

GETTING STARTED WITH THE CONTROL SYSTEM

Getting Started with the Control System

Table of Contents

- Getting Started with the ScreenSteps Documentation4
 - Getting Started With the Screen Steps Documentation5
- System Overview.....7
 - New for 2018!.....8
 - FRC Control System Hardware Overview..... 14
 - FRC Software Component Overview 35
 - Offline Installation Preparation..... 52
 - 3rd Party Libraries 54
- Getting Started: Step-By-Step..... 57
 - Wiring the FRC Control System 58
 - Rookie USB stick..... 81
 - Imaging your Classmate (Veteran Image Download)..... 82
 - Installing LabVIEW for FRC 2018 (LabVIEW only) 104
 - Installing Eclipse (C++/Java) 127
 - Installing the FRC Update Suite (All Languages)..... 150
 - Imaging your roboRIO 171
 - FRC Driver Station Powered by NI LabVIEW 177
 - Running your Benchtop Test Program - Tethered..... 188
 - Programming your radio 190
 - Running your Benchtop Test Program - Wireless..... 205
 - Updating and Configuring Pneumatics Control Module and Power Distribution Panel..... 207

Getting Started with the Control System

- Getting Started 222
- System Details 224
 - RoboRIO Networking..... 225
 - Networking Basics 229
 - 2018 Game Data Details 239

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.

System Overview

Getting Started with the Control System

New for 2018!

CAN Webdash Plugin

- ! The CAN Webdashboard plugin is no longer included in the base roboRIO image. To enable CAN device functionality in the roboRIO WebDashboard you must now install the plugin using CTRE's Phoenix Lifeboat. Instructions can be found on the [CTRE Phoenix Wiki](#).

Third Party Libraries

- CTRE has made extensive changes to the TalonSRX libraries. Documentation on the new Phoenix Framework can be found on the [CTRE Github wiki](#). A migration guide can be found on the wiki [here](#).

Java 9

- ! Warning: Java 9 is not currently supported by the FRC tools. Java 9 introduces many breaking changes, and is not provided for 32-bit systems which we need to support. Java 9 will not be supported for 2018.

Deprecated features

If you use WPILib with a text-based language (C++ or Java) you might see some deprecated functions or classes. This means that the function or class still exists and is supported but it is not the best choice and you should look for some newer, more robust replacement. The replacement function to use will often be mentioned in the message or in the documentation below.

Deprecated functions and classes will work this year but maybe removed or changed in a future year.

Getting Started with the Control System

C++/Java WPILib

- Added new base class, TimedRobot. TimedRobot is similar to IterativeRobot but loops on a timer instead of syncing to the arrival of DS packets. This provides for more consistent timing of robot actions and controls, at the expense of potentially processing some inputs for multiple loops, or skipping some inputs (depending on loop time and jitter).
- RobotDrive has been split into separate classes for each drive base type. SpeedControllerGroup is used to specify multiple motor controllers per side.
 - MecanumDrive
 - KilloughDrive
 - DifferentialDrive (tank drive, WCD, etc.)
 - Now also has CurvatureDrive() (aka cheesy drive).
 - Uses of Drive() member function for differential drive base should be replaced with CurvatureDrive()
- TalonSRX renamed to PWMTalonSRX
 - CTRE's new software also renames CANTalon to WPI_TalonSRX
- A periodic method has been added inside subsystems
- Joystick/XboxController interface changes
 - Added functions to Joystick and XboxController for detecting button press and release events
- SetAxisChannel() and GetAxisChannel() have been replaced with setters and getters for each axis channel.
- GetAxis() and GetButton() have been replaced with getters for each axis/button.
- Added interface for DS match data
- Added support for Nidec Brushless motor (NidecBrushless class)
- Changed Command templates to use TimedRobot instead of IterativeRobot
- Changed Java I2C and SPI to use byte arrays instead of ByteBuffer to avoid memory leak

Java

- New JRE (zulu) from Azul Systems. This allows us to package the JRE with the Eclipse plugins and deploy it automatically, removing the need for the manual Java Installer tool.
- Sendable Chooser now keeps entries ordered
- OpenCV Mat's can now be created before calling CameraServer functions

Getting Started with the Control System

C++

- Restored use of priority mutexes in HAL and C++ (available as `wpi::mutex` and `wpi::condition_variable` in C++)

LabVIEW

- Increase robustness of USB Camera connections and streaming
- New simulation tools. The FRC roboRIO Project wizard has new "Learn LabVIEW" simulation choices which are a maze and a ball shooter. These include fully functional simulation robots. Documentation for these is not yet complete.
- Updated to LabVIEW 2017

FRC Driver Station powered by NI LabVIEW

- Improved connectivity diagnostics (firewall info, etc.)
- Removed protocol selector (to connect to cRIO systems, use the FRC 2014 DS)

Network Tables

- Improved Connection Synchronization Behavior
 - This should reduce or eliminate the occurrences of "phantom keys" on the Dashboard when changes are made to code to rename or remove keys without restarting the Dashboard
 - The original synchronization behavior was troublesome for two reasons:
 1. It had unpredictable behavior for updated values
 2. It brought back to life deleted values
 - Instead of relying on the server to inform the client regarding reconnections, the client keeps track of what values have been modified by user code on the client. When the client connects to the server, the following occurs:
 - For entries that have been modified by user code on the client:
 - If the entry is not persistent, the server value is overwritten with the client value
 - If the entry does not exist on the server, the client sends an assignment to the server to recreate it on the server

Getting Started with the Control System

- For entries that have not been modified by user code on the client:
 - The client value is overwritten with the server value
 - If the entry does not exist on the server, the client deletes the entry
- JSON/CBