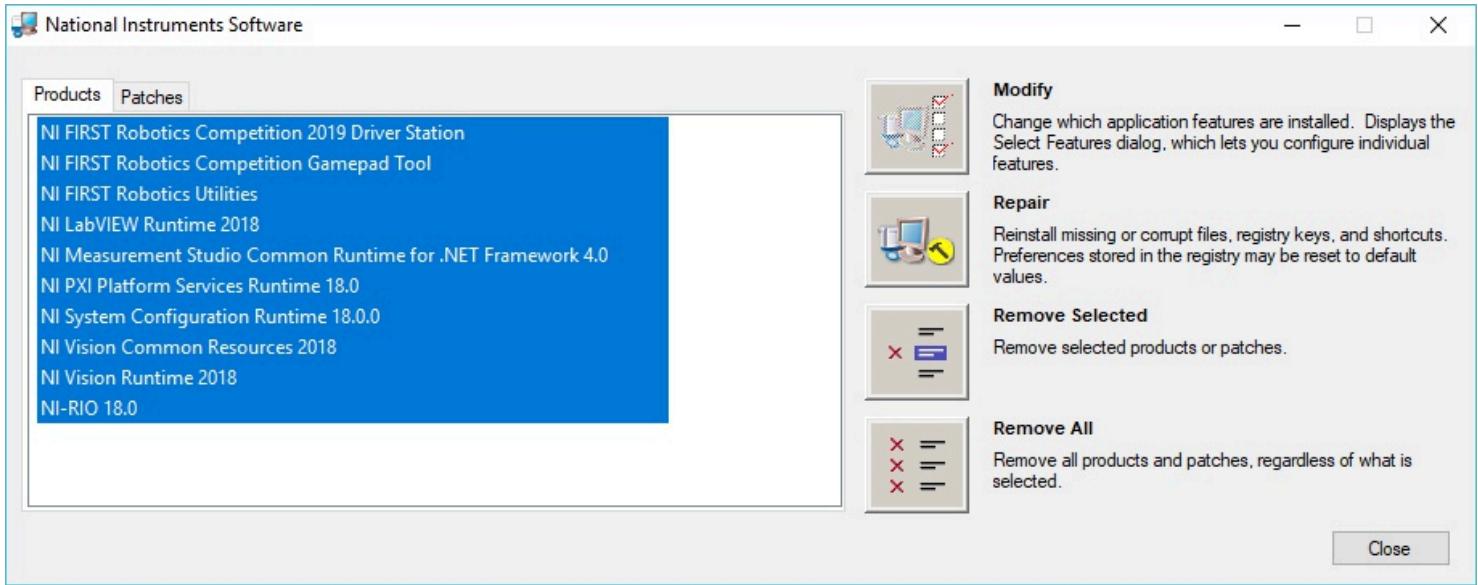
## Uninstall Old Versions (Recommended)



**LabVIEW teams have already completed this step, do not repeat it.** Before installing the new version of the NI Update it is recommended to remove any old versions. The new version will likely properly overwrite the old version, but all testing has been done with FRC 2019 only. Make sure to back up any team code located in the "User\LabVIEW Data" directory before un-installing. Then click **Start >> Control Panel >> Uninstall a Program**. Locate the entry labeled "**National Instruments Software**", right-click on it and select **Uninstall/Change**.

# Getting Started with the Control System

## Select Components to Uninstall



Click **Remove All** and follow any prompts to remove all previous NI products.

## Downloading the Update

Download the update from <http://www.ni.com/download/first-robotics-software-2017/7904/en/>

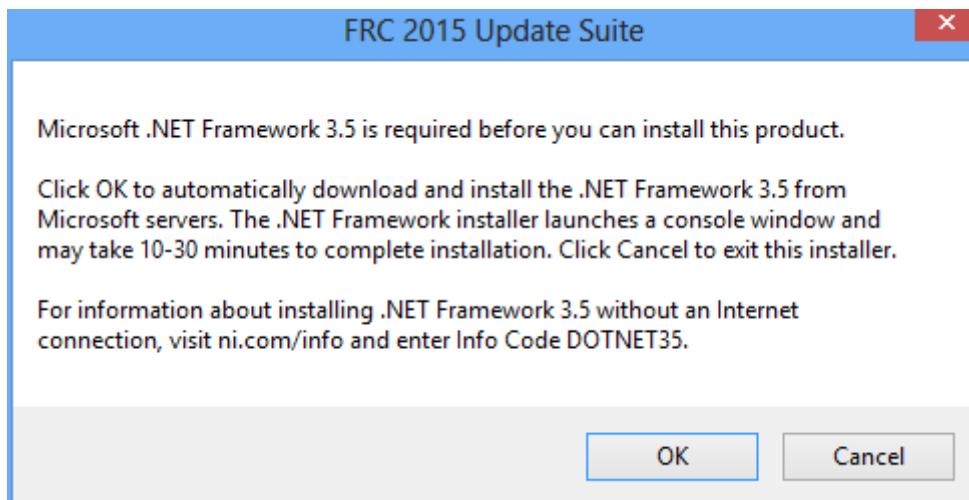
**⚠ Note:** This download will require the decryption key from the Kickoff broadcast. The key is  
**\$Robots&in#SPACE!!**

## .NET Framework 4.6.2

The Update installer may prompt that .NET Framework 4.6.2 needs to be updated or installed. Follow prompts on-screen to complete the installation, including rebooting if requested. Then resume the installation of the NI FRC Update, restarting the installer if necessary.

# Getting Started with the Control System

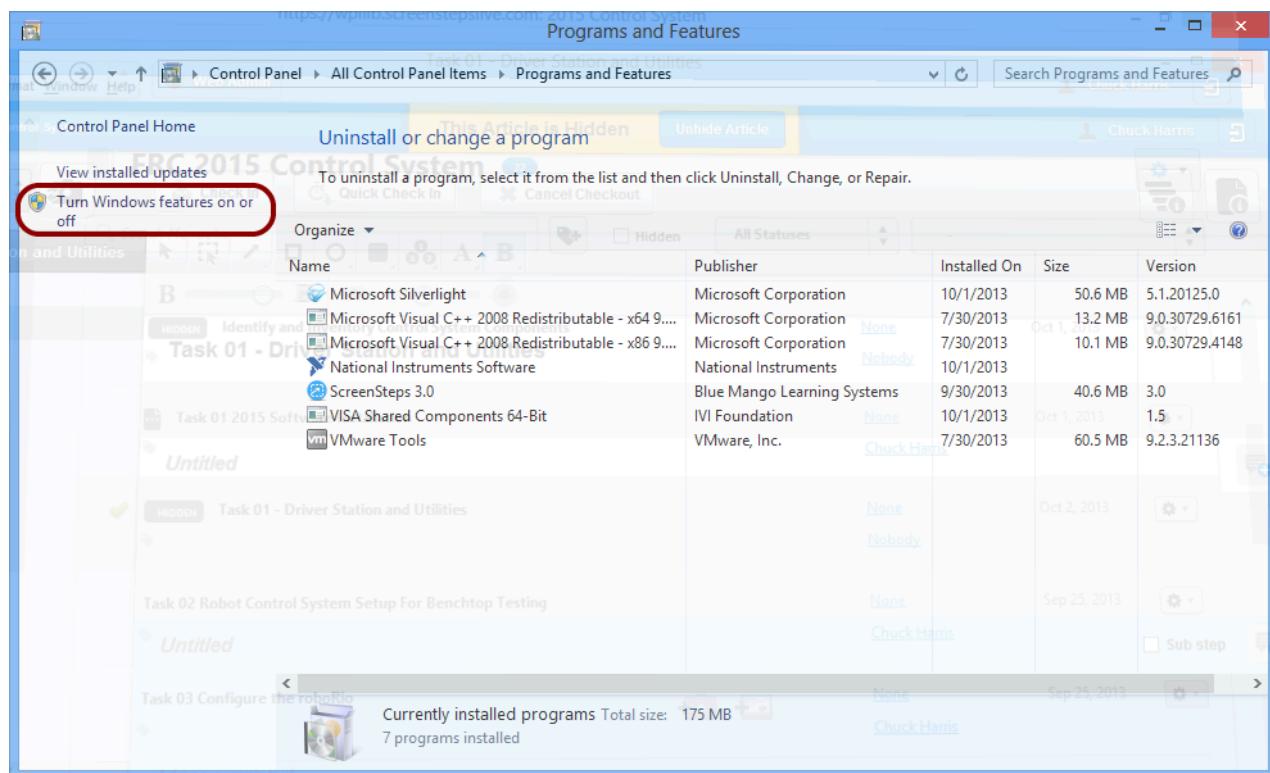
## .NET Framework 3.5



If installing on Windows 8 or 10, the Microsoft .NET Framework 3.5 may need to be installed. If you see the dialog shown above, click "Cancel" and perform the steps shown below. An internet connection is required to complete these steps.

# Getting Started with the Control System

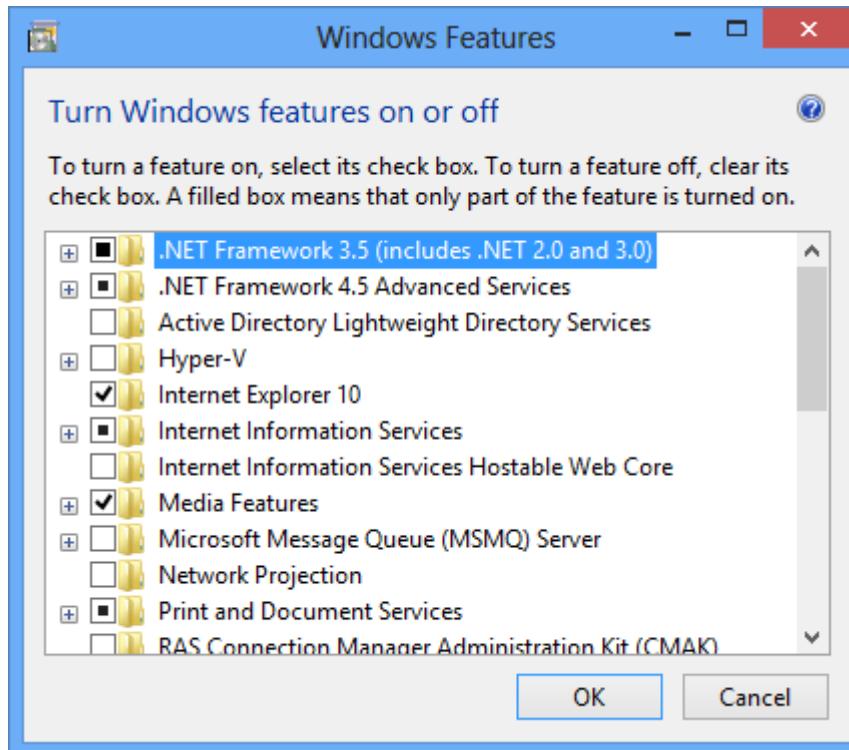
## Programs and Features



Open the "Programs and Features" window from the control panel and click on "Turn Windows features on or off"

# Getting Started with the Control System

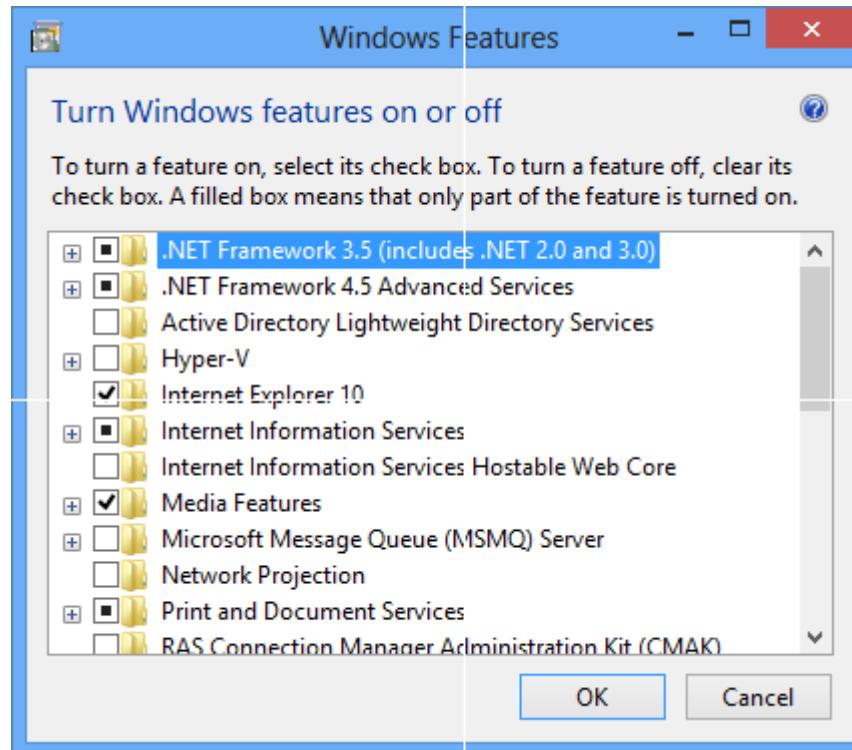
## Windows Features (.NET Framework 3.5 not on)



Select ".NET Framework 3.5 (includes .NET 2.0 and 3.0)" to enable it (a black dot, not a check box will appear) and then click "OK". When installation finishes [restart installation of FRC 2019 Update Suite](#).

# Getting Started with the Control System

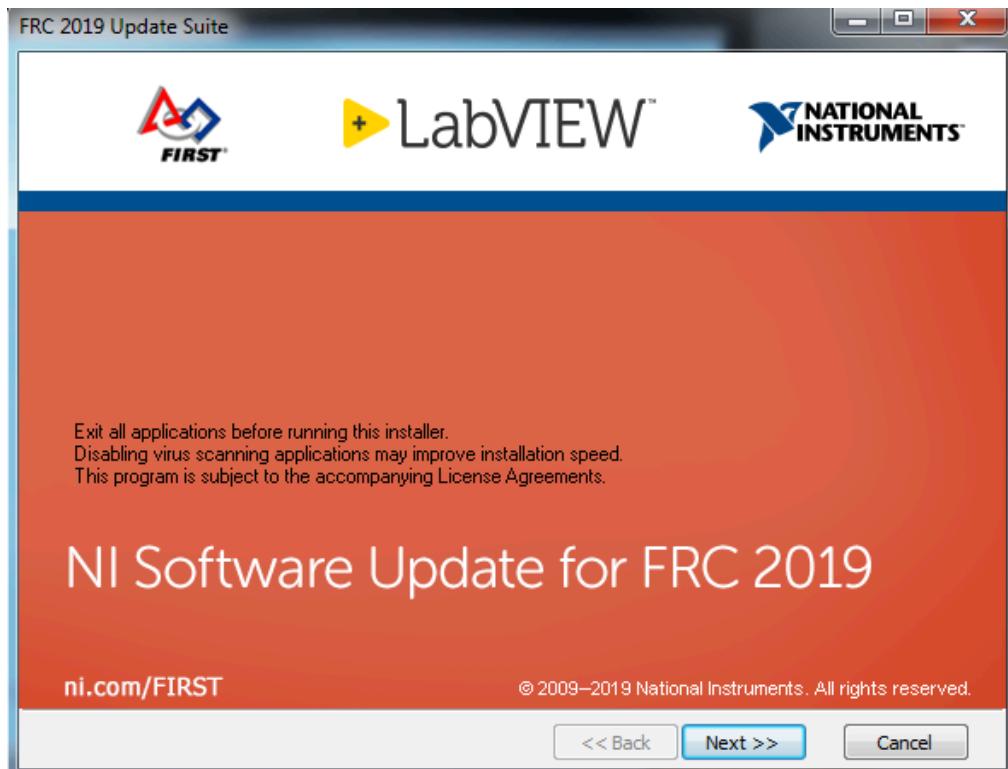
## Windows Features (.NET Framework 3.5 already on)



If a black dot is shown next to ".NET Framework 3.5" the feature is already on. Click "Cancel" and [restart installation of FRC 2019 Update Suite](#).

# Getting Started with the Control System

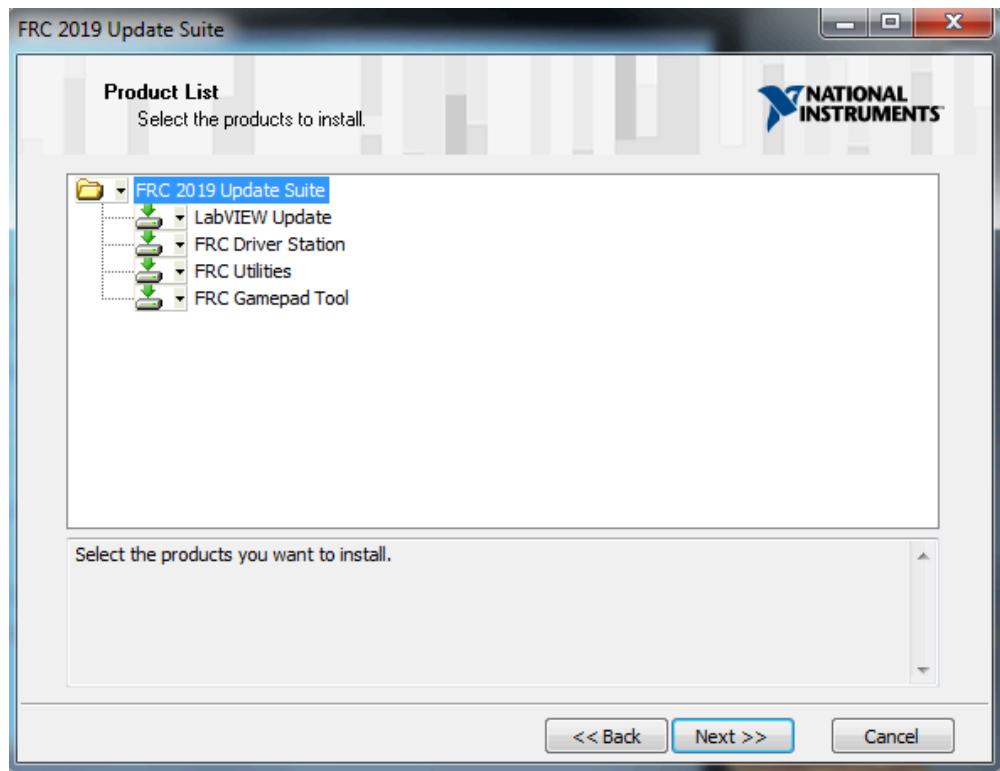
## Welcome



Right click on the downloaded zip file and select Extract All. If you downloaded the encrypted zip file, you will be prompted for the encryption key which will be released at Kickoff. Open the extracted folder and any subfolders until you reach the folder containing "setup" (may say "setup.exe" on some machines). Double click on the setup icon to launch the installer. Click "Yes" if a Windows Security prompt appears. Click "Next" on the splash screen that appears.

# Getting Started with the Control System

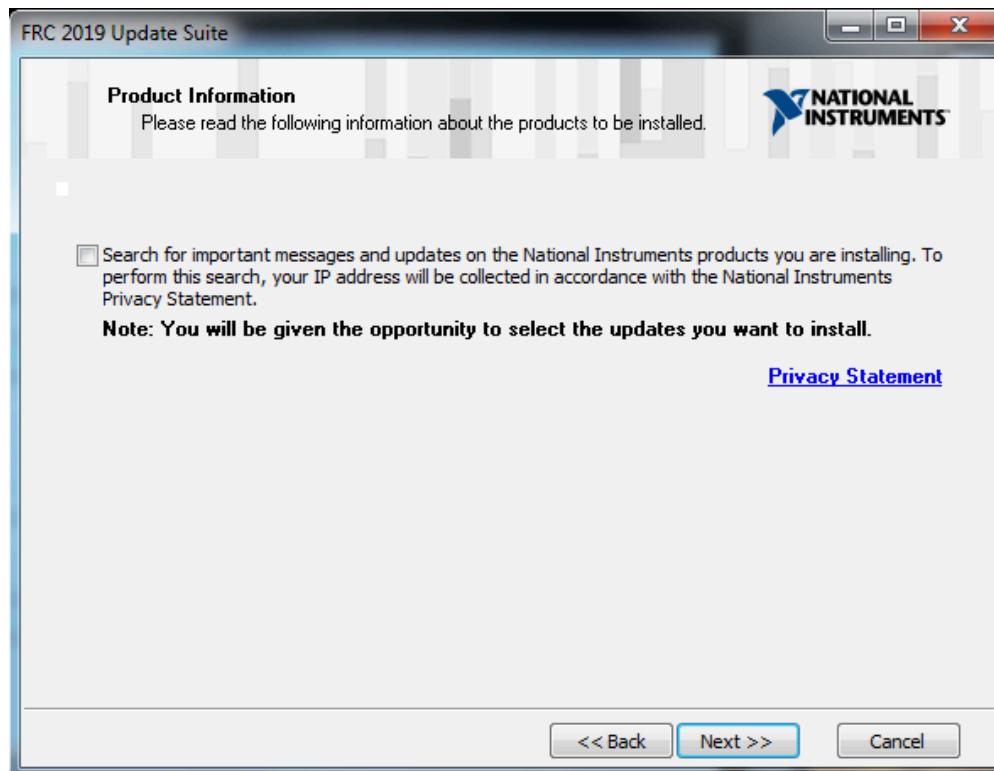
## Product List



Click "Next". There is no need to de-select "LabVIEW Update" for C++ or Java teams, if you do not have the base LabVIEW installation (because you are not programming in LabVIEW) this installation will be skipped automatically.

# Getting Started with the Control System

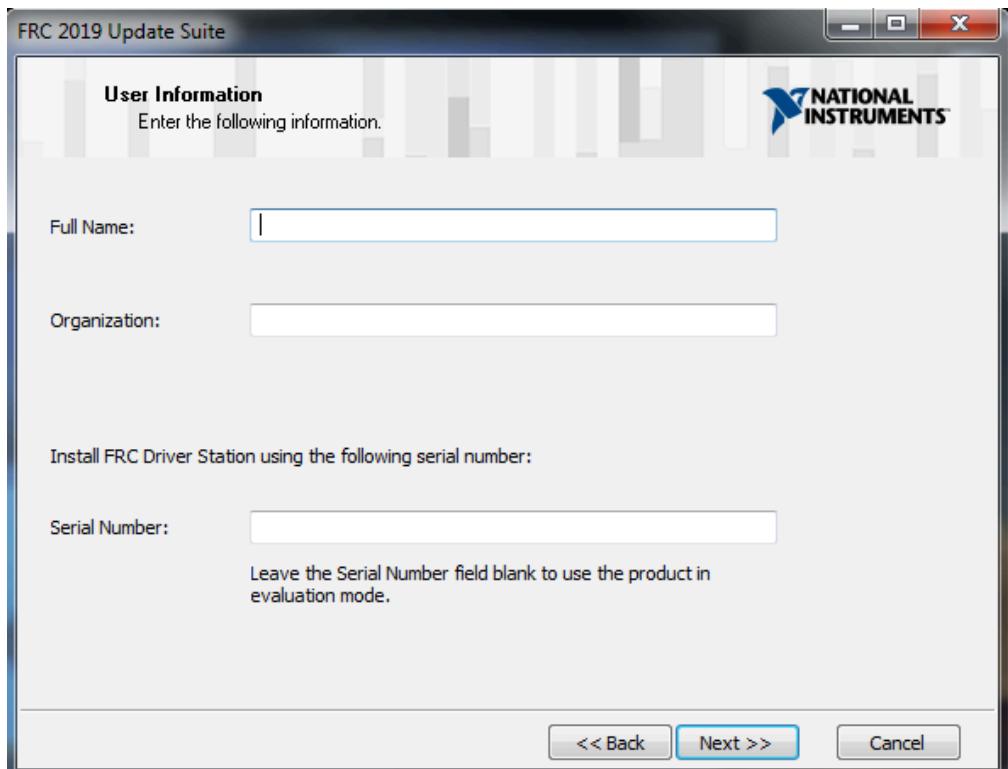
## Product Information



Un-check the box, then Click "Next".

# Getting Started with the Control System

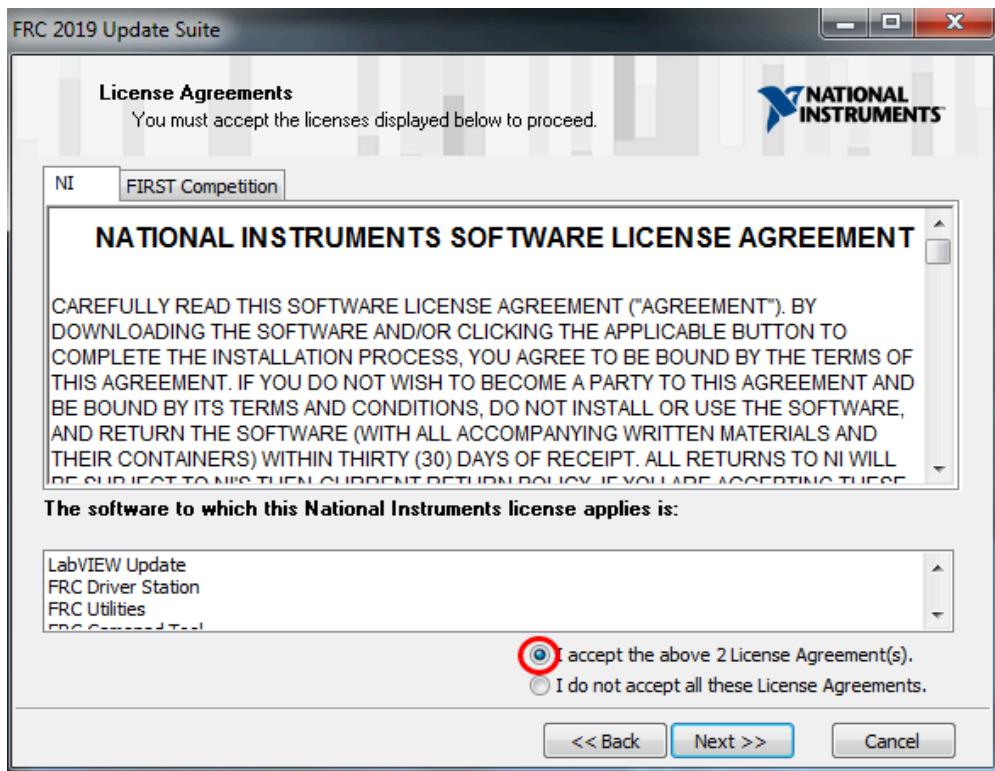
## User Information



Enter name and organization and the serial number from the Team Registration System (in the Password/Voucher code section) then click **Next**

# Getting Started with the Control System

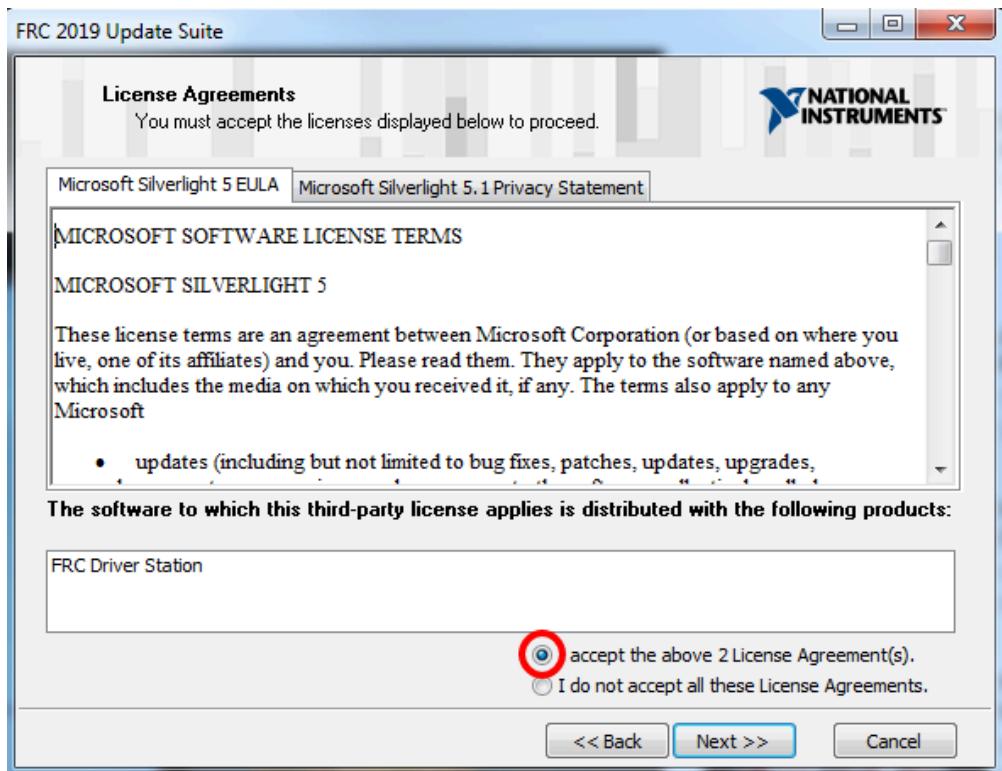
## License Agreements



Select "I accept..." then click "Next"

# Getting Started with the Control System

## License Agreements Page 2



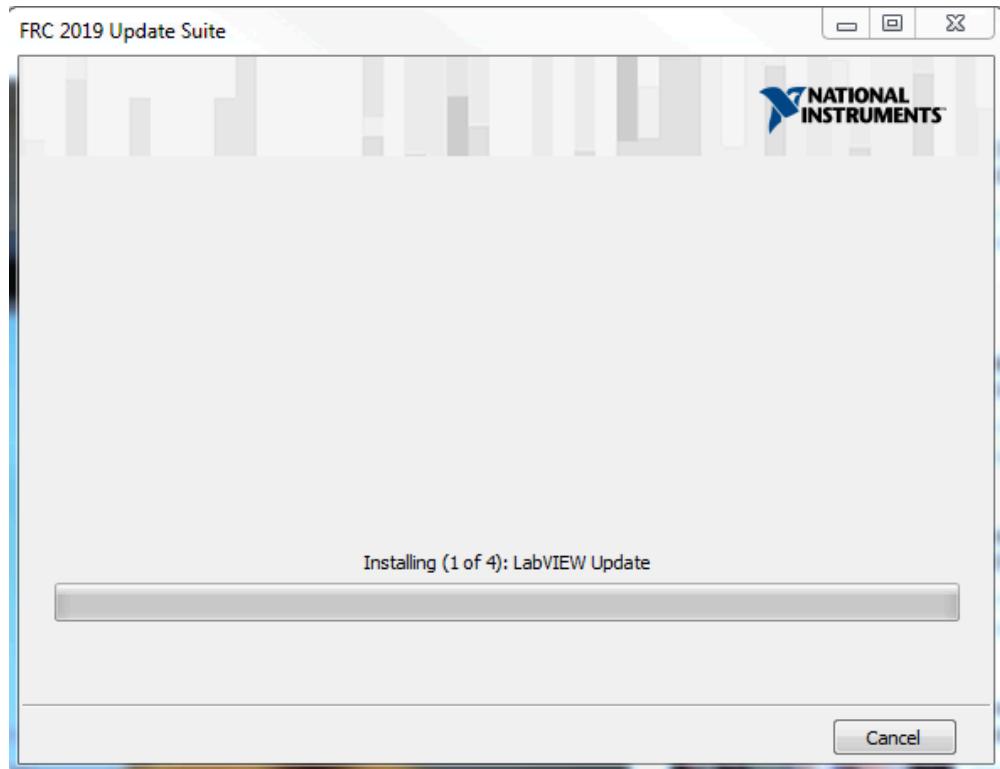
Select "I accept..." then click "Next"

If you see a screen asking to disable Windows Fast Startup, leave it at the recommended option (disable Fast Startup) and click **Next**.

If you see a screen talking about Windows Firewall, click **Next**.

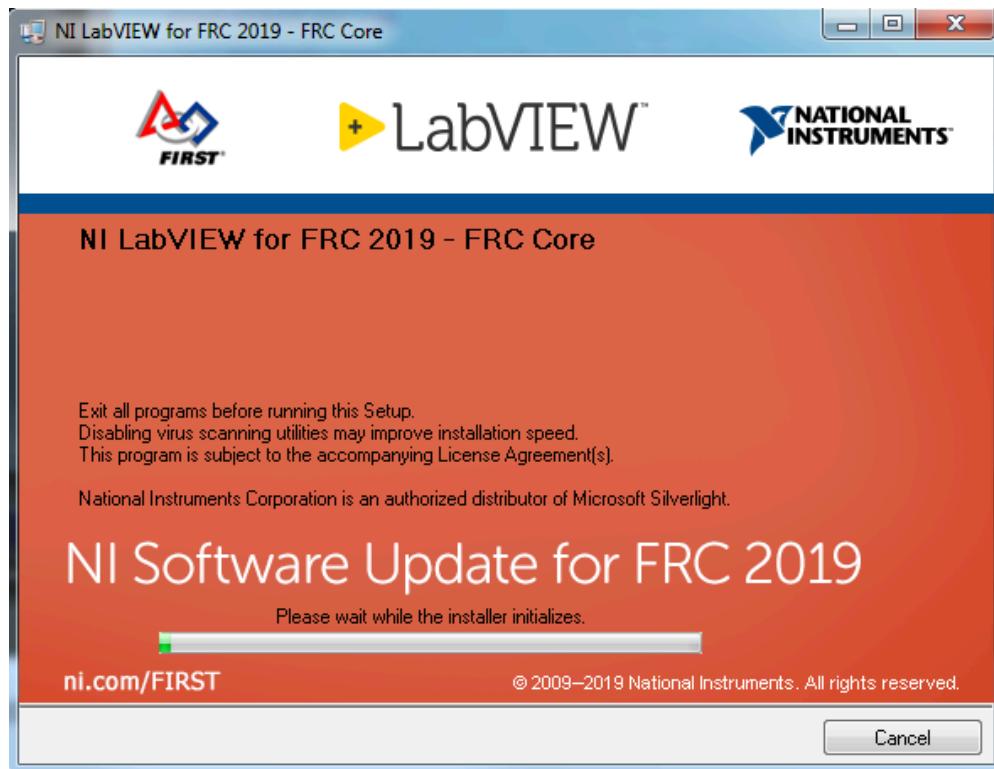
# Getting Started with the Control System

## Summary Progress



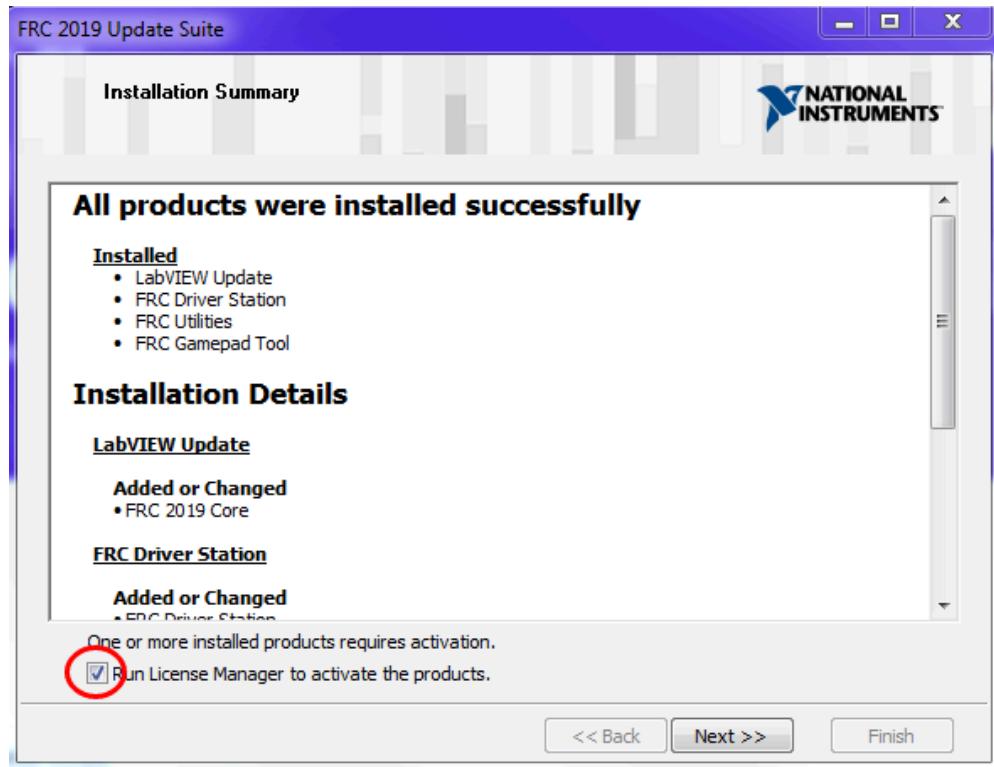
# Getting Started with the Control System

## Detail Progress



# Getting Started with the Control System

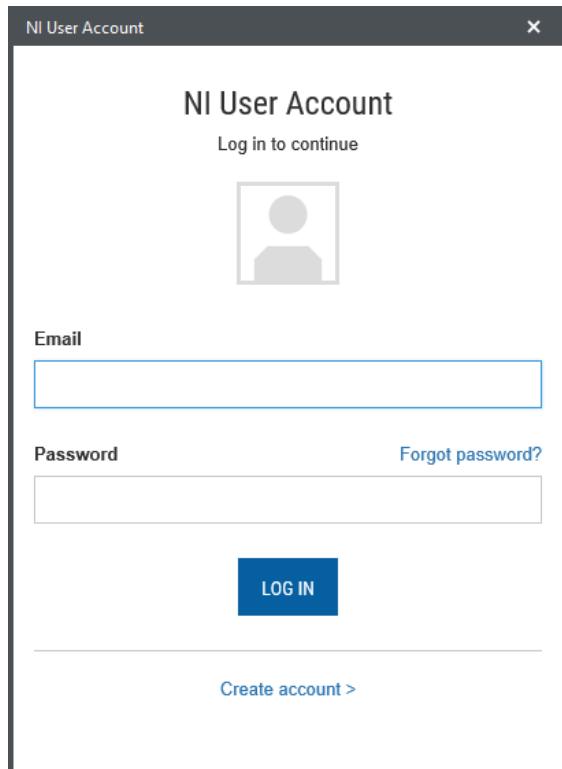
## Installation Summary



Make sure the box is checked to **Run License Manager...** then click **Next** or **Finish**

# Getting Started with the Control System

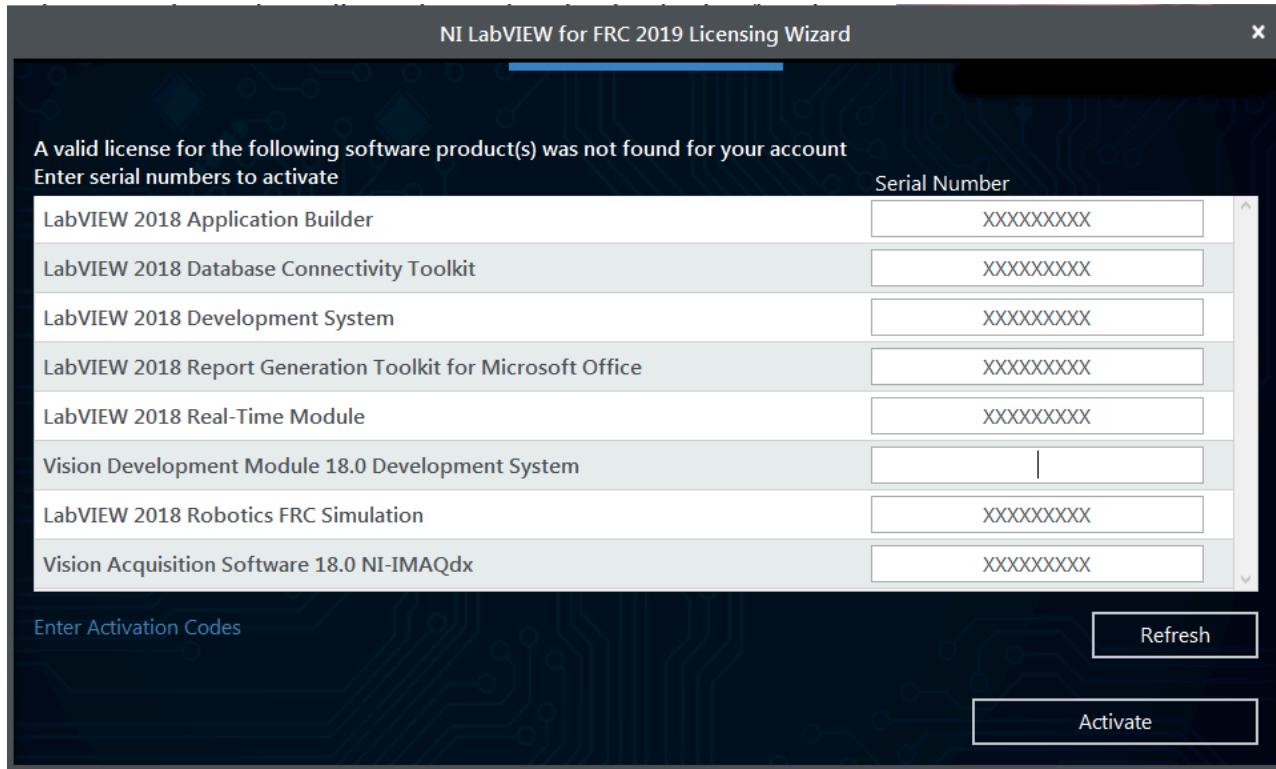
## NI Activation Wizard



Log into your ni.com account. If you don't have an account, select 'Create account' to create a free account.

# Getting Started with the Control System

## NI Activation Wizard (2)

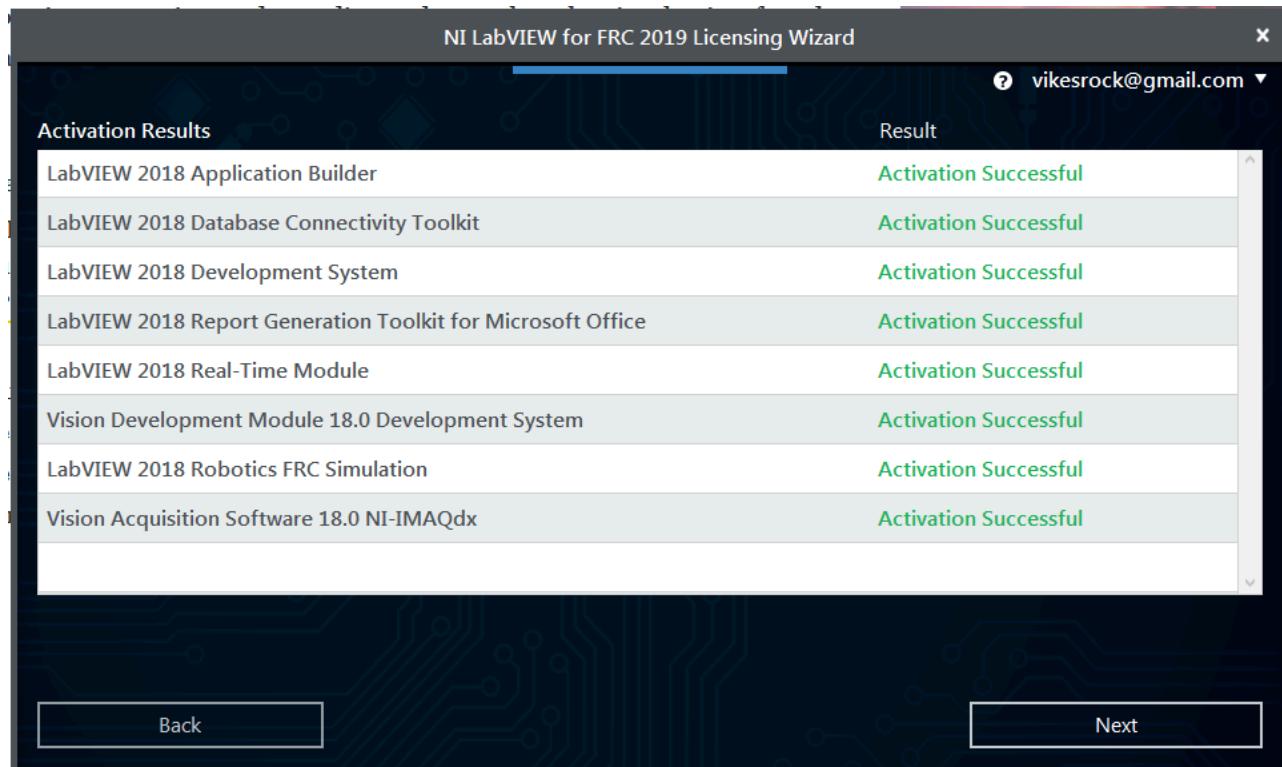


The serial number you entered at the "User Information" screen should appear in all of the text boxes, if it doesn't, enter it now. Click "Activate".

Note: If this is the first time activating the 2019 software on this account, you will see the message shown above about a valid license not being found. You can ignore this.

# Getting Started with the Control System

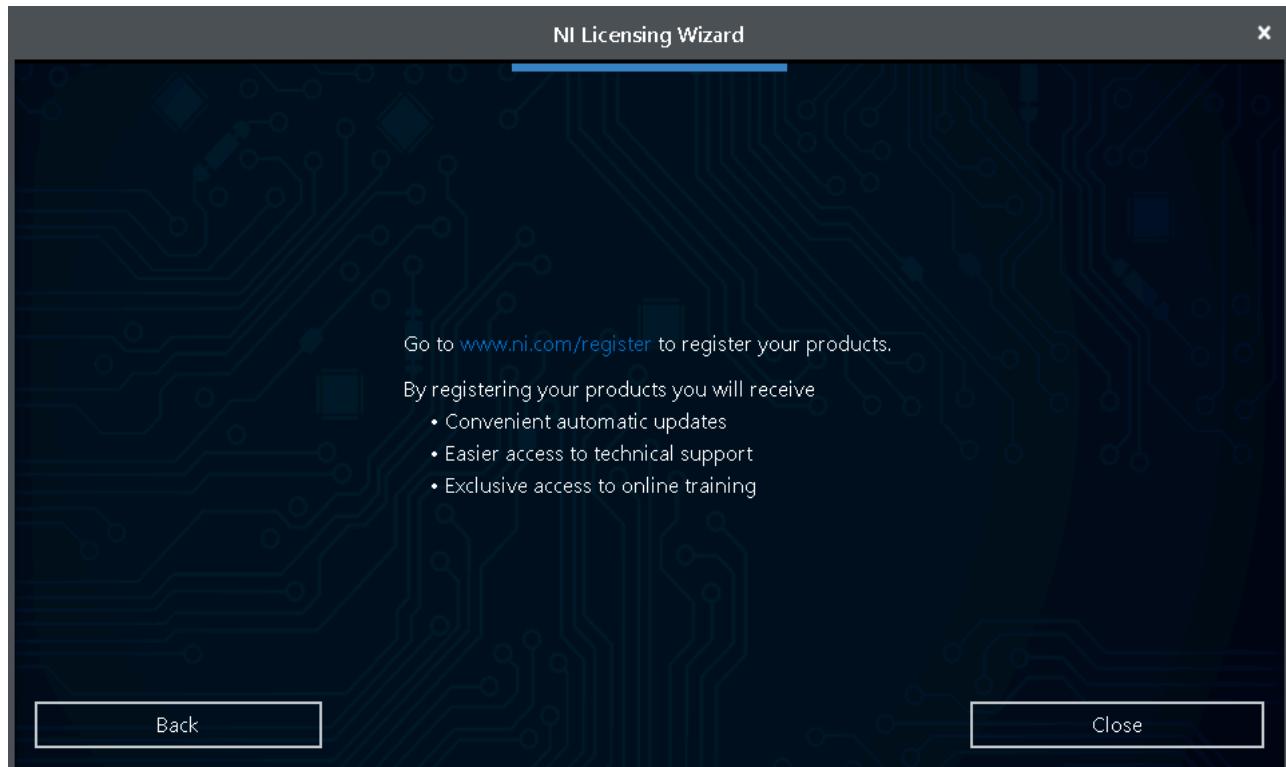
## NI Activation Wizard (3)



If your products activate successfully, an “Activation Successful” message will appear. If the serial number was incorrect, it will give you a text box and you can re-enter the number and select “Try Again”. If everything activated successfully, click “Next”.

# Getting Started with the Control System

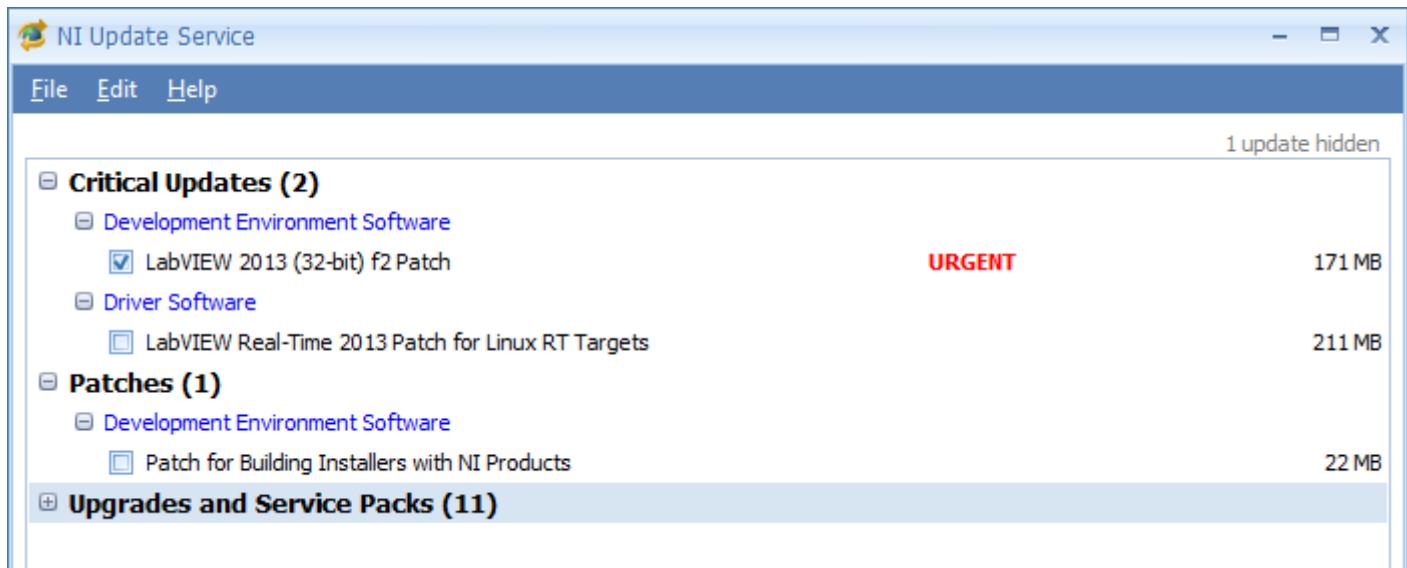
## NI Activation Wizard (4)



Click "Close".

# Getting Started with the Control System

## NI Update Service



On occasion you may see alerts from the NI Update Service about patches to LabVIEW. It is not recommended to install these patches. **FRC will communicate any recommended updates through our usual channels** (Frank's Blog, Team Updates or E-mail Blasts).

## Installing C++ and Java Development Tools for FRC

Install instructions for C++ and Java development tools are provided below. Click the arrow to expand the instructions for your specific platform.

### Windows

#### Offline Installer

Note:

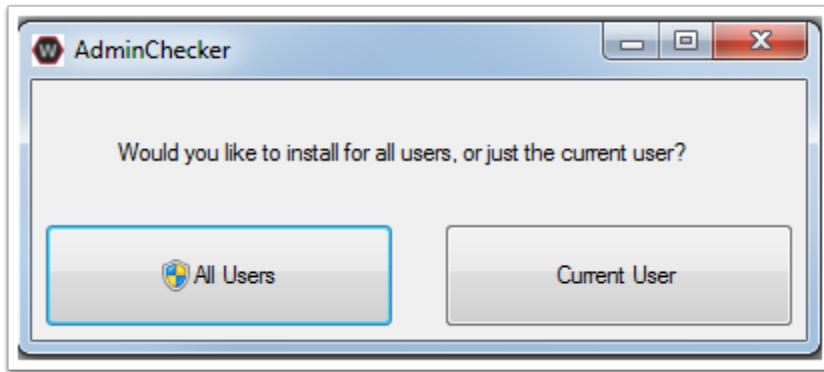
- ⚠** Windows 7: You must install the NI Update or .NET Version 4.62 (or later) before proceeding with the install of VSCode for FRC. The NI Update installer will automatically install the proper version of .NET. The stand alone .NET installer is here: <https://support.microsoft.com/en-us/help/3151800/the-.net-framework-4-6-2-offline-installer-for-windows>

Download the appropriate installer for your Windows installation (32 bit or 64 bit) [from GitHub](#). If you're not sure, open the Control Panel -> System to check.

Double click on the installer to run it. If you see any Security warnings, click Run (Windows 7) or More Info->Run Anyway (Windows 8+).

# Getting Started with the Control System

## Installation Type

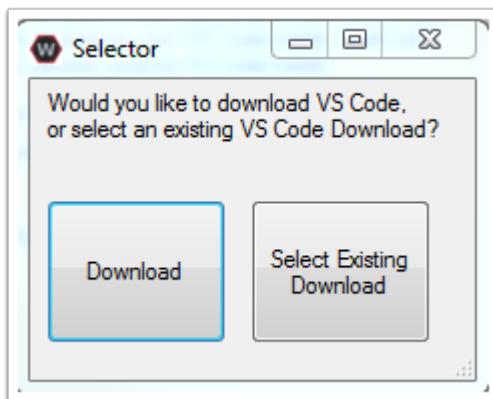


Choose whether to install for All Users on the machine or the Current User. The All Users option requires Admin privileges, but installs in a way that is accessible to all user accounts, the Current User install is only accessible from the account it is installed from.

If you select All Users, you will need to accept the security prompt that appears.

## Download VSCode

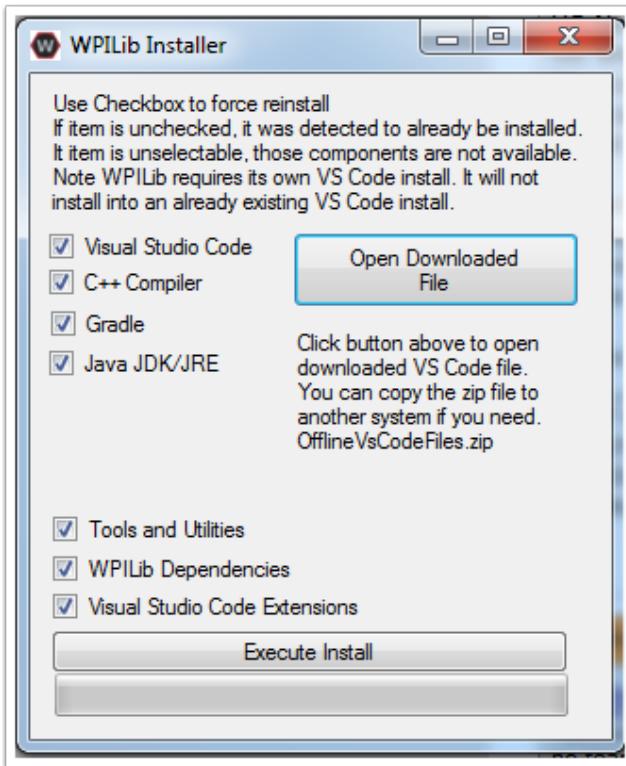
For licensing reasons, the installer cannot contain the VSCode installer bundled in. Click Select/Download VSCode to either Download the VSCode installer or select a pre-downloaded copy. If you intend to install on other machines without internet connections, after the download completes, you can click Open Downloaded File to be taken to the zip file on the file system to copy along with the Offline Installer.



# Getting Started with the Control System

## Execute Install

Make sure all checkboxes are checked (unless you have already installed 2019 WPILib software on this machine and the software unchecked them automatically), then click Execute Install.



## What's Installed?

The Offline Installer installs the following components:

- Visual Studio Code - The supported IDE for 2019 robot code development. The offline installer sets up a separate copy of VSCode for WPILib development, even if you already have VSCode on your machine. This is done because some of the settings that make the WPILib setup work may break existing workflows if you use VSCode for other projects.
- C++ Compiler - The toolchains for building C++ code for the roboRIO
- Gradle - The specific version of Gradle used for building/deploying C++ or Java robot code

# Getting Started with the Control System

- Java JDK/JRE - A specific version of the Java JDK/JRE that is used to build Java robot code and to run any of the Java based Tools (Dashboards, etc.). This exists side by side with any existing JDK installs and does not overwrite the JAVA\_HOME variable
- WPILib Tools - SmartDashboard, Shuffleboard, Robot Builder, Outline Viewer, Pathweaver
- WPILib Dependencies - OpenCV, etc.
- VSCode Extensions - WPILib extensions for robot code development in VSCode

## What's Installed - Continued

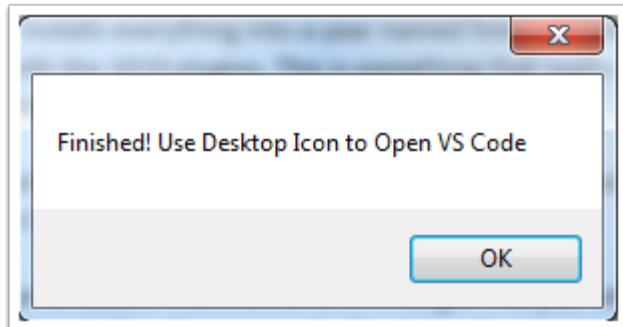
The Offline Installer also installs a Desktop Shortcut to the WPILib copy of VSCode and sets up a command shortcut so this copy of VSCode can be opened from the command line using the command "frccode2019"

Both of these reference the specific year as the WPILib C++\Java tools will now support side-by-side installs of multiple environments from different seasons.



## Finished

When the installer completes, you will now be able to open and use the WPILib version of VSCode. If you are using any 3rd party libraries, you will still need to install those separately before using them in robot code.



# Getting Started with the Control System

## Mac OS

The tools (except the Driver Station and the roboRIO Imaging Tool) will run natively on a Mac.

***Note: if you have the alpha release of VSCode for FRC installed, you should uninstall it before proceeding or create a new VSCode install. Failing to do this will have both versions installed at the same time causing things to not operate properly.***

To install it follow these steps.

## Getting Visual Studio Code

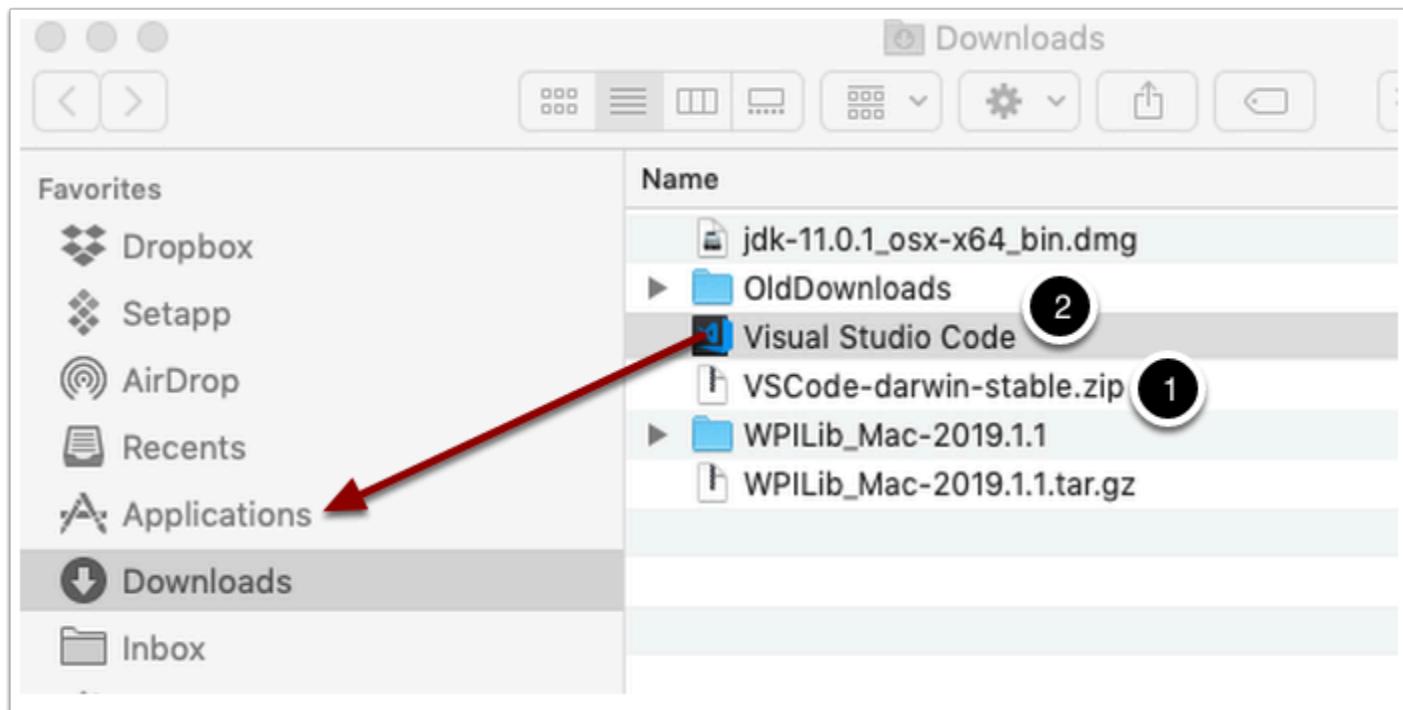
VSCode is the IDE (Integrated Development Environment) that is used for 2019 and beyond. It needs to be installed on any development computer. It can be downloaded here:

<https://code.visualstudio.com>.

The downloaded file will be "VSCode-darwin-stable.zip" (1)

Once downloaded, double-click on the zip file to expand it and copy the new file: "Visual Studio Code" to the Applications folder (2).

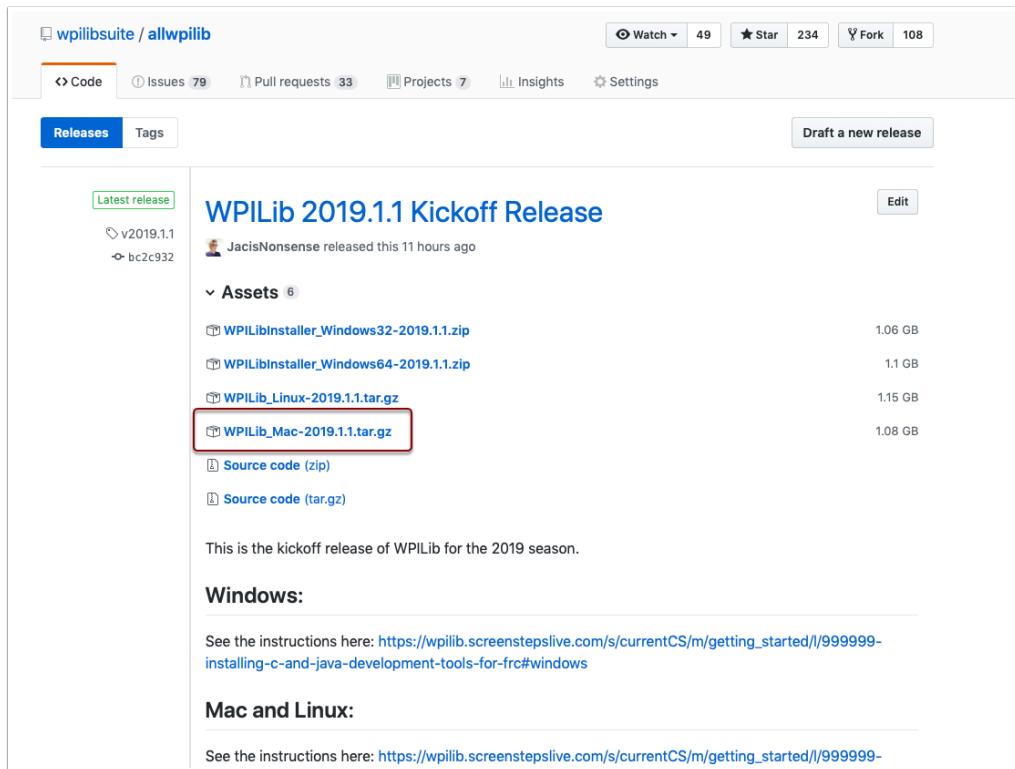
# Getting Started with the Control System



## Download the WPIlib release and move the directory

Download the software release by navigating to this page: <https://github.com/wpilibsuite/allwpilib/releases> and downloading the Mac release.

# Getting Started with the Control System

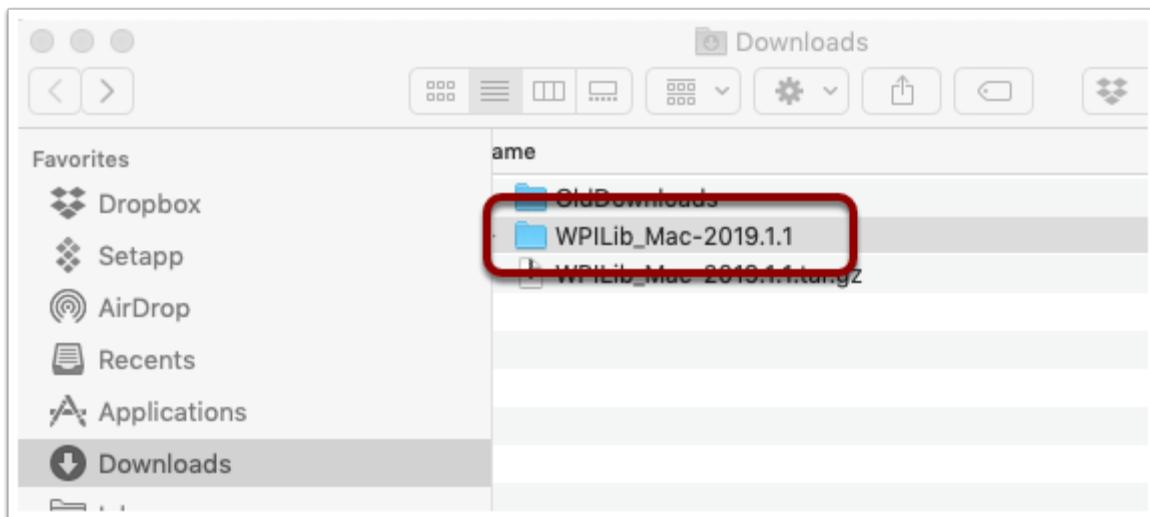


The screenshot shows the GitHub release page for the WPILib 2019.1.1 Kickoff Release. The 'Releases' tab is selected. The latest release is v2019.1.1, made by JacisNonsense 11 hours ago. The 'Assets' section lists several files:

Asset	Size
WPILibInstaller_Windows32-2019.1.1.zip	1.06 GB
WPILibInstaller_Windows64-2019.1.1.zip	1.1 GB
WPILib_Linux-2019.1.1.tar.gz	1.15 GB
WPILib_Mac-2019.1.1.tar.gz	1.08 GB

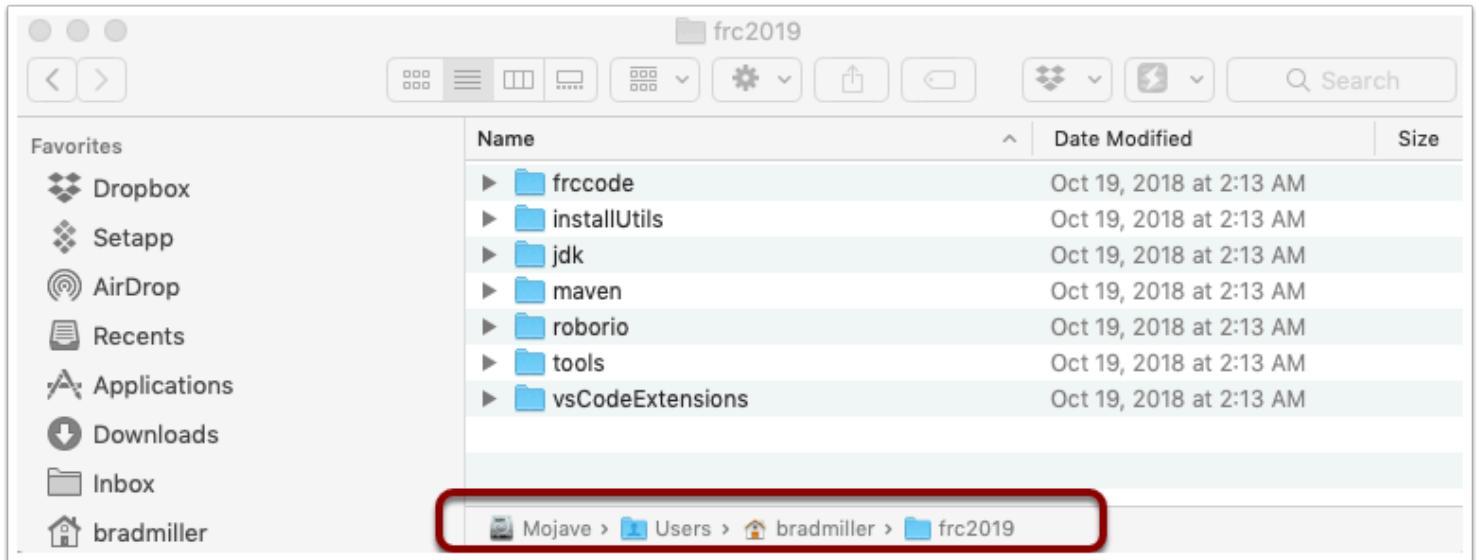
The 'WPILib\_Mac-2019.1.1.tar.gz' file is highlighted with a red box. Below the assets, there is a note about the kickoff release and links for Windows and Mac/Linux instructions.

Unzip and untar the file by looking at the file in the explorer and double-clicking on it, once or twice to unzip (remove the .gz extension) and again to untar it (remove the .tar extension). When finished it should like like the folder shown below.



# Getting Started with the Control System

Using Finder (or command line) copy the contents of the folder to a new folder in your home directory, ~/frc2019 as shown below.



## Run the ToolsUpdater.py script

To update all the additional tools WPILib tools, open a terminal window and change directory to ~/frc2019/tools and run the script ToolsUpdater.py with the command:

```
python ToolsUpdater.py
```

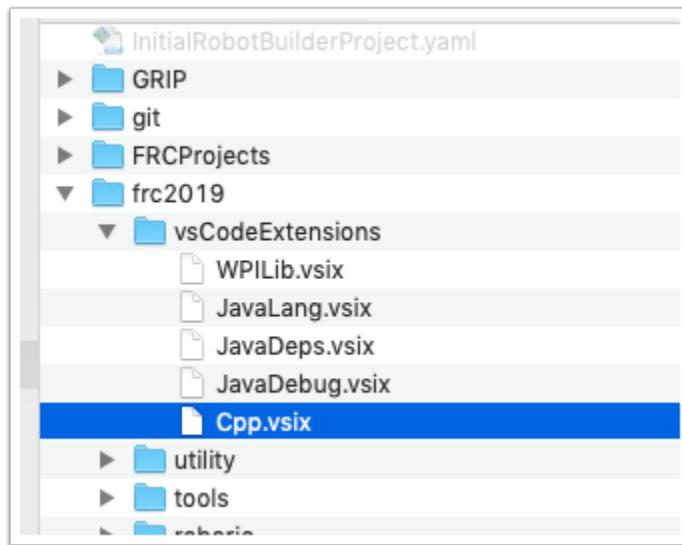
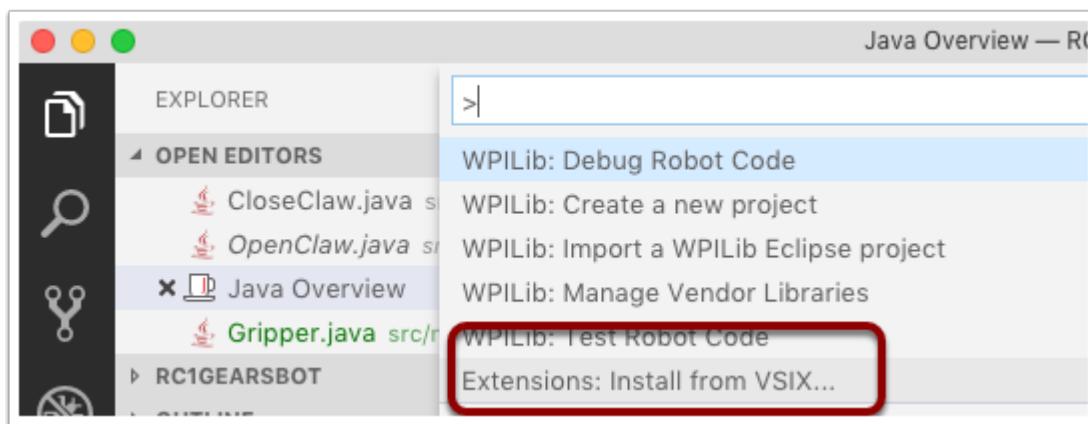
This should populate the tools directory with all of the WPILib tools (Shuffleboard, Robot Builder, PathWeaver, etc.)

```
Last login: Sun Oct 28 10:01:24 on console  
Brads-MacBook-Pro-514:~ bradmiller$ cd frc2019/  
Brads-MacBook-Pro-514:frc2019 bradmiller$ cd tools  
Brads-MacBook-Pro-514:tools bradmiller$ ls  
ScriptBase.sh      ScriptBase.vbs      TrailSimulator.jar  
Brads-MacBook-Pro-514:tools bradmiller$ python ToolsUpdater.py  
ToolsUpdater.sh      tools.json
```

# Getting Started with the Control System

## Installing the extensions for WPILib development

Before using VSCode for WPILib development there are a number of extensions that need to be installed. Start up VSCode and type the shortcut Cmd-Shift-P to bring up the list of commands available. Start typing "Install from VSIX" into the search box. Choose that command. In the file selection box select Cpp.vsix.



You should see a message confirming the install and asking to reload vscode. Click the reload button then repeat the vsix installation for the rest of the vsix files **in this order:**

1. Cpp.vsix
2. JavaLang.vsix

# Getting Started with the Control System

3. JavaDeps.vsix
4. JavaDebug.vsix
5. WPILib.vsix

## Setting up VSCode to use Java 11

The WPILib installation includes a JDK, however you need to point VS Code at where it is. To do this:

1. Open VS Code
2. Press **Ctrl + Shift + P** and type **WPILib** or click on the WPILib icon in the top right to open the WPILib Command Palette
3. Begin typing **Set VS Code Java Home to FRC Home** and select that item from the dropdown

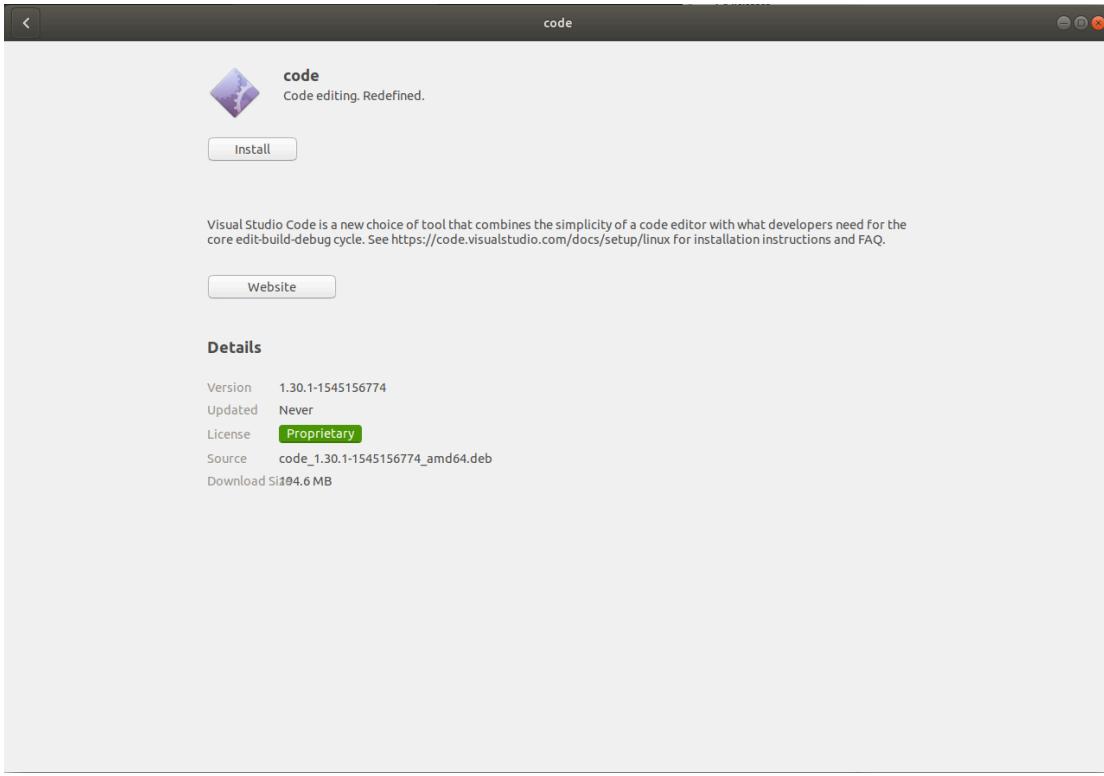


## Linux

### Installing VS Code

1. Download the Linux .deb file from [code.visualstudio.com](https://code.visualstudio.com)
2. Double-click on the .deb file in the file explorer
3. Click the "Install" button to install VSCode

# Getting Started with the Control System

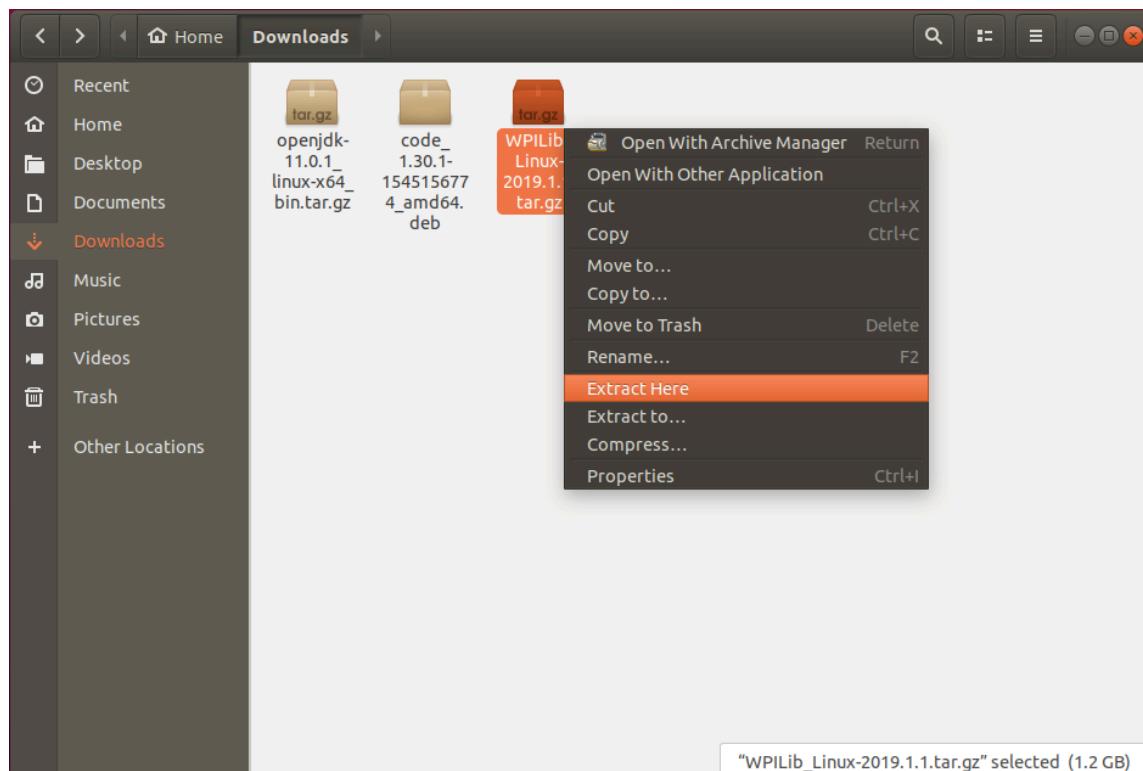
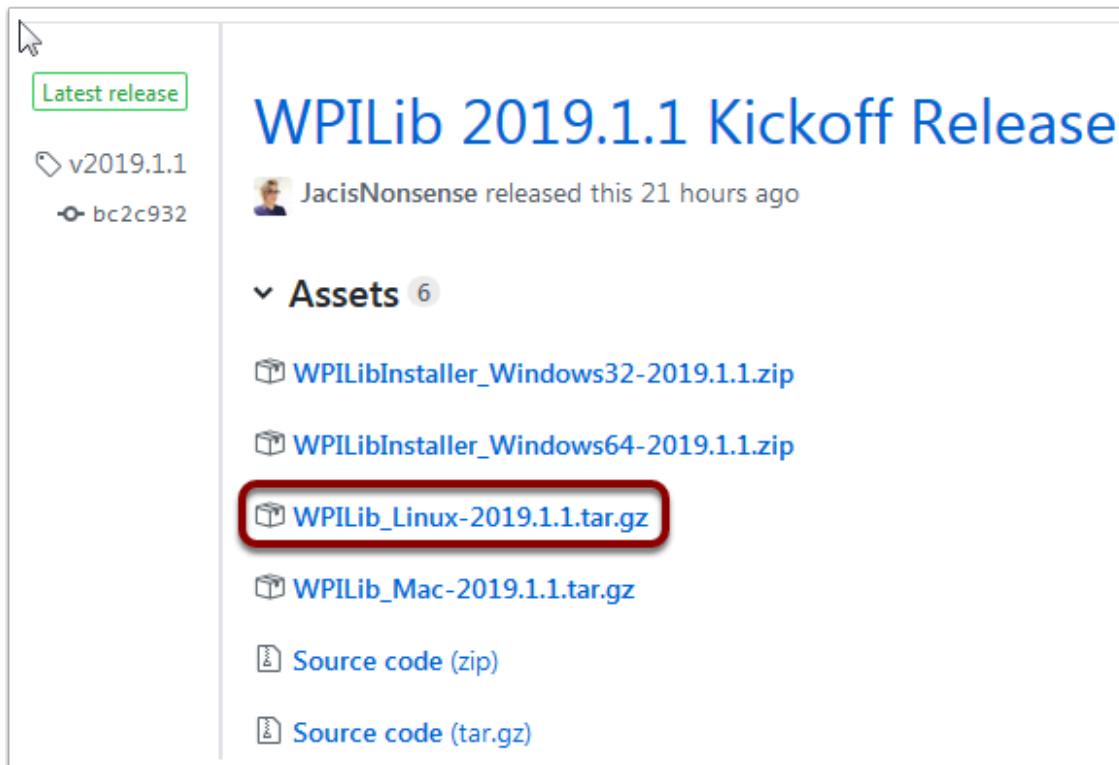


## Download the WPILib Release

Download the latest Linux release from <https://github.com/wpilibsuite/allwpilib/releases>

Right-click on the downloaded archive, click "Extract Here"

# Getting Started with the Control System



# Getting Started with the Control System

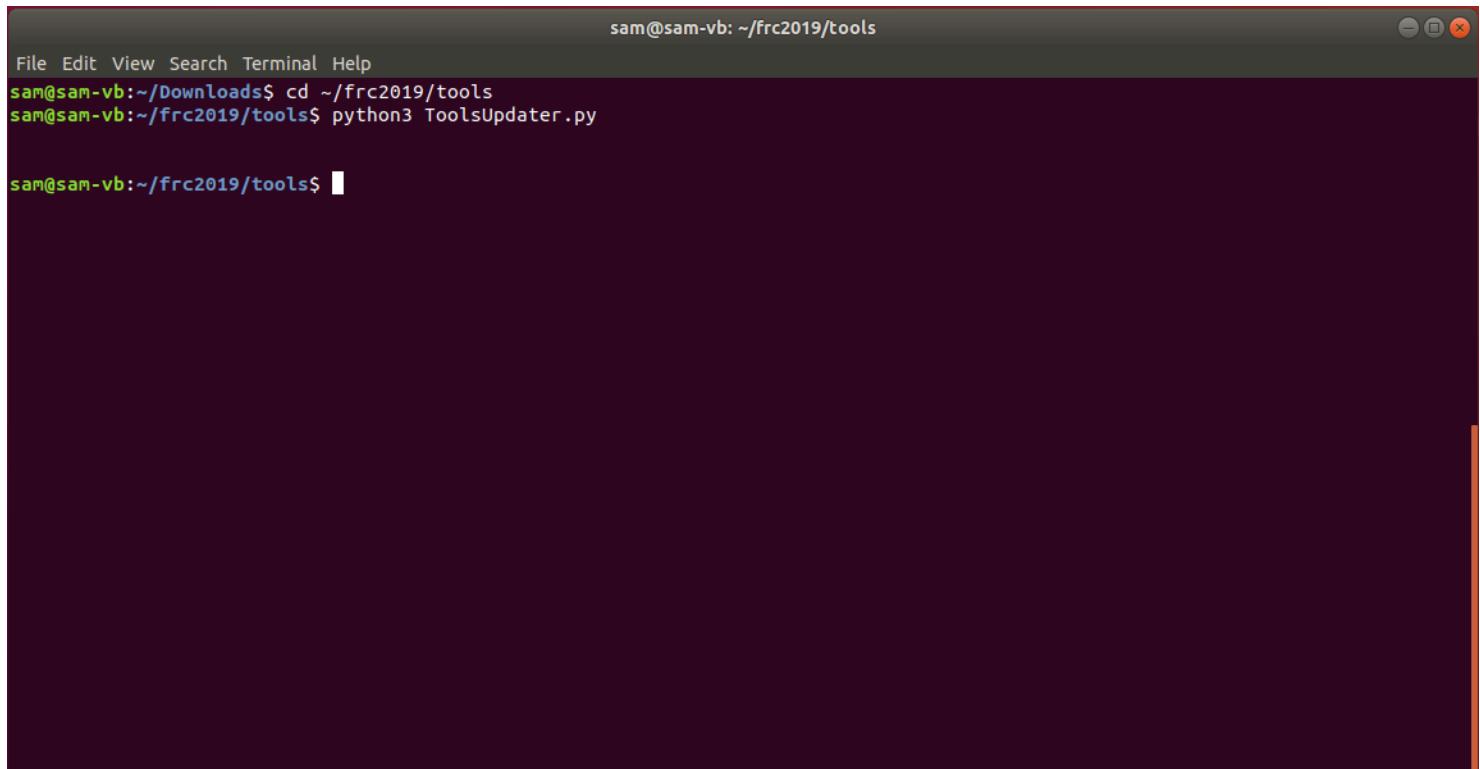
## Moving to FRC2019

1. Create a directory in your home directory called frc2019 - either from the file manager or with `$ mkdir ~/frc2019`
2. Drag the contents of WPILib\_Linux-2019.1.1 directory to ~/frc2019 or run `$ mv -v WPILib_Linux-2019.1.1/* ~/frc2019`

## Running Tools Updater

To extract the WPILib tools (Dashboards, Robot Builder, etc.), run:

- `$ cd ~/frc2019/tools`
- `$ python3 ToolsUpdater.py`

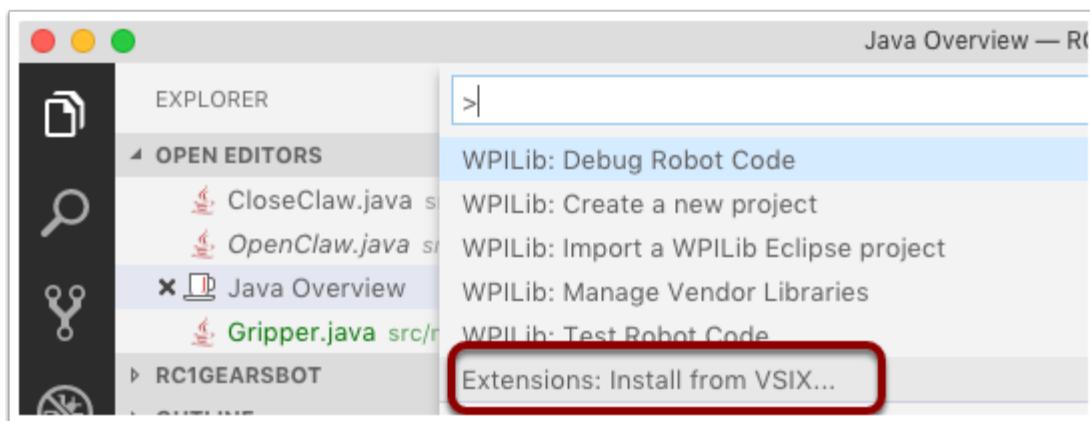


The screenshot shows a terminal window with a dark background and light-colored text. At the top, it says "sam@sam-vb: ~/frc2019/tools". Below that is a menu bar with "File Edit View Search Terminal Help". The terminal itself has the following text:  
sam@sam-vb:~/Downloads\$ cd ~/frc2019/tools  
sam@sam-vb:~/frc2019/tools\$ python3 ToolsUpdater.py  
  
The command "python3 ToolsUpdater.py" is highlighted in green, indicating it was just typed or is being processed.

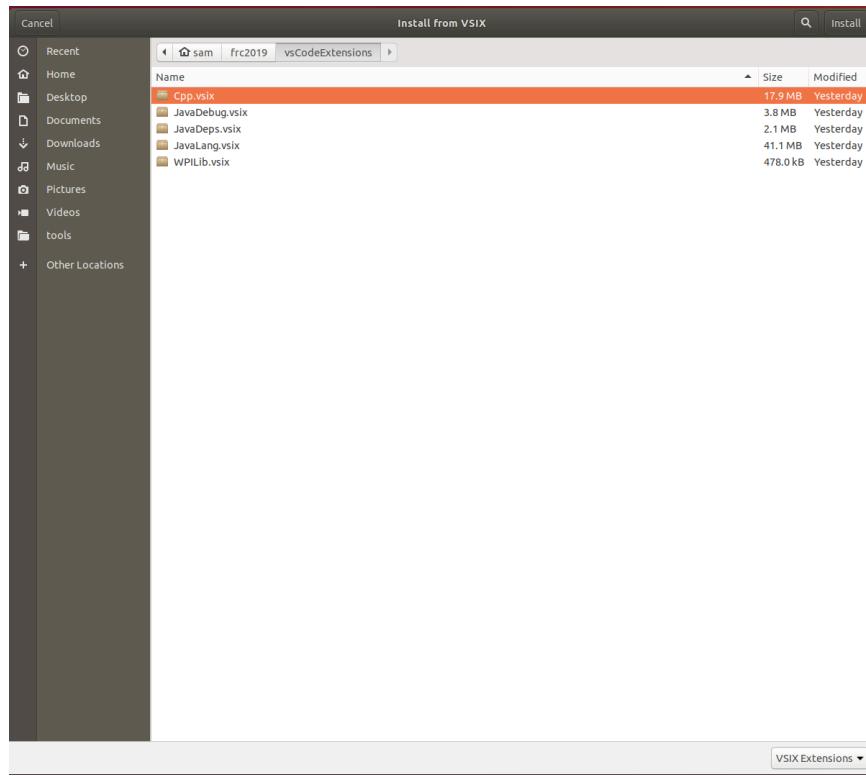
# Getting Started with the Control System

## Installing the extensions for WPILib VS Code

1. Start VSCode (\$ code or search "Visual Studio Code" in your application launcher)
2. Control+Shift+P to bring up the command palette, type "Install from VSIX"
3. Select Extensions: Install from VSIX...
4. Navigate to `~/frc2019/vsCodeExtensions` and select Cpp.vsix
5. Repeat for JavaLang.vsix, JavaDeps.vsix, JavaDebug.vsix, and WPILib.vsix in that order



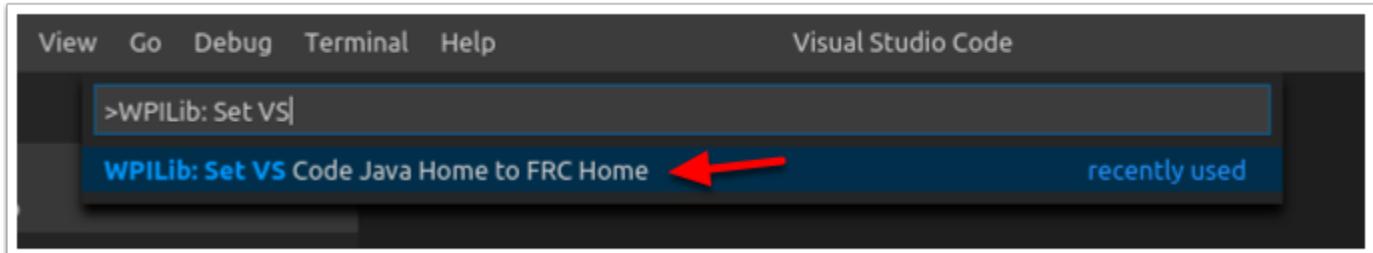
# Getting Started with the Control System



## Setting up VSCode to use Java 11

The WPILib installation includes a JDK, however you need to point VS Code at where it is. To do this:

1. Open VS Code
2. Press **Ctrl + Shift + P** and type **WPILib** or click on the WPILib icon in the top right to open the WPILib Command Palette
3. Begin typing **Set VS Code Java Home to FRC Home** and select that item from the dropdown



## Getting Started with the Control System

# Imaging your roboRIO

Before imaging your roboRIO, you must have completed installation of the [FRC Update Suite](#). You also must have the roboRIO power properly wired to the Power Distribution Panel as described [here](#).

**Make sure the power wires to the roboRIO are secure and that the connector is secure firmly to the roboRIO (4 total screws to check).**

## Configuring the roboRIO

The roboRIO Imaging Tool will be used to image your roboRIO with the latest software.

### USB Connection



# Getting Started with the Control System

Connect a USB cable from the roboRIO USB Device port to the PC. This requires a USB Type A male (standard PC end) to Type B male cable (square with 2 cut corners), most commonly found as a printer USB cable.

**Note: The roboRIO should only be imaged via the USB connection. It is not recommended to attempt imaging using the Ethernet connection.**

## Driver Installation

The device driver should install automatically. If you see a "New Device" pop-up in the bottom right of the screen, wait for the driver install to complete before continuing.

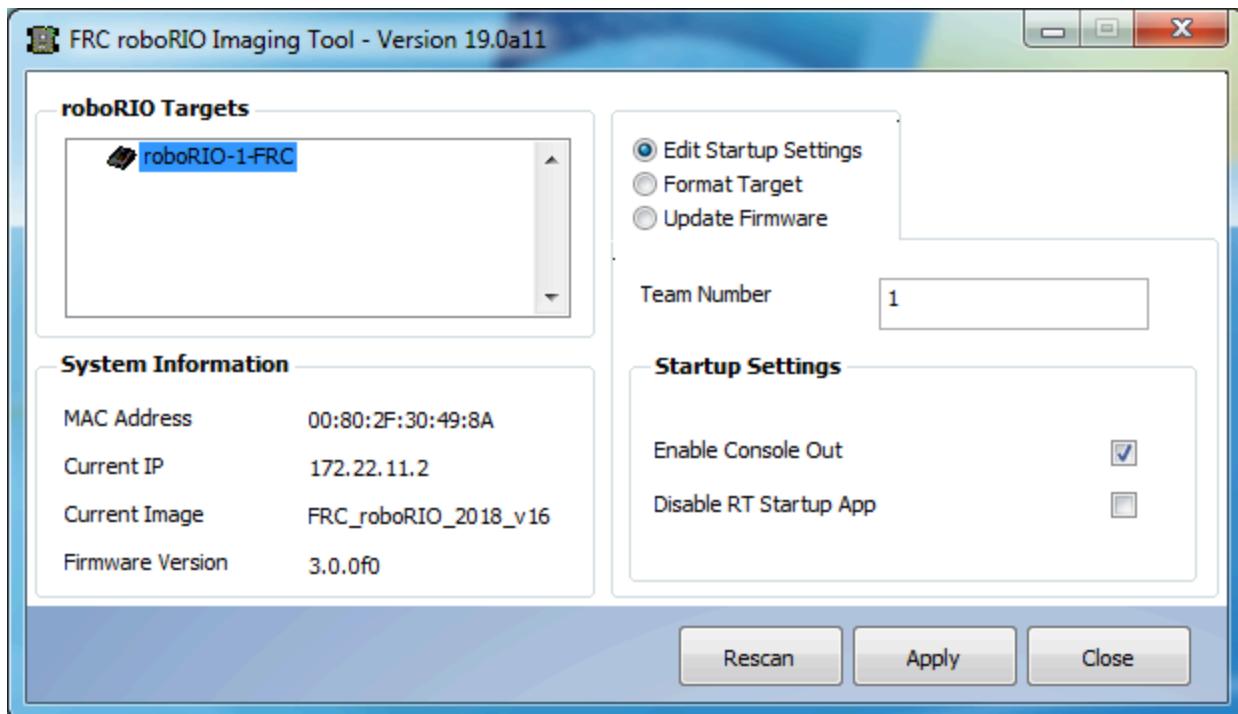
## Launching the Imaging Tool



The roboRIO imaging tool and latest image are installed with the NI Update Suite. Launch the imaging tool by double clicking on the shortcut on the Desktop. If you have difficulties imaging your roboRIO, you may need to try right-clicking on the icon and selecting Run as Administrator instead.

# Getting Started with the Control System

## roboRIO Imaging Tool

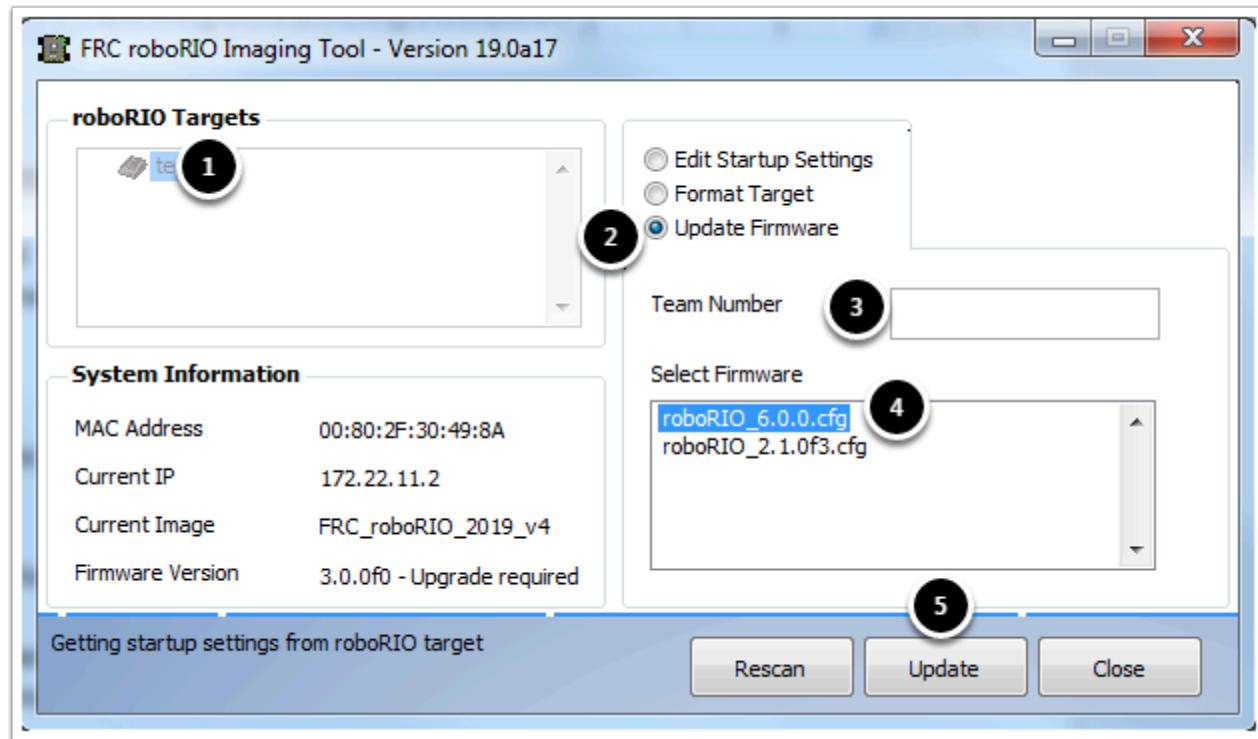


After launching, the roboRIO Imaging Tool will scan for available roboRIOs and indicate any found in the top left box. The bottom left box will show information and settings for the roboRIO currently selected. The right hand pane contains controls for modifying the roboRIO settings:

- **Edit Startup Settings** - This option is used when you want to configure the startup settings of the roboRIO (the settings in the right pane), without imaging the roboRIO.
- **Format Target** - This option is used when you want to load a new image on the roboRIO (or reflash the existing image). This is the most common option.
- **Update Firmware** - This option is used to update the roboRIO firmware. For this season, the imaging tool will require roboRIO firmware to be version 5.0 or greater.

# Getting Started with the Control System

## Updating Firmware



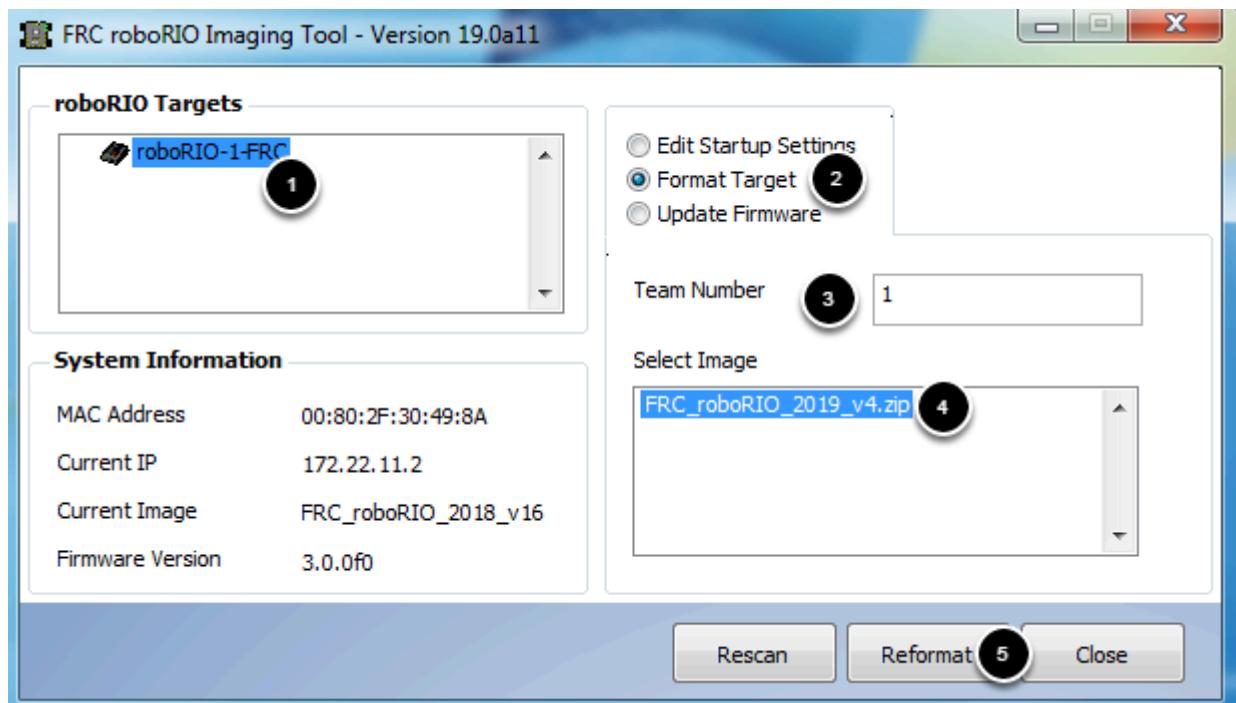
RoboRIO firmware must be at least v5.0 to work with the 2019 image. If your roboRIO is at least version 5.0 (as shown in the bottom left of the imaging tool) you do not need to update.

To update roboRIO firmware:

1. Make sure your roboRIO is selected in the top left pane.
2. Select Update Firmware in the top right pane
3. Enter a team number in the Team Number box
4. Select the latest firmware file in the bottom right
5. Click the **Update** button

# Getting Started with the Control System

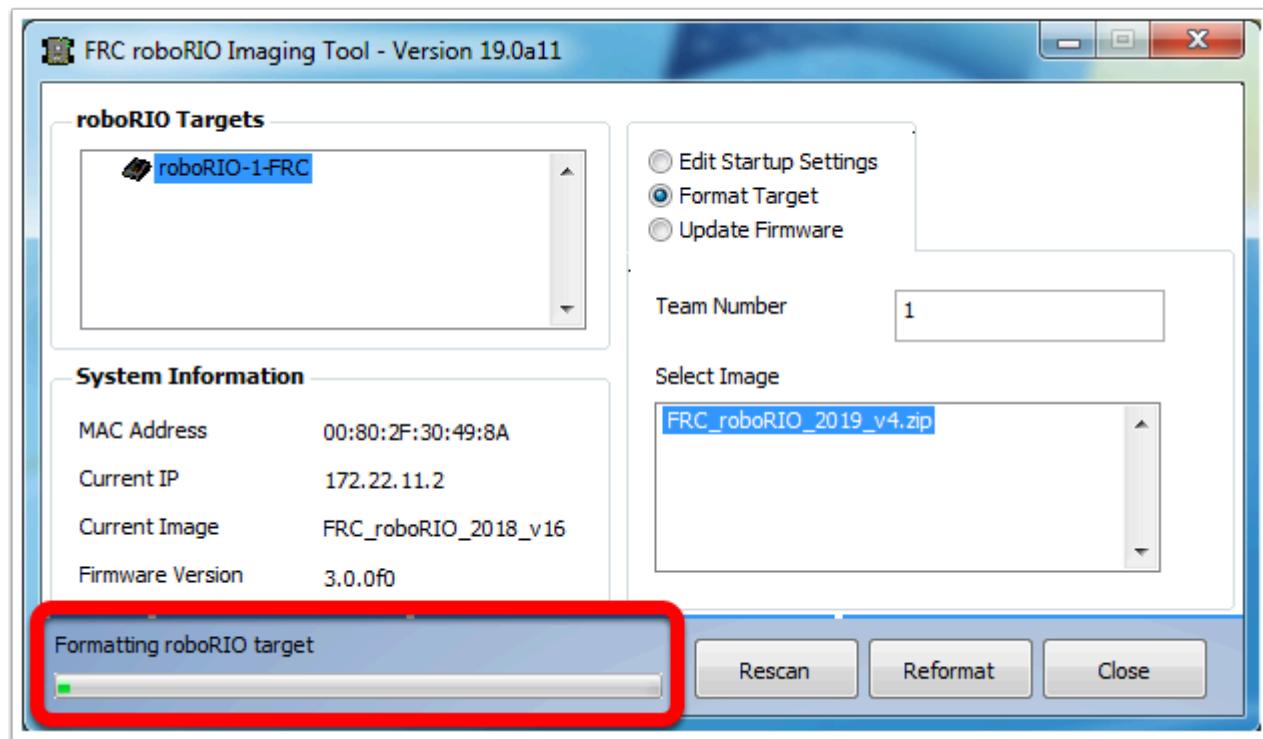
## Imaging the roboRIO



1. Make sure the roboRIO is selected in the top left pane
2. Select Format Target in the right pane
3. Enter your team number in the box
4. Select the latest image version in the box.
5. Click Reformat to begin the imaging process.

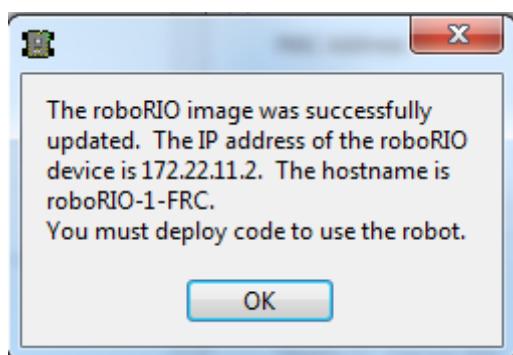
# Getting Started with the Control System

## Imaging Progress



The imaging process will take approximately 3-10 minutes. A progress bar in the bottom left of the window will indicate progress.

## Imaging Complete



# Getting Started with the Control System

When the imaging completes you should see the dialog above. Click Ok, then click the Close button at the bottom right to close the imaging tool. **Reboot the roboRIO using the Reset button to have the new team number take effect.**

**⚠** Note: The default CAN webdash functionality has been removed from the image (CAN devices will still work from robot code). You will need to use the tools provided by individual vendors to service their CAN devices.

## Troubleshooting

If you are unable to image your roboRIO, troubleshooting steps include:

1. Try running the roboRIO Imaging Tool as Administrator by right-clicking on the Desktop icon to launch it.
2. Try accessing the roboRIO webpage with a web-browser at <http://172.22.11.2/> and/or verify that the NI network adapter appears in your list of Network Adapters in the Control Panel. If not, try re-installing the NI Update Suite or try a different PC.
3. Make sure your firewall is turned off. More information on this can be found here: [Windows Firewall Configuration](#)
4. Try a different PC
5. Try booting the roboRIO into Safe Mode by pressing and holding the reset button for at least 5 seconds.

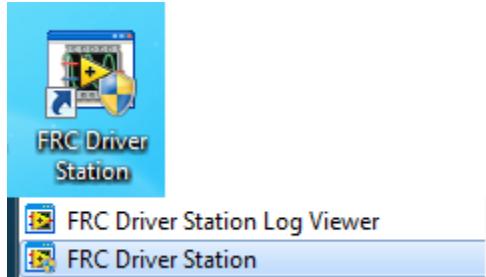
## Getting Started with the Control System

# FRC Driver Station Powered by NI LabVIEW

This article describes the use and features of the FRC Driver Station Powered by NI LabVIEW.

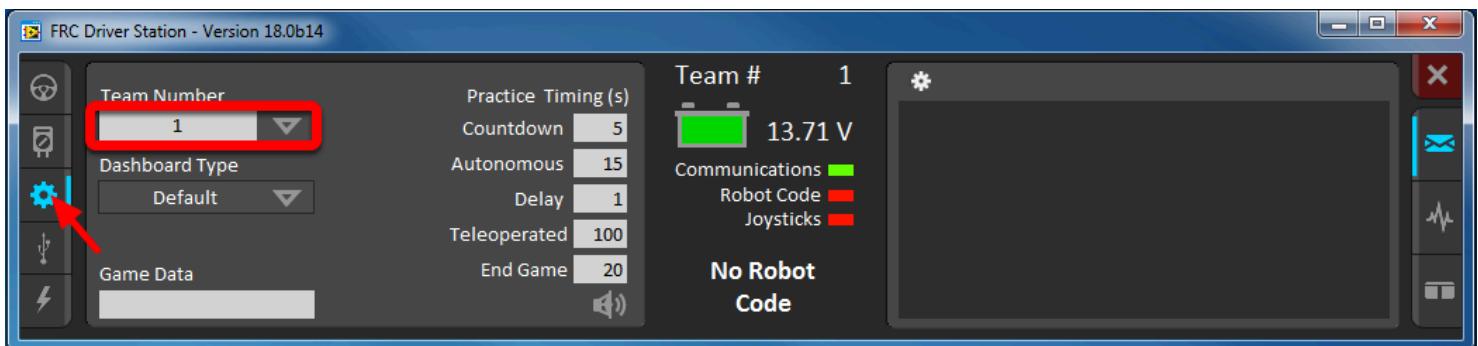
For information on installing the Driver Station software see [this document](#).

## Starting the FRC Driver Station



The FRC Driver Station can be launched by double-clicking the icon on the Desktop or by selecting Start->All Programs->FRC Driver Station.

## Setting Up the Driver Station

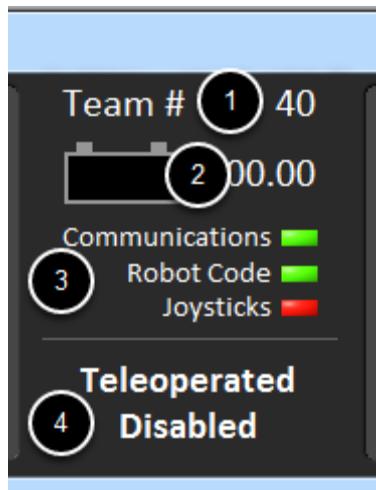


The DS must be set to your team number in order to connect to your robot. In order to do this click the **Setup** tab then enter your team number in the team number box. Press return or click outside the box for the setting to take effect.

# Getting Started with the Control System

PCs will typically have the correct network settings for the DS to connect to the robot already, but if not, make sure your Network adapter is set to DHCP.

## Status Pane

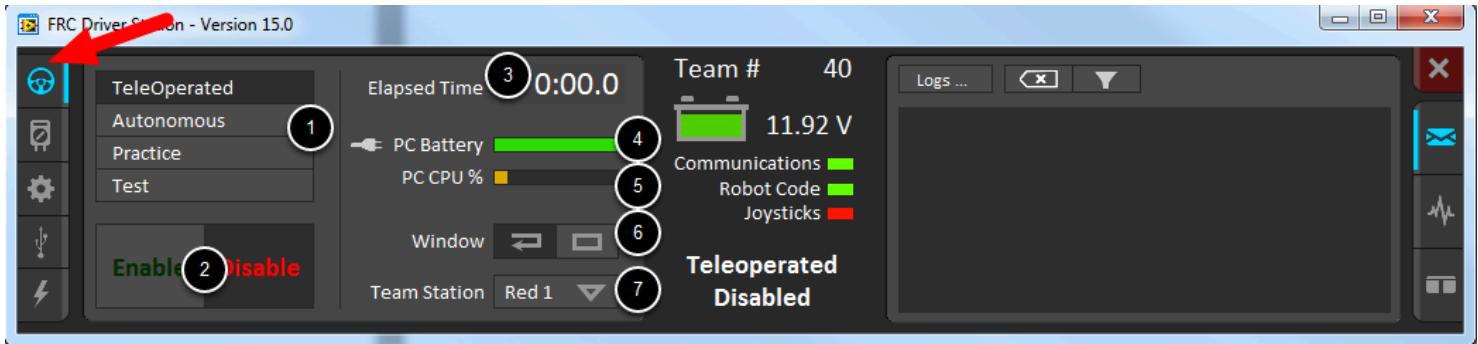


The Status Pane of the Driver Station is located in the center of the display and is always visible regardless of the tab selected. It displays a selection of critical information about the state of the DS and robot:

1. Team # - The Team number the DS is currently configured for. This should match your FRC team number, to change the number see the Setup Tab.
2. Battery Voltage - If the DS is connected and communicating with the roboRIO this displays current battery voltage as a number and with a small chart of voltage over time in the battery icon. The background of the numeric indicator will turn red when the roboRIO brownout is triggered. See [RoboRIO Brownout and Understanding Current Draw](#) for more information.
3. Major Status Indicators - These three indicators display major status items for the DS. The "Communications" indicates whether the DS is currently communicating with the FRC Network Communications Task on the roboRIO (this year it is split in half for the TCP and UDP communication). The "Robot Code" indicator shows whether the team Robot Code is currently running (determined by whether or not the Driver Station Task in the robot code is updating the battery voltage), The "Joysticks" indicator shows if at least one joystick is plugged in and recognized by the DS.
4. Status String - The Status String provides an overall status message indicating the state of the robot, some examples are "No Robot Communication", "No Robot Code", "Emergency Stopped", and "Teleoperated Enabled". When the roboRIO brownout is triggered this will display "Voltage Brownout".

# Getting Started with the Control System

## Operation Tab



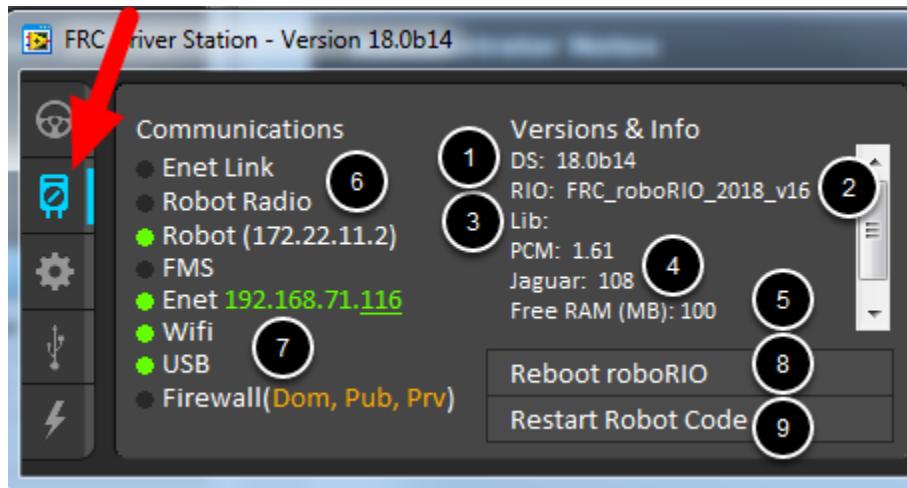
The Operations Tab is used to control the mode of the robot and provide additional key status indicators while the robot is running.

1. Robot Mode - This section controls the Robot Mode. Practice Mode causes the robot to cycle through the same transitions as an FRC match after the Enable button is pressed (timing for practice mode can be found on the setup tab).
2. Enable/Disable - These controls enable and disable the robot. You can also use the key combination [J]\ (the 3 keys above the enter key on most keyboards) to enable the robot and the Enter key to Disable the robot. **The Spacebar will Emergency Stop the Robot**
3. Elapsed Time - Indicates the amount of time the robot has been enabled
4. PC Battery - Indicates current state of DS PC battery and whether the PC is plugged in
5. PC CPU% - Indicates the CPU Utilization of the DS PC
6. Window Mode - When not on the Driver account on the Classmate allows the user to toggle between floating (arrow) and docked (rectangle)
7. Team Station - When not connected to FMS, sets the team station to transmit to the robot.

Note: When connected to the Field Management System the controls in sections 1, and 2 will be replaced by the words FMS Connected and the control in Section 7 will be greyed out.

# Getting Started with the Control System

## Diagnostics Tab



The Diagnostics Tab contains additional status indicators that teams can use to diagnose issues with their robot:

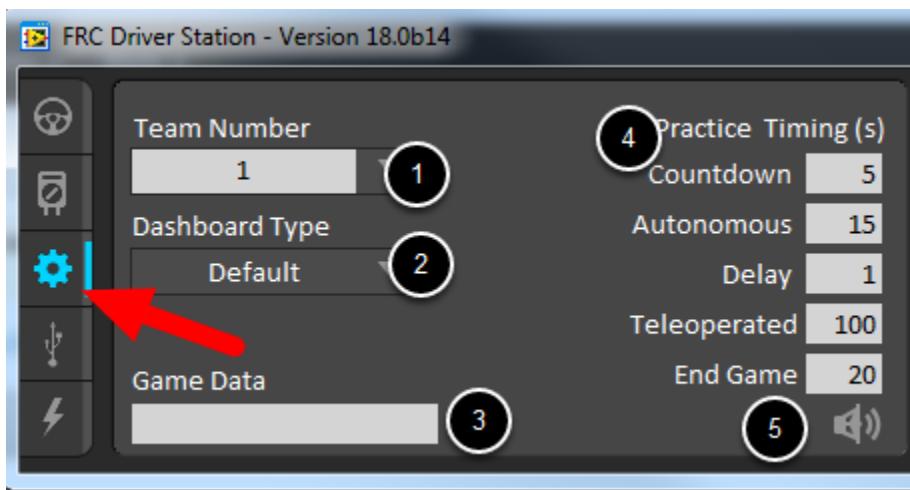
1. DS Version - Indicates the Driver Station Version number
2. roboRIO Image Version - String indicating the version of the roboRIO Image
3. WPIlib Version - String indicating the version of WPIlib in use
4. CAN Device Versions - String indicating the firmware version of devices connected to the CAN bus. These items may not be present if the webdash plugin has not been installed using [CTRE Phoenix Lifeboat](#)
5. Memory Stats - This section shows stats about the roboRIO memory
6. Connection Indicators - The top half of these indicators show connection status to various components.
  - "Enet Link" indicates the computer has something connected to the ethernet port.
  - "Robot Radio" indicates the ping status to the robot wireless bridge at 10.XX.YY.1.
  - "Robot" indicates the ping status to the roboRIO using mDNS (with a fallback of a static 10.TE.AM.2 address).
  - "FMS" indicates if the DS is receiving packets from FMS (this is NOT a ping indicator).
7. Network Indicators - The second section of indicators indicates status of network adapters and firewalls. **These are provided for informational purposes, communication may be established with one or more unlit indicators in this section**
  - "Enet" indicates the IP address of the detected Ethernet adapter
  - "WiFi" indicates if a wireless adapter has been detected as enabled

# Getting Started with the Control System

- "USB" indicates if a roboRIO USB connection has been detected
- "Firewall" indicates if any firewalls are detected as enabled. Enabled firewalls will show in orange ( Dom = Domain, Pub = Public, Prv = Private)

8. Reboot roboRIO - This button attempts to perform a remote reboot of the roboRIO (after clicking through a confirmation dialog)
9. Restart Robot Code - This button attempts to restart the code running on the robot (but not restart the OS)

## Setup Tab



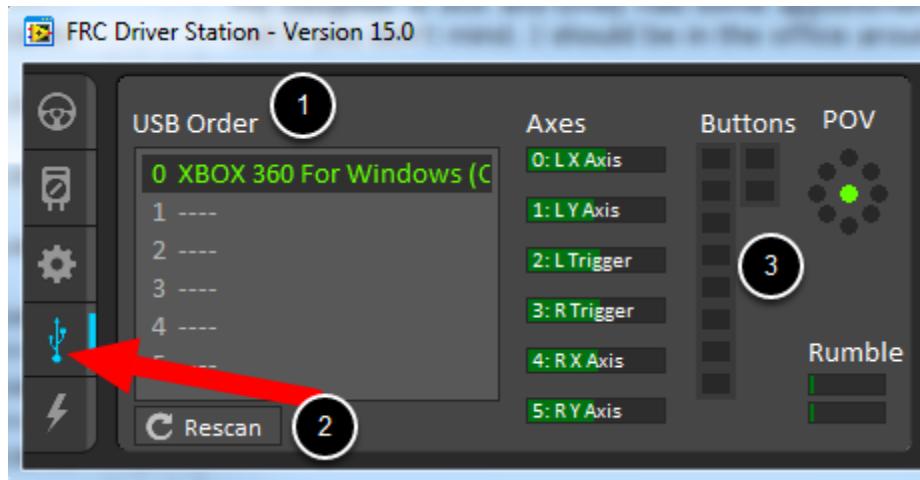
The Setup Tab contains a number of buttons teams can use to control the operation of the Driver Station:

1. Team Number - Should contain your FRC Team Number. This controls the mDNS name that the DS expects the robot to be at. Shift clicking on the dropdown arrow will show all roboRIO names detected on the network for troubleshooting purposes.
2. Dashboard Type - Controls what Dashboard is launched by the Driver Station. **Default** launches the file pointed to by the "FRC DS Data Storage.ini" file, by default this is Dashboard.exe in the Program Files\FRC Dashboard folder. **LabVIEW** attempts to launch a dashboard at the default location for a custom built LabVIEW dashboard, but will fall back to the default if no dashboard is found. [SmartDashboard](#) and [Shuffleboard](#) launch the respective dashboards included with the C++ and Java Eclipse plugins.
3. Game Data - This box can be used for at home testing of the Game Data API. Text entered into this box will appear in the Game Data API on the Robot Side. When connected to FMS, this data will be populated by the field automatically.

# Getting Started with the Control System

4. Practice Mode Timing - These boxes control the timing of each portion of the practice mode sequence. When the robot is enabled in practice mode the DS automatically proceeds through the modes indicated from top to bottom.
5. Audio Control - This button controls whether audio tones are sounded when the Practice Mode is used.

## USB Devices Tab

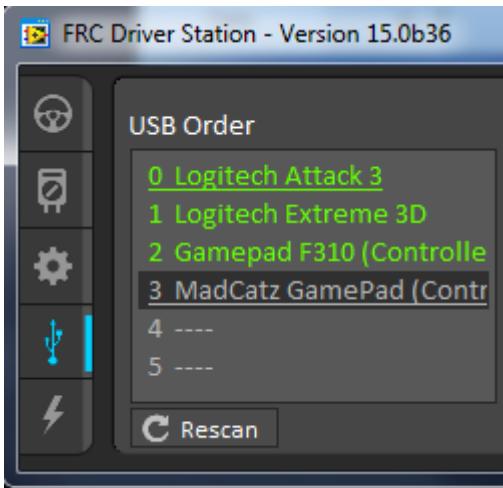


The USB Devices tab includes the information about the USB Devices connected to the DS

1. USB Setup List - This contains a list of all compatible USB devices connected to the DS. Pressing a button on a device will highlight the name in green and put 2 \*s before the device name
2. Rescan - This button will force a Rescan of the USB devices. While the robot is disabled, the DS will automatically scan for new devices and add them to the list. To force a complete re-scan or to re-scan while the robot is Enabled (such as when connected to FMS during a match) press F1 or use this button.
3. Device indicators - These indicators show the current status of the Axes, buttons and POV of the joystick.
4. Rumble - For XInput devices (such as X-Box controllers) the Rumble control will appear. This can be used to test the rumble functionality of the device. The top bar is "Right Rumble" and the bottom bar is "Left Rumble". Clicking and holding anywhere along the bar will activate the rumble proportionally (left is no rumble = 0, right is full rumble = 1). This is a control only and will not indicate the Rumble value set in robot code.

# Getting Started with the Control System

## Re-Arranging and Locking Devices



The Driver Station has the capability of "locking" a USB device into a specific slot. This is done automatically if the device is dragged to a new position and can also be triggered by double clicking on the device. "Locked" devices will show up with an underline under the device. A locked device will reserve it's slot even when the device is not connected to the computer (shown as grayed out and underlined). Devices can be unlocked (and unconnected devices removed) by double clicking on the entry.

Note: If you have two or more of the same device, they should maintain their position as long as all devices remain plugged into the computer in the same ports they were locked in. If you switch the ports of two identical devices the lock should follow the port, not the device. If you re-arrange the ports (take one device and plug it into a new port instead of swapping) the behavior is not determinate (the devices may swap slots). If you unplug one or more of the set of devices, the positions of the others may move, they should return to the proper locked slots when all devices are reconnected.

Example: The image above shows 4 devices:

- A Locked "Logitech Attack 3" joystick. This device will stay in this position unless dragged somewhere else or unlocked
- An unlocked "Logitech Extreme 3D" joystick
- An unlocked "Gamepad F310 (Controller)" which is a Logitech F310 gamepad
- A Locked, but disconnected "MadCatz GamePad (Controller)" which is a MadCatz Xbox 360 Controller

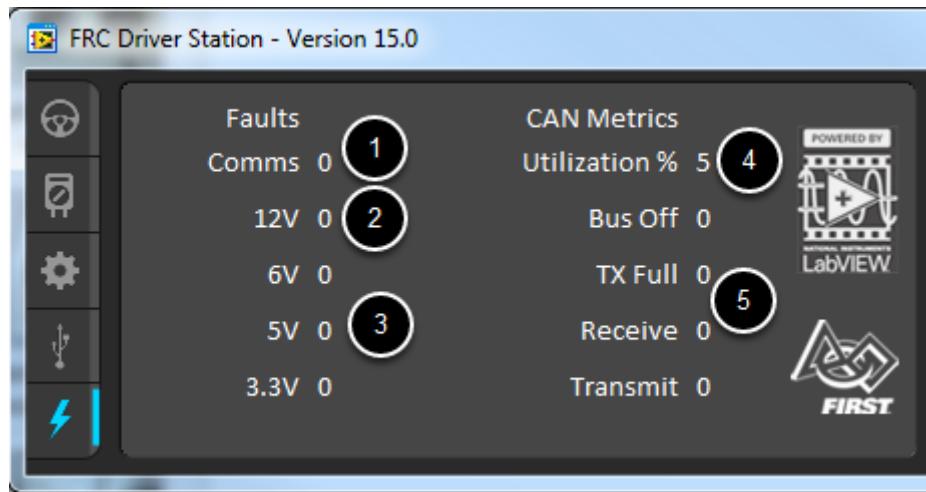
# Getting Started with the Control System

In this example, unplugging the Logitech Extreme 3D joystick will result in the F310 Gamepad moving up to slot 1. Plugging in the MadCatz Gamepad (even if the devices in Slots 1 and 2 are removed and those slots are empty) will result in it occupying Slot 3.

## Joystick Setup Best Practice

**Note:** When using the Re-Arranging and Locking feature described above, teams should take care to make sure devices behave as they expect when the DS is restarted, and when the DS computer is rebooted with the devices connected, after initial setup. XInput devices such as Xbox controller may enumerate differently when they are connected 1-at-a-time versus all at once.

## CAN\Power Tab



The last tab on the left side of the DS is the CAN\Robot Power Tab. This tab contains information about the power status of the roboRIO and the status of the CAN bus:

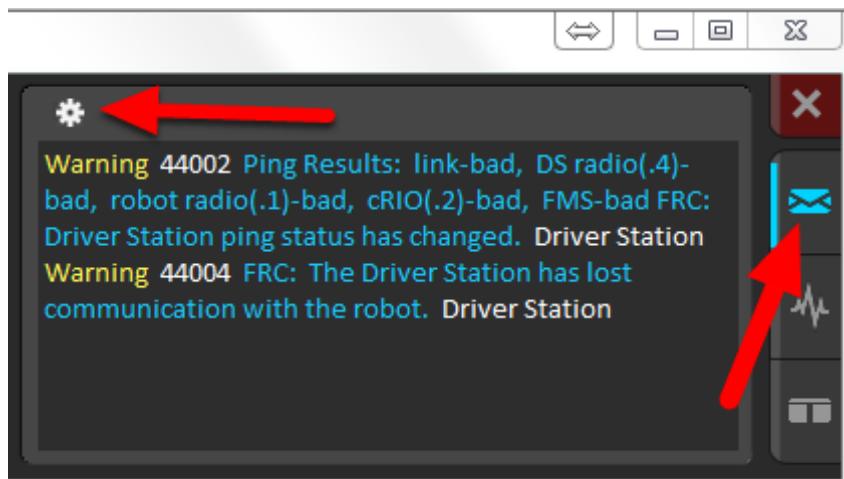
1. Comms Faults - This indicates the number of Comms faults that have occurred since the DS has been connected
2. 12V Faults - This indicates the number of input power faults (Brownouts) that have occurred since the DS has been connected
3. 6V/5V/3.3V Faults - This indicates the number of faults (typically caused by short circuits) that have occurred on the User Voltage Rails since the DS has been connected
4. CAN Bus Utilization - This indicates the percentage utilization of the CAN bus

# Getting Started with the Control System

5. CAN faults - These indicate the counts of each of the 4 types of CAN faults since the DS has been connected

If a fault is detected, the indicator for this tab (shown in blue in the image above) will turn red.

## Messages Tab



The Messages tab displays diagnostic messages from the DS, WPILib, User Code, and/or the roboRIO. The messages are filtered by severity. By default, only Errors are displayed.

To access settings for the Messages tab, click the Gear icon. This will display a menu that will allow you to select the detail level (Errors, Errors+Warnings or Errors+Warnings+Prints), Clear the box, launch a larger Console window for viewing messages, or launch the DS Log Viewer.

# Getting Started with the Control System

## Charts Tab



The Charts tab plots and displays advanced indicators of robot status to help teams diagnose robot issues:

1. The top graph charts trip time in milliseconds in green (against the axis on the right) and lost packets per second in orange (against the axis on the left)
2. The bottom graph plots battery voltage in yellow (against the axis on the left), roboRIO CPU in red (against the axis on the right), DS Requested mode as a continuous line on the bottom of the chart and robot mode as a discontinuous line above it.
3. This key shows the colors used for the DS Requested and Robot Reported modes in the bottom chart.
4. Chart scale - These controls change the time scale of the DS Charts
5. This button launches the DS Log File Viewer

The DS Requested mode is the mode that the Driver Station is commanding the robot to be in. The Robot Reported mode is what code is actually running based on reporting methods contained in the coding frameworks for each language.

## Both Tab

The last tab on the right side is the Both tab which displays Messages and Charts side by side

# Getting Started with the Control System

## Driver Station Keys

The following keys can be used to control Driver Station operation:

- F1 - Force a Joystick refresh.
- '[' + ']' + '\`' - Enable the robot (the 3 keys above Enter on most keyboards)
- Enter - Disable the Robot
- Space - Emergency Stop the robot. After an emergency stop is triggered the roboRIO will need to be rebooted before the robot can be enabled again. **Note: This will E-Stop the robot regardless of if the Driver Station window has focus or not**

# Running your Benchtop Test Program - Tethered

Running your benchtop testing program while tethered to the Driver Station via ethernet or USB cable will confirm the the program was successfully deployed and that the driver station and roboRIO are properly configured.

## Overview

You should create and download a Benchtop Test Program as described for your programming language:

- [C++](#)
- [Java](#)
- [LabVIEW](#)

The roboRIO should be powered on and connected to the PC over Ethernet or USB. The Driver Station software should be configured with your team number as described in the [previous article](#).

## Confirm Connectivity

Confirm Connectivity

Using the Driver Station software

Click Diagnostics and confirm that the Enet Link and Robot leds are green.

## Tethered Operation

Tethered Operation

Click the Operation Tab

## Getting Started with the Control System

1. Confirm that battery voltage is displayed
2. Communications, Robot Code, and Joysticks indicators are green.
3. Put the robot in Teleop Mode
4. Click **Enable**.- Move the joysticks and observe how the robot responds.
5. Click **Disable**

# Programming your radio

This guide will show you how to use the FRC Radio Configuration Utility software to configure your robot's wireless bridge for use outside of FRC events.

Before you begin using the software:

1. Disable WiFi connections on your computer, as it may prevent the configuration utility from properly communicating with the bridge
2. Make sure no devices are connected to your computer via ethernet, other than the wireless bridge. **Note that for the OM5P-AN and AC bridge, it is recommended to use a particular Ethernet port. See the on screen image and instructions for more information.**

The OM5P-AN and AC use the same power plug as the D-Link DAP1522, however they are 12V radios. Wire the radio to the 12V 2A terminals on the VRM (center-pin positive).

**Note: Teams will need to update firmware on both OM5P-AN and OM5P-AC radios in order for the programming utility to program them, or for them to be used at events. This must be done before you attempt to program them.**

## Pre-Requisites

The FRC Radio Configuration Utility requires the Java Runtime Engine (JRE). If you do not have Java installed, you can download the JRE from here: <https://www.java.com/en/download/>

The FRC Radio Configuration Utility requires Administrator privileges to configure the network settings on your machine. The program should request the necessary privileges automatically (may require a password if run from a non-Administrator account), but if you are having trouble, try running it from an Administrator account.

# Getting Started with the Control System

## Application Notes

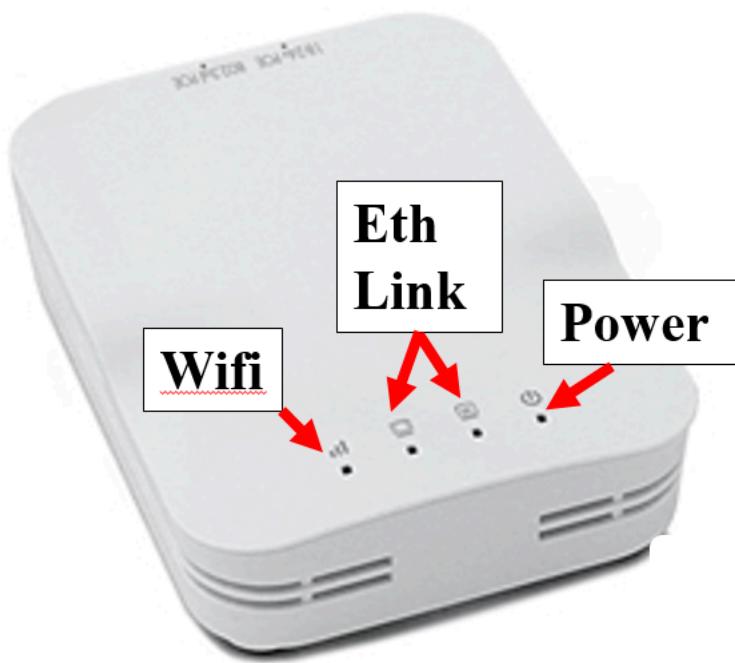
By default, the Radio Configuration Utility will program the radio to enforce the 4Mbps bandwidth limit on traffic exiting the radio over the wireless interface. In the home configuration (AP mode) this is a total, not a per client limit. This means that streaming video to multiple clients is not recommended.

The Utility has been tested on Windows 7, 8 and 10. It may work on other operating systems, but has not been tested.

## Programmed Configuration

Power	
Blue	On or Powering Up
Blue Blinking	Powering Up
Eth Link	
Blue	Link Up
Blue Blinking	Traffic Present
WiFi	
Off	
Off	Bridge Mode, Unlinked or non-FRC firmware
Red	AP, Unlinked
Yellow\Orange	AP, Linked
Green	Bridge Mode, Linked

**WiFi light only works after radio has been power cycled.**



The Radio Configuration Utility programs a number of configuration settings into the radio when run. These settings apply to the radio in all modes (including at events). These include:

- Set a static IP of 10.TE.AM.1
- Set an alternate IP on the wired side of 192.168.1.1 for future programming
- Bridge the wired ports so they may be used interchangeably
- The LED configuration noted in the graphic above.

# Getting Started with the Control System

- 4Mb/s bandwidth limit on the outbound side of the wireless interface (may be disabled for home use)
- QoS rules for internal packet prioritization (affects internal buffer and which packets to discard if bandwidth limit is reached). These rules are Robot Control and Status (UDP 1110, 1115, 1150) >> Robot TCP & Network Tables (TCP 1735, 1740) >> Bulk (All other traffic). (disabled if BW limit is disabled)
- DHCP server enabled. Serves out 10.TE.AM.11 - 10.TE.AM.111 on the wired side, 10.TE.AM.130 - 10.TE.AM.230 on the wireless side, subnet mask of 255.255.255.0, broadcast address 10.TE.AM.255
- DNS server enabled. DNS server IP and domain suffix (.lan) are served as part of the DHCP.

At home only:

- SSID may have a "Robot Name" appended to the team number to distinguish multiple networks.
- Firewall option may be enabled to mimic the field firewall rules (open ports may be found in the Game Manual)

**Note: It is not possible to modify the configuration manually**

## Download the software

Download the latest FRC Radio Configuration Utility Installer from the following links:

[FRC Radio Configuration 19.1.1](#)

[FRC Radio Configuration 19.1.1 Israel Version](#)

**Note: The \_IL version is for Israel teams and contains a version of the OM5PAC firmware with restricted channels for use in Israel.**

**⚠** Version 19.1.1 corrects an issue with applying the Bandwidth Limit present in version 19.1.0. Teams should install the new version, then re-program their radio (there is no need to re-flash the firmware).

# Getting Started with the Control System

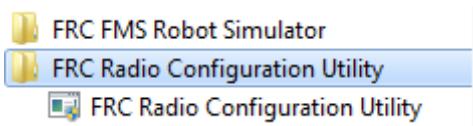
## Install the software



Double click on FRC\_Radio\_Configuration\_VERSION.exe to launch the installer. Follow the prompts to complete the installation.

Part of the installation prompts will include installing WinPCap if it is not already present. The WinPCap installer contains a checkbox (checked by default) to start the WinPCap driver on boot. You should leave this box checked.

## Launch the software

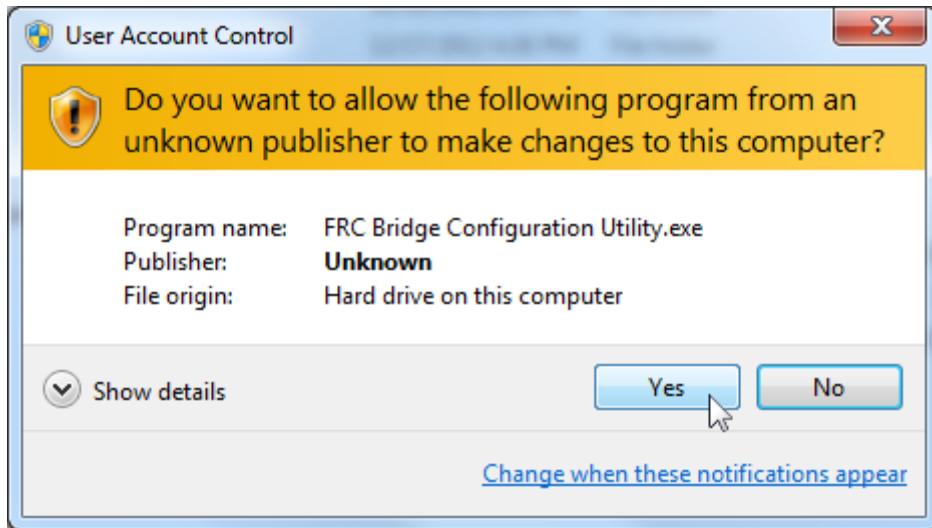


Use the Start menu or desktop shortcut to launch the program.

Note: If you need to locate the program it is installed to C:\Program Files (x86)\FRC Radio Configuration Utility. For 32-bit machines the path is C:\Program Files\FRC Radio Configuration Utility\

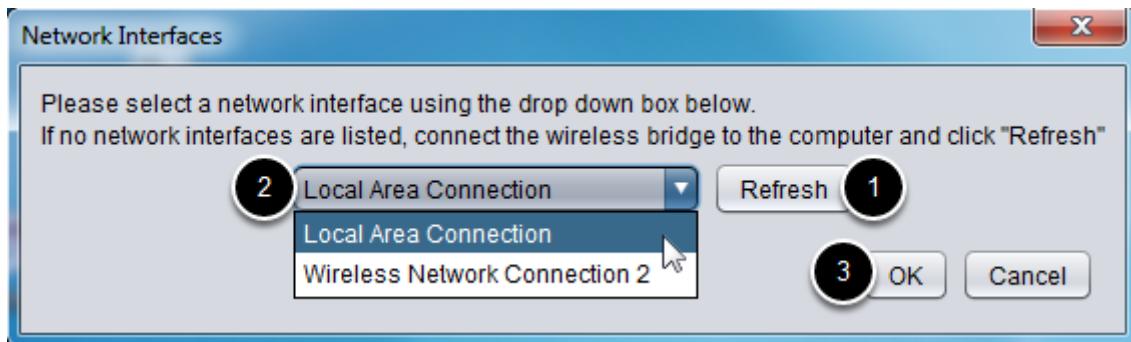
# Getting Started with the Control System

## Allow the program to make changes, if prompted



If your computer is running Windows Vista or Windows 7, a prompt may appear about allowing the configuration utility to make changes to the computer. Click "Yes" if the prompt appears.

## Select the network interface



Use the pop-up window to select the which ethernet interface the configuration utility will use to communicate with the wireless bridge. On Windows machines, ethernet interfaces are typically named "Local Area Connection". The configuration utility can not program a bridge over a wireless connection.

1. If no ethernet interfaces are listed, click "Refresh" to re-scan for available interfaces

# Getting Started with the Control System

2. Select the interface you want to use from the drop-down list
3. Click "OK"

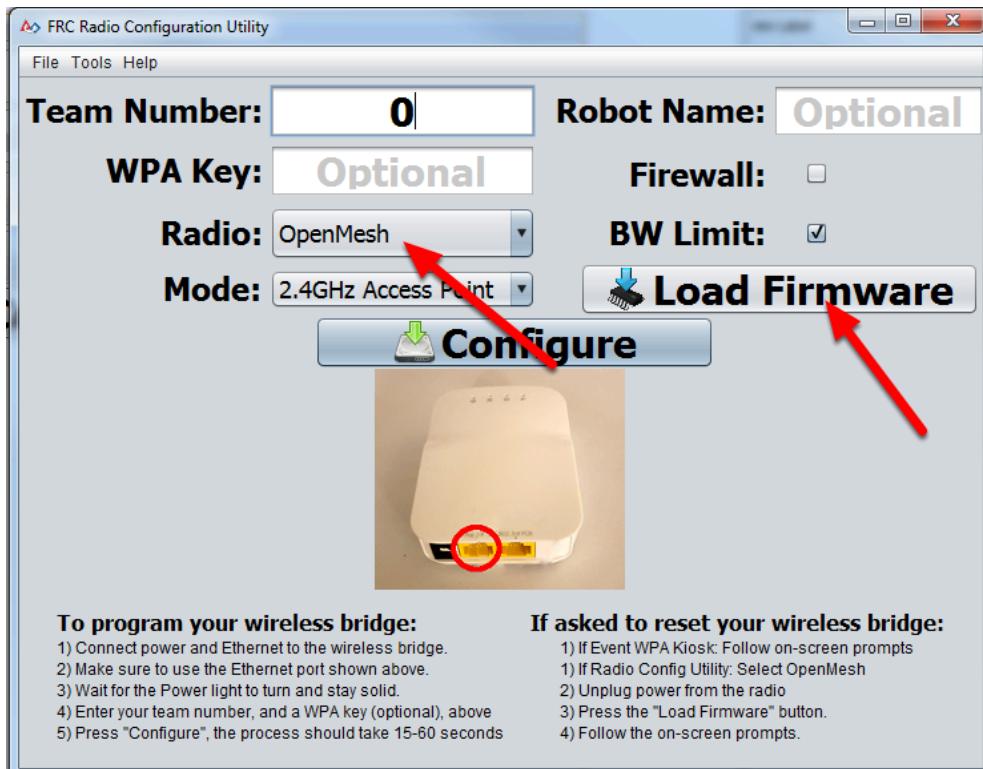
## Open Mesh Firmware Note

For the FRC Radio Configuration Utility to program the OM5P-AN and OM5P-AC radio, the radio must be running an FRC specific build of the OpenWRT firmware. **OM5P-AC radios in the 2019 KoP should not need an update.**

If you do not need to update or re-load the firmware, skip the next step.

 Note: Radios used in 2018 will need to be updated to 2019 firmware before configuring.

## Loading FRC Firmware to OpenMesh radio



# Getting Started with the Control System

If you need to load the FRC firmware (or reset the radio), you can do so using the FRC Radio Configuration Utility.

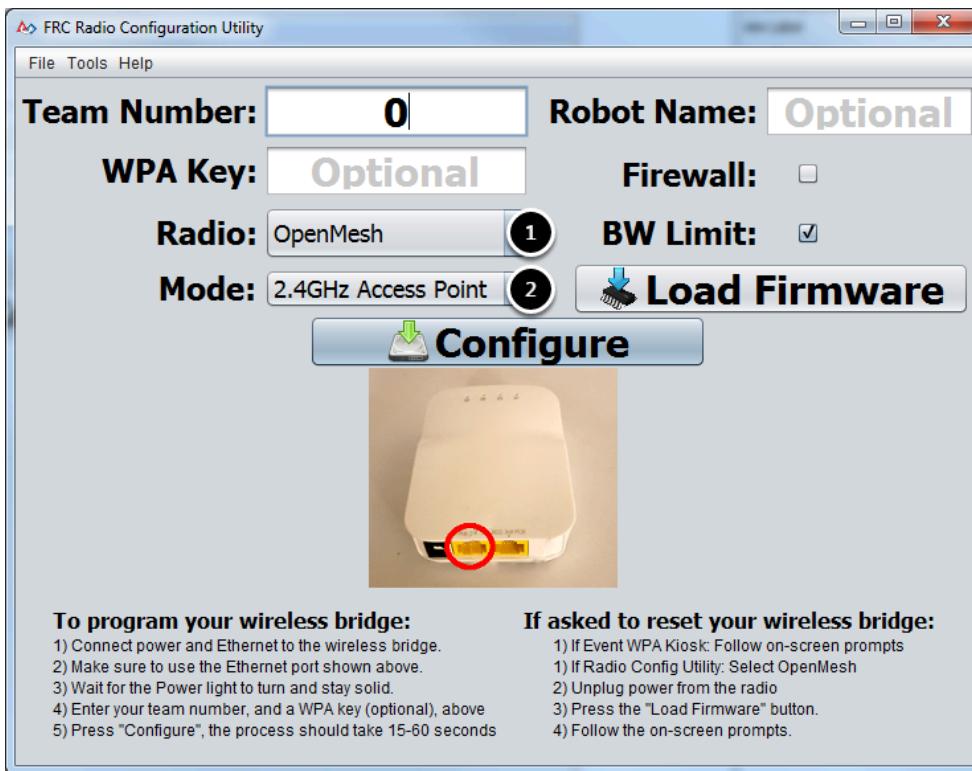
1. Follow the instructions above to install the software, launch the program and select the Ethernet interface.
2. Make sure the OpenMesh radio is selected in the Radio dropdown.
3. Make sure the radio is connected to the PC via Ethernet.
4. Unplug the power from the radio. (If using a PoE cable, this will also be unplugging the Ethernet to the PC, this is fine)
5. Press the Load Firmware button
6. When prompted, plug in the radio power. The software should detect the radio, load the firmware and prompt you when complete.

 If you see an error about NPF name, try disabling all adapters other than the one being used to program the radio. If only one adapter is found, the tool should attempt to use that one. See the [steps at the bottom of this article](#) for more info.

Teams may also see this error with foreign language Operating Systems. If you experience issues loading firmware or programming on a foreign language OS, try using an English OS, such as on the KOP provided PC or setting the Locale setting to "en\_us" as described on [this page](#).

# Getting Started with the Control System

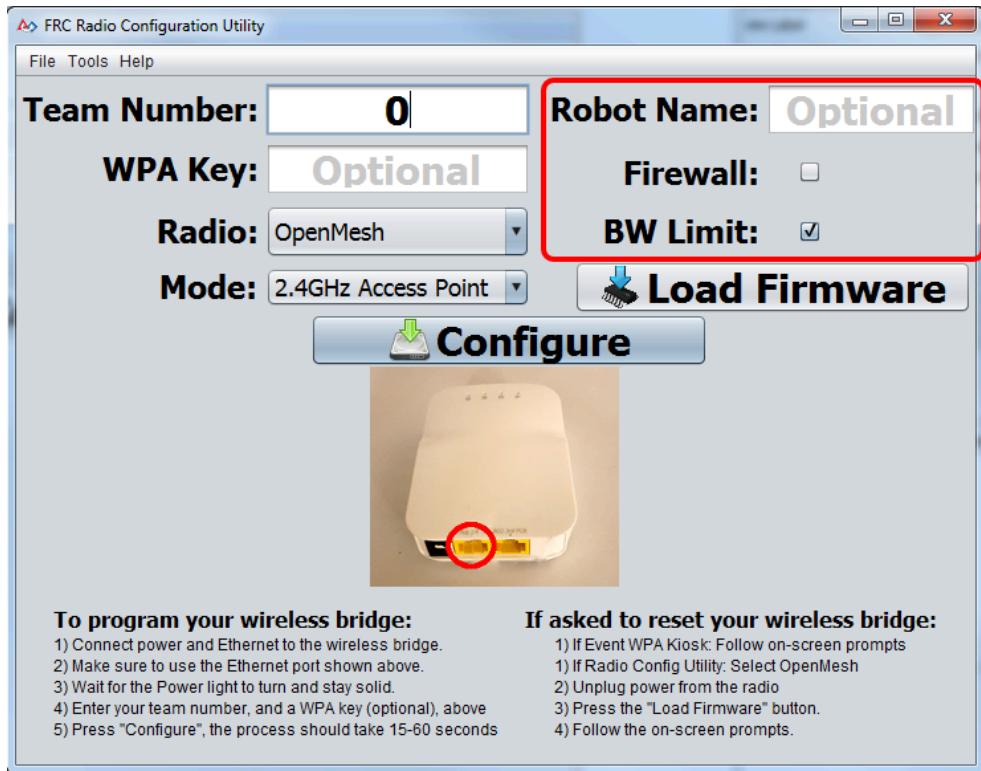
## Select a bridge model and operating mode



1. Select which radio you are configuring using the drop-down list.
2. Select which operating mode you want to configure. For most cases, the default selection of 2.4GHz Access Point will be sufficient. If your computers support it, the 5GHz AP mode is recommended, as 5GHz is less congested in many environments.

# Getting Started with the Control System

## Select Options



The default values of the options have been selected to match the use case of most teams, however, you may wish to customize these options to your specific scenario:

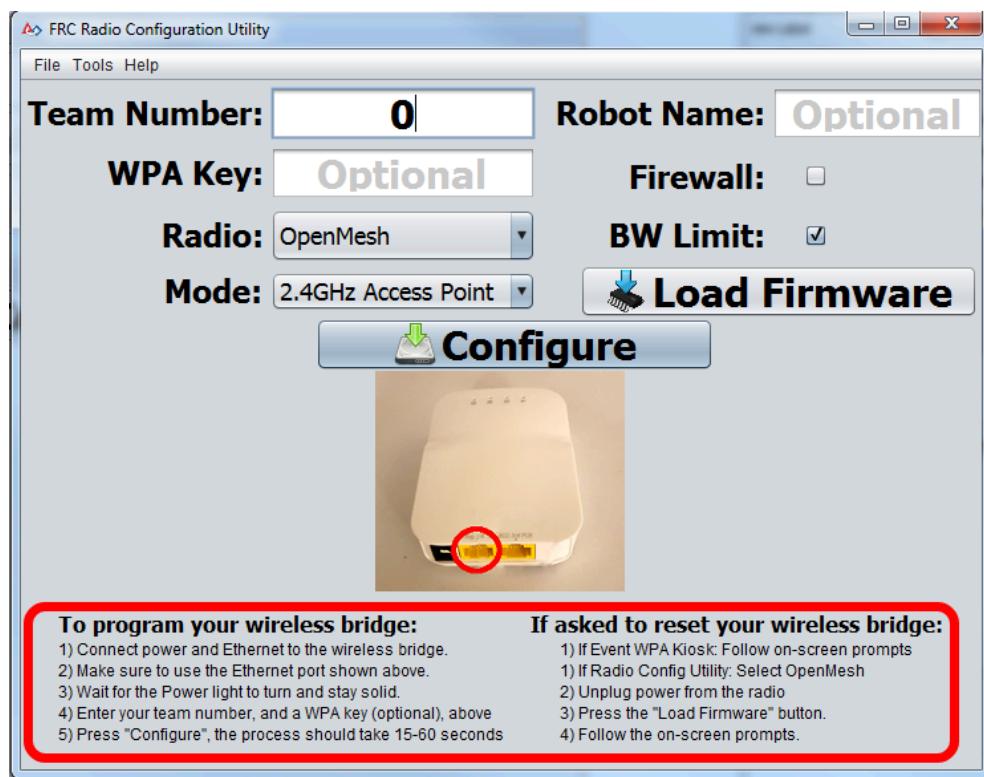
1. Robot Name: This is a string that gets appended to the SSID used by the radio. This allows you to have multiple networks with the same team number and still be able to distinguish them.
2. Firewall: If this box is checked, the radio firewall will be configured to attempt to mimic the port blocking behavior of the firewall present on the FRC field. For a list of open ports, please see the FRC Game Manual.
3. BW Limit: If this box is checked, the radio enforces a 4MB/s bandwidth limit like it does when programmed at events. Note that in AP mode, this is a total limit, not per client, so streaming video to multiple clients simultaneously may cause undesired behavior.

**Note: Firewall and BW Limit only apply to the OpenMesh radios. These options have no effect on D-Link radios.**

# Getting Started with the Control System

**⚠** The "Firewall" option configures the radio to emulate the field firewall. This means that you will not be able to deploy code wirelessly with this option enabled.

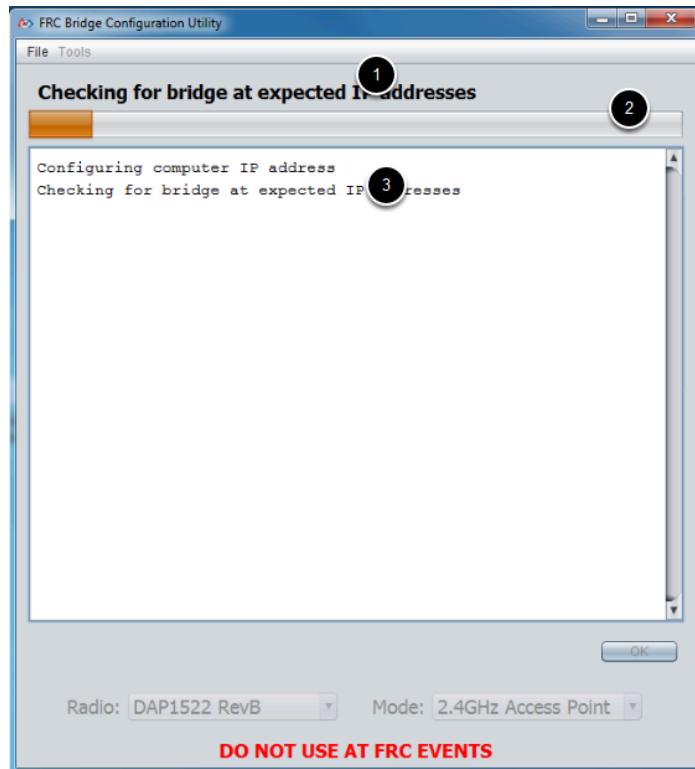
## Prepare and start the configuration process



Follow the on-screen instructions for preparing your wireless bridge, entering the settings the bridge will be configured with, and starting the configuration process. These on-screen instructions update to match the bridge model and operating mode chosen.

# Getting Started with the Control System

## Configuration Progress

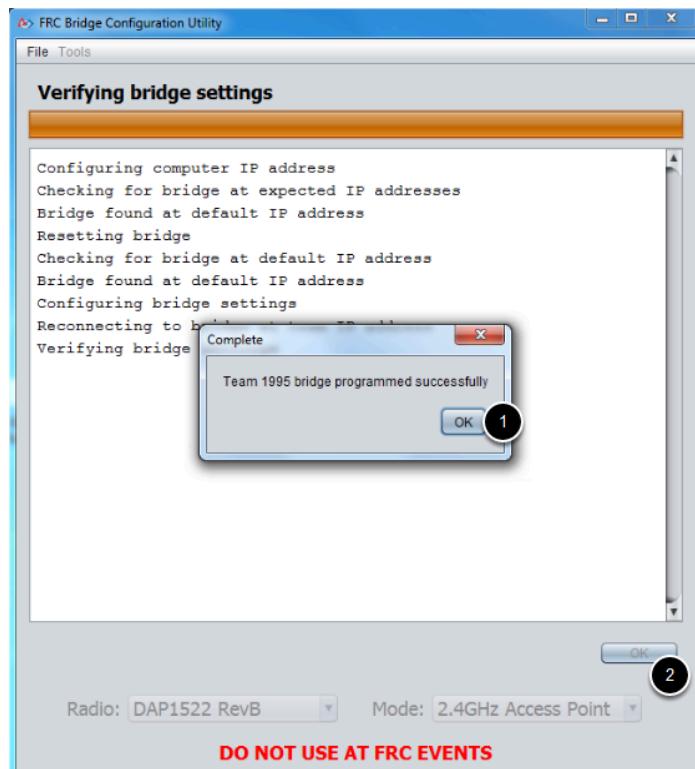


Throughout the configuration process, the window will indicate:

1. The step currently being executed
2. The overall progress of the configuration process
3. All steps executed so far

# Getting Started with the Control System

## Configuration completed

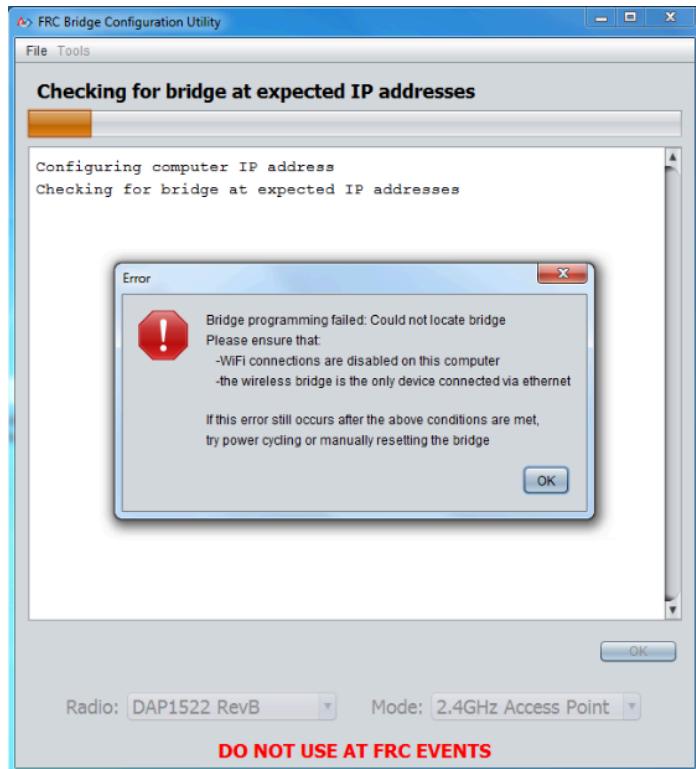


Once the configuration is complete:

1. Press "OK" on the dialog window
2. Press "OK" on the main window to return to the settings screen

# Getting Started with the Control System

## Configuration errors

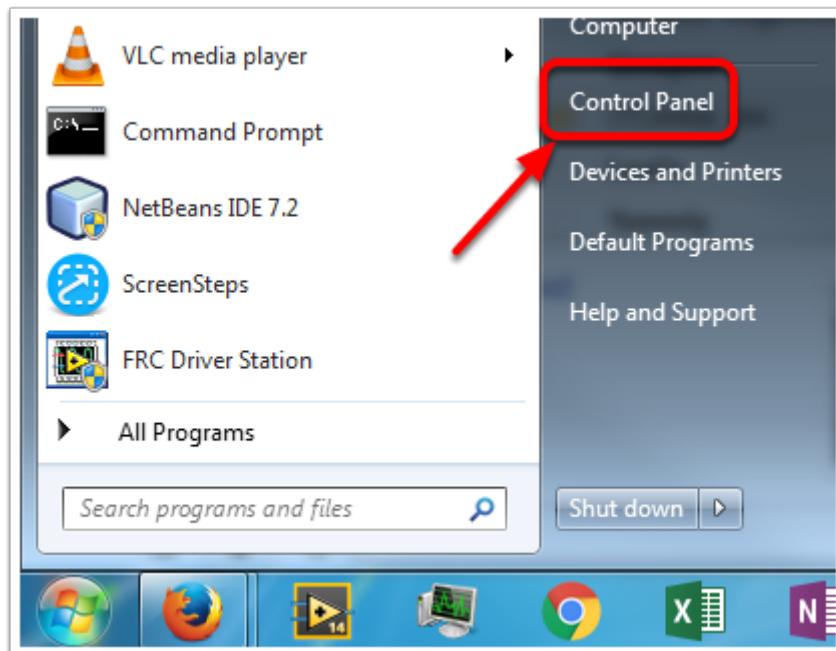


If an error occurs during the configuration process, follow the instructions in the error message to correct the problem.

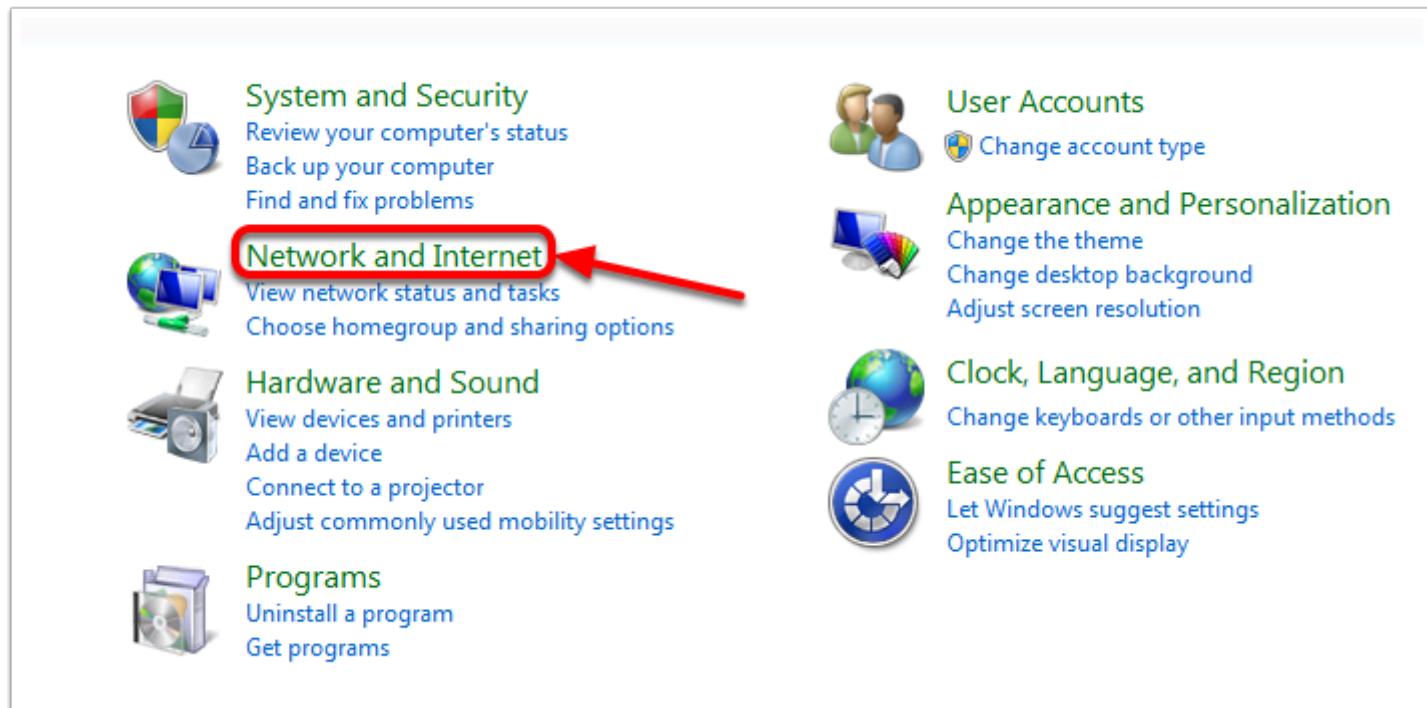
## Troubleshooting: Disabling Network Adapters

If you get an error message about "NPF adapter" when attempting to load firmware, you need to disable all other adapters. This is not always the same as turning the adapters off with a physical button or putting the PC into airplane mode. The following steps provide more detail on how to disable adapters.

# Getting Started with the Control System

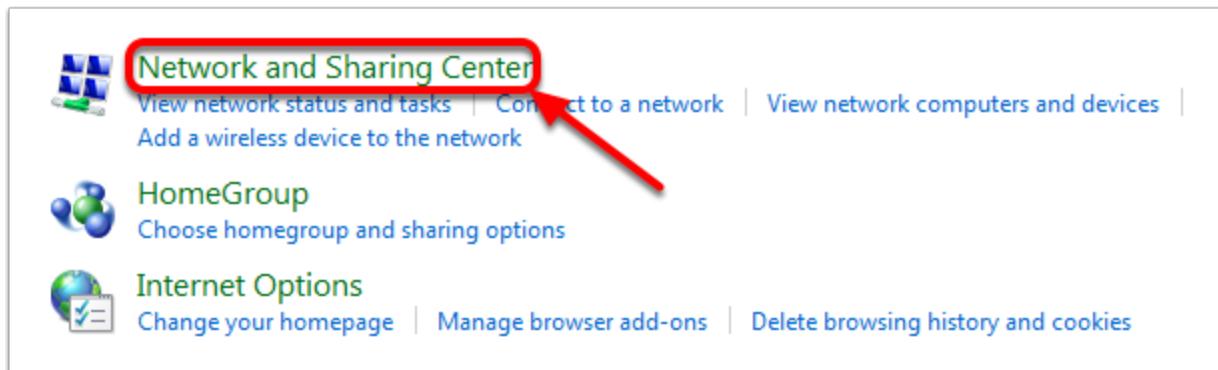


Open the Control Panel by going to **Start->Control Panel**

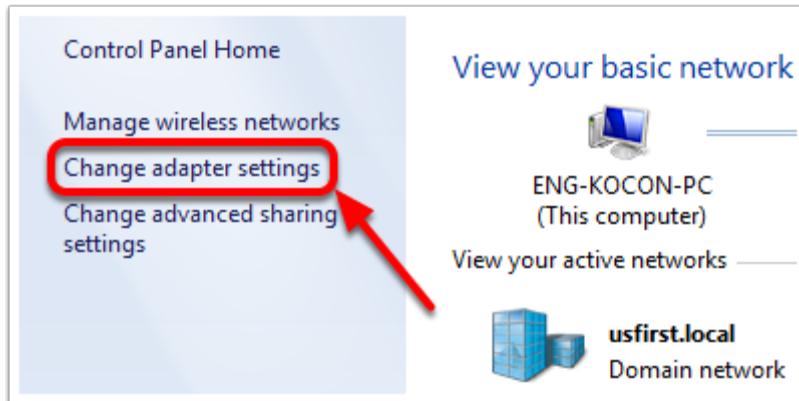


Choose the **Network and Internet** category.

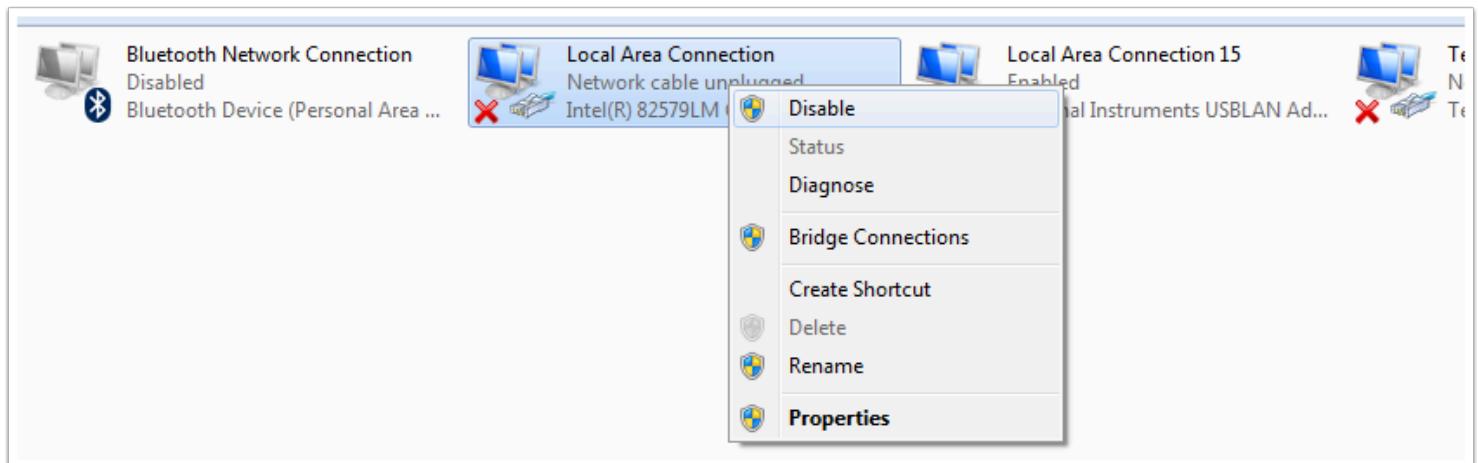
# Getting Started with the Control System



Click **Network and Sharing Center**



On the left pane, click **Change Adapter Settings**



## Getting Started with the Control System

For each adapter other than the one connected to the radio, right click on the adapter and select **Disable** from the menu.

## Getting Started with the Control System

# Running your Benchtop Test Program - Wireless

Before attempting wireless operation, tethered operation should have been confirmed as described in [Running your Benchtop Test Program - Tethered](#). Running your benchtop testing program while connected to the Driver Station via WiFi will confirm that the access point is properly configured

## Configuring the access point

See the article [Programming your radio for home use](#) for details on configuring the robot radio for use as an access point.

After configuring the access point, connect the driver station wirelessly to the robot. The SSID will be your team number (as entered in the Bridge Configuration Utility). If you set a key when using the Bridge Configuration Utility you will need to enter it to connect to the network. Make sure the computer network adapter is set to DHCP ("Obtain an IP address automatically").

## Confirm Connectivity

Confirm Connectivity

Using the Driver Station software

Click Diagnostics and confirm that the Bridge, and Robot LEDs are green.

## Wireless Operation

Wireless Operation

Click the Operation Tab

# Getting Started with the Control System

1. Confirm that battery voltage is displayed
2. Communications, Robot Code, and Joysticks indicators are green.
3. Put the robot in Teleop Mode
4. Click **Enable**.- Move the joysticks and observe how the robot responds.
5. Click **Disable**

# Updating and Configuring Pneumatics Control Module and Power Distribution Panel

- ! The CAN Web Dashboard plugin is no longer supported by the roboRIO Web Dashboard. To configure CTRE CAN devices such as the PCM and PDP, use CTRE Phoenix Tuner:  
[https://phoenix-documentation.readthedocs.io/en/latest/ch03\\_PrimerPhoenixSoft.html#what-is-phoenix-tuner](https://phoenix-documentation.readthedocs.io/en/latest/ch03_PrimerPhoenixSoft.html#what-is-phoenix-tuner)

# Getting Started with the Control System

## Getting Started

### Download FRC 2017 Software and Setup Guide

#### [roboRIO Details and Specifications](#)

- roboRIO User Manual and Specifications Documents
- CAD Models
- MXP Developer's Guide

#### [FRC Training Material and Resources](#)

- Basics of an FRC System
- Concepts for Programming and Building Robots
- Bringing Your Robot to Life

#### [FRC LabVIEW Quick Start Guide](#)

- LabVIEW Basics
- Vision, PID, and Simulation Resources
- Advanced Programming Resources

#### [Setting Up and Imaging Your roboRIO](#)

- Communicating With roboRIO
- How to Image roboRIO
- Additional Troubleshooting

#### [LabVIEW and roboRIO Training by our partner, Intelitek](#)

# Getting Started with the Control System

[Need more help? Click Here!](#)

## Getting Started with the Control System

# System Details

# Getting Started with the Control System

## RoboRIO Networking

The network setup used on the roboRIO system is a little bit different than the previous Control System. The new scheme utilizes mDNS to allow for the use of DHCP addressing and seamless transition from ethernet to USB and back.

This document discusses the typical setup at home. For more information about the networking environment at events, or about using Static IPs see [IP Networking at the Event](#)

### mDNS

The FRC Driver Station, LabVIEW, and the Eclipse plugins for C++ and Java are all programmed to discover your roboRIO using the mDNS protocol. This means that the roboRIO can be detected regardless of the interface or IP being used.

### mDNS - Principles

Multicast Domain Name System (mDNS) is a system which allows for resolution of host names to IP addresses on small networks with no dedicated name server. To resolve a host name a device sends out a multicast message to the network querying for the device. The device then responds with a multicast message containing it's IP. Devices on the network can store this information in a cache so subsequent requests for this address can be resolved from the cache without repeating the network query.

### mDNS - Providers

To use mDNS, an mDNS implementation is required to be installed on your PC. Here are some common mDNS implementations for each major platform:

Windows:

- NI mDNS Responder - **Installed with the NI FRC Update Suite**
- Apple Bonjour - Installed with iTunes

# Getting Started with the Control System

OSX:

- Apple Bonjour - **Installed by default**

Linux:

- nss-mDNS/Avahi/Zeroconf - Installed and enabled by default on some Linux variants (such as Ubuntu or Mint). May need to be installed or enabled on others (such as Arch)

## mDNS - Firewalls

To work properly mDNS must be allowed to pass through your firewall. **Depending on your PC configuration, no changes may be required, this section is provided to assist with troubleshooting.** Because the network traffic comes from the mDNS implementation and not directly from the Driver Station or IDE, allowing those applications through may not be sufficient. There are two main ways to resolve mDNS firewall issues:

- Add an application/service exception for the mDNS implementation (NI mDNS Responder is C:\Program Files\National Instruments\Shared\mDNS Responder\nimdnsResponder.exe)
- Add a port exception for traffic to/from UDP 5353 (IP ranges 10.0.0.0-10.255.255.255  
172.16.0.0-172.31.255.255 192.168.0.0-192.168.255.255 169.254.0.0-169.254.255.255  
224.0.0.251)

## mDNS - Browser support

Most web-browsers should be able to utilize the mDNS address to access the roboRIO webserver as long as an mDNS provider is installed. To access the webdashboard, the browser must also support Microsoft Silverlight. Internet Explorer is recommended.

## USB

If using the USB interface, no network setup is required (you do need the [NI Update Suite](#) installed to provide the roboRIO USB Driver). The roboRIO driver will automatically configure the IP address of the host (your computer) and roboRIO and the software listed above should be able to locate and utilize your roboRIO

# Getting Started with the Control System

## Ethernet/Wireless

The FRC Radio Configuration Utility will enable the DHCP server on the OpenMesh radio in the home use case (AP mode), if you are putting the OpenMesh in bridge mode and using a router, you can enable DHCP addressing on the router. The bridge is set to the same team based IP address as before (10.TE.AM.1) and will hand out DHCP address from 10.TE.AM.20 to 10.TE.AM.199. When connected to the field, FMS will also hand out addresses in the same IP range.

## roboRIO Ethernet Configuration

**The roboRIO Ethernet interface should be set to DHCP.** When connected to the OpenMesh bridge, the roboRIO will receive an IP from the bridge. When tethered directly to a PC, both devices will self-assign IPs.

## PC Adapter Configuration

**When connecting via Ethernet (to either the radio or directly to the roboRIO) or Wireless (to the OpenMesh radio), your computer adapter should be set to DHCP.** When connecting through the OpenMesh, your PC will receive an IP address from the radio. If tethered directly to the roboRIO both devices will self-assign IPs.

## IP Lists

IPs for system components:

roboRIO USB: 172.22.11.2

roboRIO mDNS: roboRIO-####-FRC.local (where #### is your team number with no leading zeroes) You should be able to use this address to communicate with the roboRIO over either interface through ping, browser, etc.

Robot Radio: 10.TE.AM.1 (where TE.AM is your 4 digit team number with leading zeroes if required)

roboRIO Ethernet: DHCP, assigned by the Robot Radio

Driver Station PC: DHCP, assigned by the Robot Radio

# Getting Started with the Control System

Additional Programming computers: DHCP, assigned by the Robot Radio

DHCP range: 10.TE.AM.20 to 10.TE.AM.199

## Troubleshooting

See [RoboRIO Network Troubleshooting](#)

# Getting Started with the Control System

## Networking Basics

This document provides a brief overview of basic networking principles and how they apply to the FRC hardware.

### Table of Contents

#### Networking Basics

##### Overview

#### IP Addressing Background

What is an IP Address?

Public vs Private IP Addresses

How are these addresses assigned?

Dynamically

Statically

What is link-local?

#### IP Addressing for FRC

On the Field

Dynamic Addressing (recommended)

Static Addressing

In the Pits

Dynamic Addressing (recommended)

Static Addressing

Mixing Dynamic and Static Configurations

#### mDNS

What is DNS?

DNS for FRC

#### Summary

# Getting Started with the Control System

## Overview

In this document, we're going to provide an overview of the basics of computer networking in order to provide some background on how FRC teams should configure their hardware to communicate at competitions, and at home.

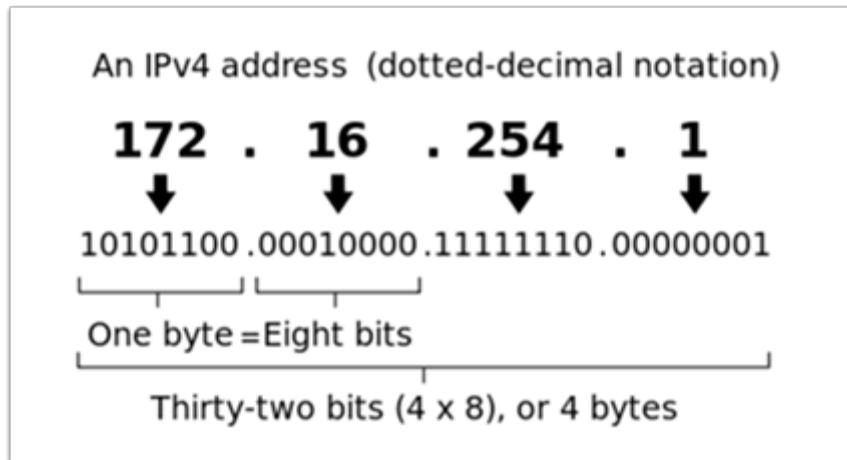
IP Addressing will be the core topic of this training, and the key for understanding how devices are communicating. By understanding how to interpret IP addresses, how they are assigned, how they are organized, and how to set them, teams should be able to configure their devices to successfully communicate.

At the end of this document, we will provide some additional resources and tools to learn more about networking topics, and troubleshooting issues that may arise.

## IP Addressing Background

### What is an IP Address?

An IP address is a unique string of numbers, separated by periods that identifies each device on a network. Each IP address is divided up into 4 sections (octets) ranging from 0-255.



As shown above, this means that each IP address is a 32-bit address meaning there are  $2^{32}$  addresses, or just over 4,000,000,00 addresses possible. However, most of these are used publicly for things like web servers. This brings up our first key point of IP Addressing: **Each device on the network must have a unique IP address**. No two devices can have the same IP address, otherwise collisions will occur.

# Getting Started with the Control System

Since there are only 4-billion addresses, and there are more than 4-billion computers connected to the internet, we need to be as efficient as possible with giving out IP addresses. This brings us to **public vs. private addresses**.

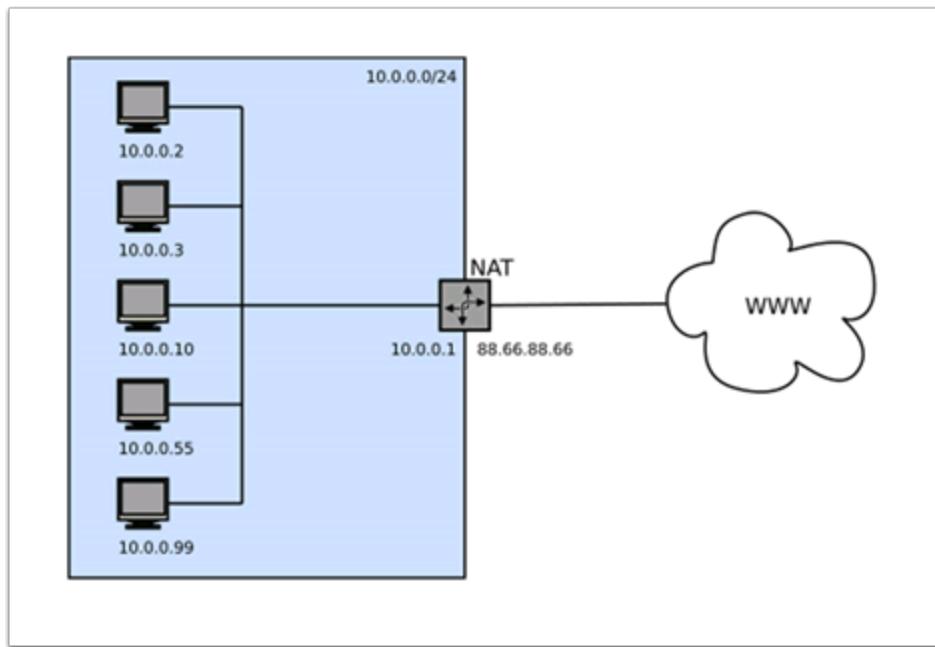
## Public vs Private IP Addresses

To be efficient with using IP Addresses, the idea of “Reserved IP Ranges” was implemented. In short, this means that there are ranges of IP Addresses that will never be assigned to web servers, and will only be used for **local** networks, such as those in your house. Key point #2: **Unless you are directly connecting to your internet provider's basic modem (no router function), your device will have an IP Address in one of these ranges.** This means that at any **local** network, such as: your school, work office, home, etc., your device will 99% of the time have an IP address in a range listed below:

CLASS	BITS	START ADDRESS	END ADDRESS	NUMBER OF ADDRESSES
A	24-BIT BLOCK	10.0.0.0	10.255.255.255	16,777,216
B	20-BIT BLOCK	172.16.0.0	172.31.255.255	1,048,576
C	16-BIT BLOCK	192.168.0.0	192.168.255.255	65,536

These reserved ranges let us assign one “unreserved IP Address” to an entire house, and then use multiple addresses in a reserved range to connect more than one computer to the internet. A process on the home’s internet router known as NAT (Network Address Translation), handles the process of keeping track which private IP is requesting data, using the public IP to request that data from the internet, and then passing the returned data back to the private IP that requested it. This allows us to use the same reserved IP addresses for many local networks, without causing any conflicts. An image of this process is presented below.

# Getting Started with the Control System



**NOTE:** For the FRC networks, we will use the 10.0.0.0 range. This range allows us to use the 10.TE.AM.xx format for IP addresses, whereas using the Class B or C networks would only allow a subset of teams to follow the format. An example of this formatting would be 10.17.50.1 for FRC Team 1750.

## How are these addresses assigned?

We've covered the basics of what IP addresses are, and which IP addresses we will use for the FRC competition, so now we need to discuss how these addresses will get assigned to the devices on our network. We already stated above that we can't have two devices on the same network with the same IP Address, so we need a way to be sure that every device receives an address without overlapping. This can be done **Dynamically** (automatic), or **Statically** (manual).

### Dynamically

Dynamically assigning IP addresses means that we are letting a device on the network manage the IP address assignments. This is done through the Dynamic Host Configuration Protocol (DHCP). DHCP has many components to it, but for the scope of this document, we will think of it as a service that automatically manages the network. Whenever you plug in a new device to the network, the DHCP service sees the new device, then provides it with an available IP address and the other network settings required for the device to communicate. This can mean that there are times we do not know the exact IP address of each device.

# Getting Started with the Control System

## *What is a DHCP Server?*

A DHCP server is a device that runs the DHCP service to monitor the network for new devices to configure. In larger businesses, this could be a dedicated computer running the DHCP service and that computer would be the DHCP server. For home networks, FRC networks, and other smaller networks, the DHCP service is usually running on the router; in this case, the router is the DHCP server.

This means that if you ever run into a situation where you need to have a DHCP server assigning IP addresses to your network devices, it's as simple as finding the closest home router, and plugging it in.

## Statically

Dynamically assigning IP addresses means that we are manually telling each device on the network which IP address we want it to have. This configuration happens through a setting on each device. By disabling DHCP on the network and assigning the addresses manually, we get the benefit of knowing the exact IP address of each device on the network, but because we set each one manually and there is no service keeping track of the used IP addresses, we have to keep track of this ourselves. While statically setting IP addresses, we must be careful not to assign duplicate addresses, and must be sure we are setting the other network settings (such as subnet mask and default gateway) correctly on each device.

## *What is link-local?*

If a device does not have an IP address, then it cannot communicate on a network. This can become an issue if we have a device that is set to dynamically acquire its address from a DHCP server, but there is no DHCP server on the network. An example of this would be when you have a laptop directly connected to a roboRIO and both are set to dynamically acquire an IP address. Neither device is a DHCP server, and since they are the only two devices on the network, they will not be assigned IP addresses automatically.

Link-local addresses give us a standard set of addresses that we can "fall-back" to if a device set to acquire dynamically is not able to acquire an address. If this happens, the device will assign itself an IP address in the 169.254.xx.yy address range; this is a link-local address. In our roboRIO and computer example above, both devices will realize they haven't been assigned an IP address and assign themselves a link-local address. Once they are both assigned addresses in the 169.254.xx.yy

# Getting Started with the Control System

range, they will be in the same network and will be able to communicate, even though they were set to dynamic and a DHCP server did not assign addresses.

## IP Addressing for FRC

For the FRC devices, the recommendation is to use dynamic addressing. As an alternate, static addressing can be used if there is additional software or hardware that does not support mDNS (explained below). If static addressing is used, take care to not set any IP addresses in the range potentially used by the field.

In either configuration, the wireless radio will be statically set to 10.TE.AM.1 by the radio configuration utility. This should not change.

## On the Field

### Dynamic Addressing (recommended)

The field network will be running the DHCP service to assign IP addresses for the team devices. These addresses will be assigned in the ranges of 10.TE.AM.20 10.TE.AM.255. A more complete description is listed below

- Robot Radio Static 10.TE.AM.1 assigned by radio configuration utility
- roboRIO DHCP in the 10.TE.AM.20 10.TE.AM.255 range
- Driver Station DHCP in the 10.TE.AM.20 10.TE.AM.255 range
- IP Camera DHCP in the 10.TE.AM.20 10.TE.AM.255 range
- Other devices DHCP in the 10.TE.AM.20 10.TE.AM.255 range

**NOTE:** It is not required to know the specific address assigned to each device due to the mDNS protocol described in a later section

### Static Addressing

It is also an option statically assign IP addresses to accommodate devices or software which do not support mDNS. When doing so you want to make sure to avoid addresses that will be in use when the robot is on the field network.

### Addresses to avoid:

# Getting Started with the Control System

The addresses listed below should not be used in order to prevent conflict with the field wireless.

- Robot Radio Static 10.TE.AM.1 assigned by radio configuration utility
- Field Access Point Static 10.TE.AM.4 assigned by FRC
- Field DHCP Range 10.TE.AM.20 - 10.TE.AM.255

## Acceptable addresses:

The addresses listed below are recommendations of acceptable addresses for a static setup.

- roboRIO Static 10.TE.AM.2 Subnet Mask of 255.255.255.0
- Driver Station Static 10.TE.AM.5 Subnet Mask of 255.0.0.0
- IP Camera/Other Static 10.TE.AM.6 - 10.TE.AM.19 Subnet Mask of 255.255.255.0

## In the Pits

### Dynamic Addressing (recommended)

Since the robot will not be connected to the field, there will not be a DHCP server present by default in the pits. Most devices should fall back to a link-local address if they are set to DHCP and there is no server present.

If there are still connection issues in the link-local configuration, a team can simulate the field DHCP server by using another router to assign the addresses. The router should be configured with an IP address of 10.TE.AM.4 and assign IP addresses in the range of 10.TE.AM.20 - 10.TE.AM.255.

**NOTE:** If a team is going to use a wireless router in the pits as a DHCP server, the **wireless functionality must be disabled**.

### Static Addressing

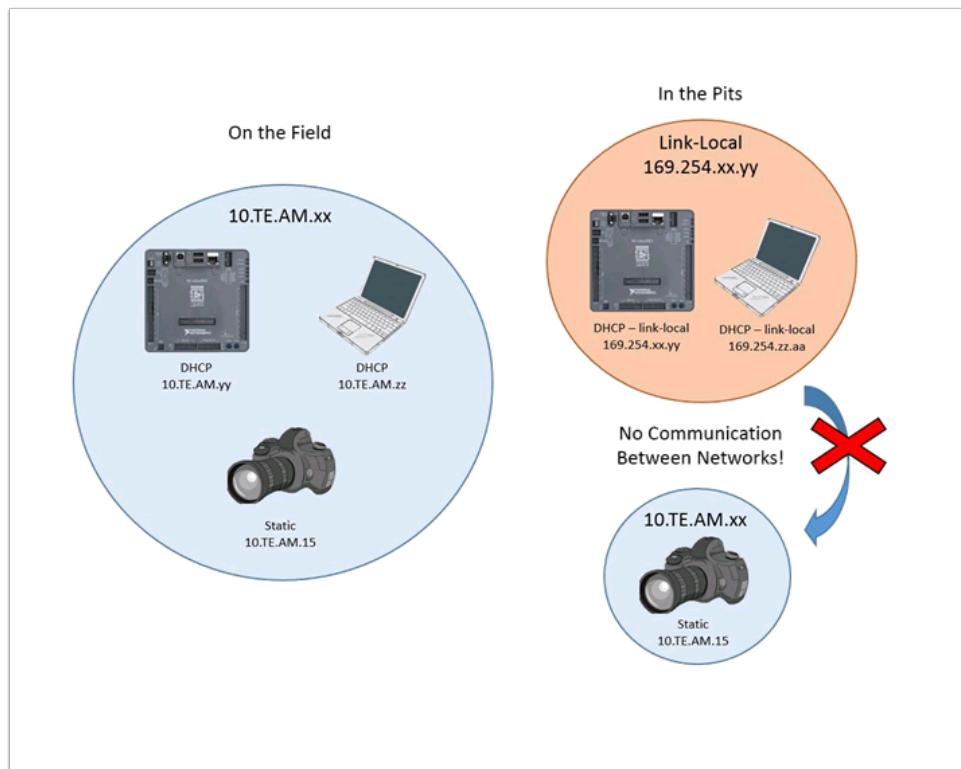
In a static addressing configuration, the configuration should be the same in the pits as it is on the field. Any programming computers will need to have an IP Address set in the 10.TE.AM.xx range with a subnet of 255.255.255.0

### Mixing Dynamic and Static Configurations

# Getting Started with the Control System

While on the field, the team should not notice any issues with having devices set statically in the 10.TE.AM.xx range, and having the field assign DHCP addresses as long as there are no IP address conflicts as referred to in the section above.

In the pits, a team may encounter issues with mixing Static and DHCP devices for the following reason. As mentioned above, DHCP devices will fall back to a link-local address (169.254.xx.yy) if a server isn't present. For static devices, the IP address will always be the same. If the DHCP server is not present and the roboRIO, driver station, and laptop fall back to link-local addresses, the statically set devices in the 10.TE.AM.xx range will be in a different network and not visible to those with link-local addresses. A visual description of this is provided below:



## mDNS

mDNS, or multicast Domain Name System is a protocol that allows us to benefit from the features of DNS, without having a DNS server on the network. To make this clearer, let's take a step back and talk about what DNS is.

### What is DNS?

# Getting Started with the Control System

DNS (Domain Name System) can become a complex topic, but for the scope of this paper, we are going to just look at the high level overview of DNS. In the most basic explanation, DNS is what allows us to relate human-friendly names for network devices to IP Addresses, and keep track of those IP addresses if they change.

**Example 1:** Let's look at the site [www.google.com](http://www.google.com). The IP address for this site is 173.227.93.118, however that is not very user friendly to remember!

Whenever a user types [www.google.com](http://www.google.com) into their computer, the computer contacts the DNS server (a setting provided by DHCP!) and asks what is the IP address on file for [www.google.com](http://www.google.com). The DNS server returns the IP address and then the computer is able to use that to connect to the Google web site.

**Example 2:** On your home network, you have a server named "MYCOMPUTER" that you want to connect to from your laptop. Your network uses DHCP so you don't know the IP Address of MYCOMPUTER, but DNS allows you to connect just by using the MYCOMPUTER name. Additionally, whenever the DHCP assignments refresh, MYCOMPUTER may end up with a different address, but because you're connecting by using the MYCOMPUTER name instead of a specific IP address, the DNS record was updated and you're still able to connect.

This is the second benefit to DNS, and the most relevant for FRC. With DNS, if we reference devices by their friendly name instead of IP Address, we don't have to change anything in our program if the IP Address changes. DNS will keep track of the changes and return the new address if it ever changes.

## DNS for FRC

On the field and in the pits, there is no DNS server that allows us to perform the lookups like we do for the Google web site, but we'd still like to have the benefits of not remembering every IP Address, and not having to guess at every device's address if DHCP assigns a different address than we expect. This is where mDNS comes into the picture.

mDNS provides us the same benefits as traditional DNS, but is just implemented in a way that does not require a server. Whenever a user asks to connect to a device using a friendly name, mDNS sends out a message asking the device with that name to identify itself. The device with the name then sends a return message including its IP address so all devices on the network can update their information. mDNS is what allows us to refer to our roboRIO as roboRIO-TEAM-FRC.local and have it connect on a DHCP network.

**Note:** If a device used for FRC does not support mDNS, then it will be assigned an IP Address in the 10.TE.AM.20 - 10.TE.AM.255 range, but we won't know the exact address to connect and we won't

# Getting Started with the Control System

be able to use the friendly name like before. In this case, the device would need to have a static IP Address.

## Summary

IP Addresses are what allow us to communicate with devices on a network. For FRC, these addresses are going to be in the 10.TE.AM.xx range if we are connected to a DHCP server or if they are assigned statically, or in the link-local 169.254.xx.yy range if the devices are set to DHCP, but there is no server present.

If all devices on the network support mDNS, then all devices can be set to DHCP and referred to using their friendly names (ex. roboRIO-TEAM-FRC.local). If some devices do not support mDNS, they will need to be set to use static addresses.

If all devices are set to use DHCP or Static IP assignments (with correct static settings), the communication should work in both the pit and on the field without any changes needed. If there are a mix of some Static and some DHCP devices, then the Static devices will connect on the field, but will not connect in the pit. This can be resolved by either setting all devices to static settings, or leaving the current settings and providing a DHCP server in the pit as referenced on page 6.

# Getting Started with the Control System

## 2018 Game Data Details

In the 2018 game, *FIRST® POWER UP<sup>SM</sup>*, the assignment of plates to alliances is randomized at the start of the match. To aid teams in programming autonomous routines, the Field Management System (FMS) will provide information to each team on the location of the plates assigned to their alliance. This page details the structure of that data and describes how to use it in each of the software languages.

## The Data

### Timing

Plate assignment data is provided to each robot (via the Driver Station) by the FMS just prior to the start of each match. The data will be sent after the "3...2...1...Go", and a short animation of the plate lighting will illustrate the randomization to ensure robots have time to receive and react to the data. To ensure your program properly responds to the data, you should either poll the data in a fairly fast loop while Disabled (~20ms) or retrieve the data once on the transition to Autonomous Enabled.

### Initial States

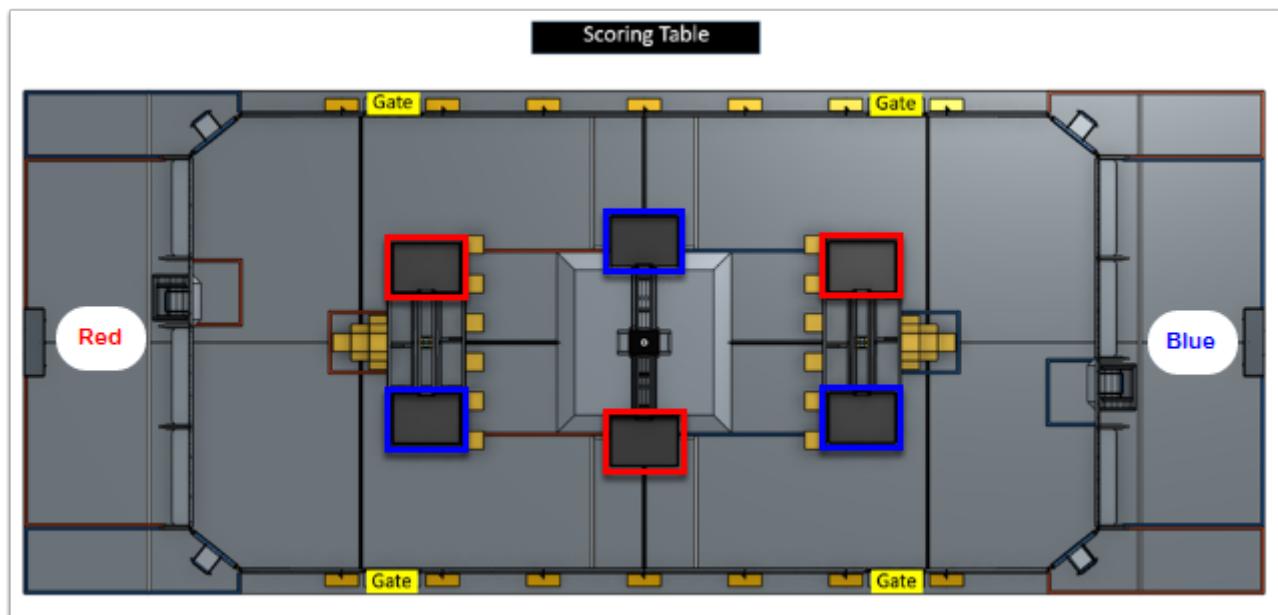
The initial state of the Game Data returned from the Robot code is non-deterministic. Before the data for the current match is sent the data returned will depend on what data is currently stored in the DS. This may be data from the last match, data from testing using the text box in the DS, or empty. If the data is empty, an empty (not NULL) string is returned. It is recommended to read the data as described above (and check string length if polling in Disabled) to avoid acting on incorrect data.

# Getting Started with the Control System

## The Data

Data regarding plate assignment is provided to each robot based on their alliance. In other words, the Blue alliance will receive data corresponding to the location of the Blue plates and the Red alliance will receive data corresponding to the location of the Red plates. The data is referenced from the perspective of the Drive Team looking out from their Player Station. The data consists of three characters, each 'L' or 'R', representing the location (Left or Right) of the Alliance's plate on each element, starting with the element closest to the Alliance.

## Example

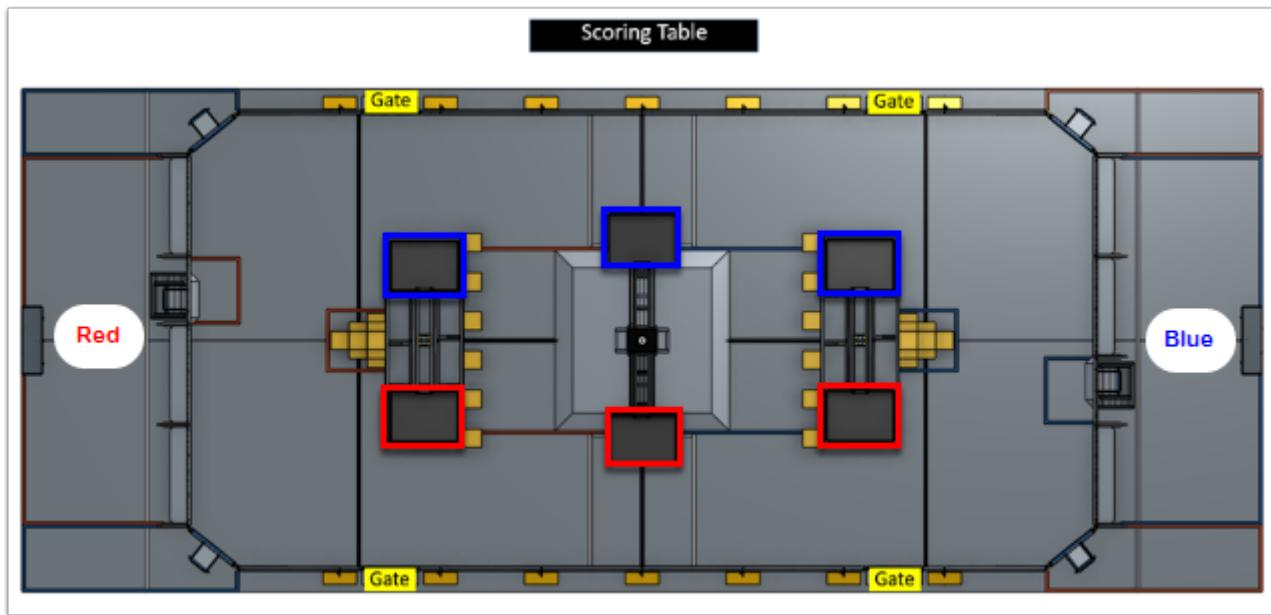


Red Alliance: "LRL"

Blue Alliance: "LRL"

# Getting Started with the Control System

## Example 2



Red Alliance: "RRR"

Blue Alliance: "RRR"

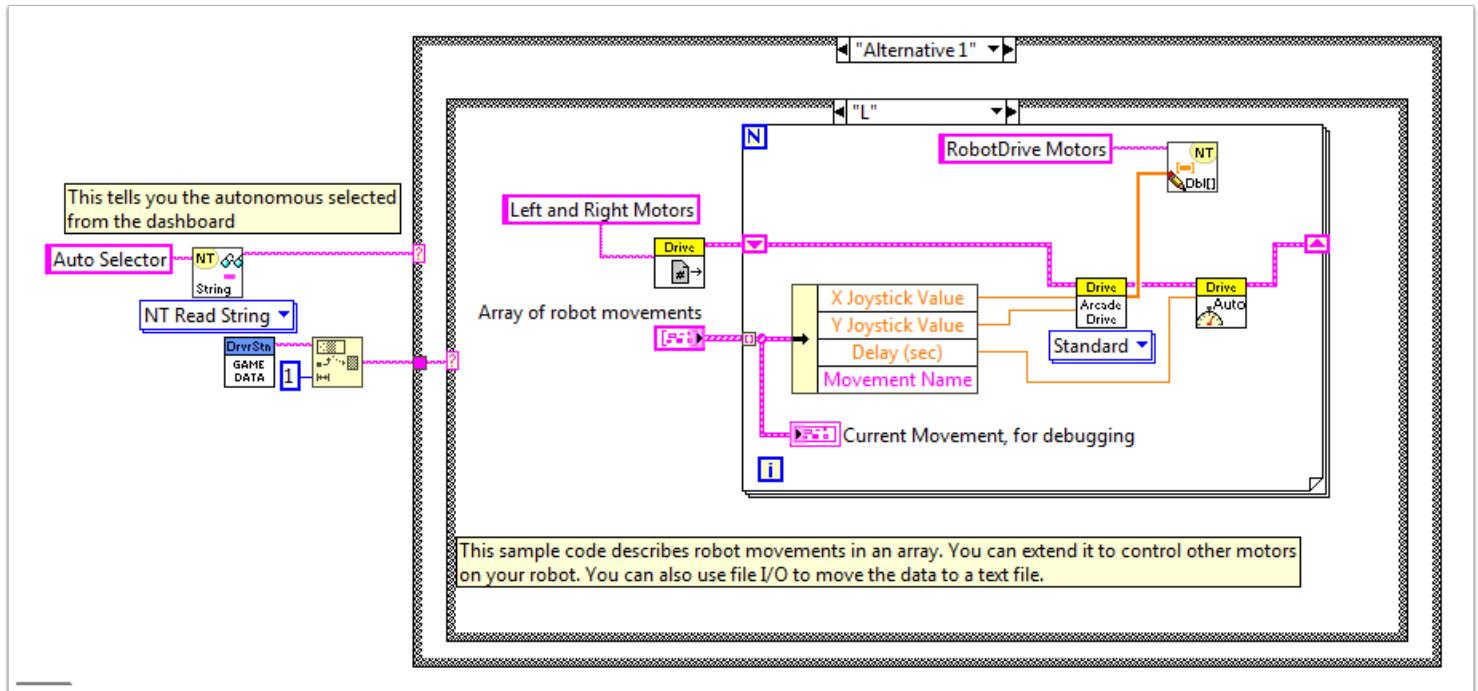
## LabVIEW

The Game Data in LabVIEW is accessed from the Game Specific Data VI. This VI can be found in the WPI Robotics Library -> Driver Station palette.

The most obvious location to query the Game Data in LabVIEW is at the beginning of the Autonomous Independent VI. You can use this information either in addition to, or in place of the "Auto Selector" code that is already present in the default template.

In the example below, the String Subset VI (from the String palette) is used to grab just the first character (the direction of the Switch). This information is then used with a Case Structure (make sure to rename the cases appropriately) inside the existing Auto Selector case structure. This construct allows for Left and Right versions of multiple auto routines (e.g. Switch, Scale, Drive, etc.). The example shows the "L" case of the inner structure.

# Getting Started with the Control System



## C++

The Game Data in C++ is accessed using the `GetGameSpecificMessage` method in the `DriverStation` class. If you are using the Iterative or Timed Robot classes (or Command Based, which is based on one of those) you can query the Game Data in the `AutonomousInit()` method. If you are using `SimpleRobot`, you can query it at the beginning of your `Autonomous()` method.

The example below grabs the first character (the direction of the Switch) and uses an if statement with a comparison to the 'L' character to determine what code to run. Similar concepts can be extrapolated to using a switch statement instead, or combining this comparison with other means of selecting auto modes such as a Sendable Chooser.

```
std::string gameData;
gameData = frc::DriverStation::GetInstance().GetGameSpecificMessage();
if(gameData.length > 0)
{
    if(gameData[0] == 'L')
    {
        //Put left auto code here
    } else {
```

# Getting Started with the Control System

```
//Put right auto code here  
}  
}  
}
```

## Java

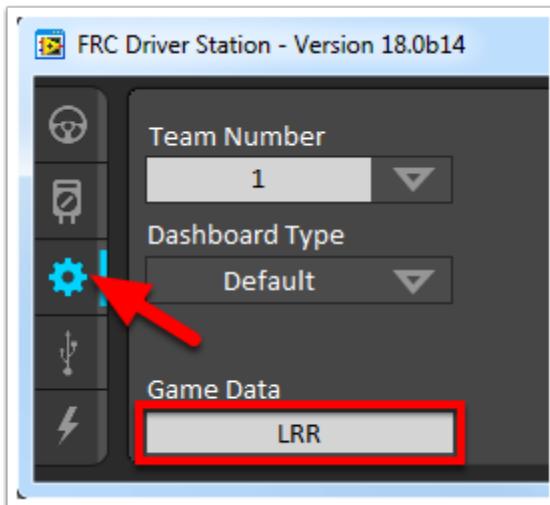
The Game Data in Java is accessed using the `getGameSpecificMessage` method in the `DriverStation` class. If you are using the Iterative or Timed Robot classes (or Command Based, which is based on one of those) you can query the Game Data in the `AutonomousInit()` method. If you are using `SimpleRobot`, you can query it at the beginning of your `Autonomous()` method.

The example below grabs the first character (the direction of the Switch) and uses an if statement with a comparison to the 'L' character to determine what code to run. Similar concepts can be extrapolated to using a switch statement instead, or combining this comparison with other means of selecting auto modes such as a Sendable Chooser.

```
String gameData;  
gameData = DriverStation.getInstance().getGameSpecificMessage();  
if(gameData.length() > 0)  
{  
    if(gameData.charAt(0) == 'L')  
    {  
        //Put left auto code here  
    } else {  
        //Put right auto code here  
    }  
}
```

# Getting Started with the Control System

## Testing Game Specific Data



You can test your Game Specific Data code without FMS by using the Driver Station. Click on the Setup tab of the Driver Station, then enter the desired test string into the Game Data text field. Wait a few seconds to make sure the data has been transmitted to the robot, then Enable the robot in Autonomous mode.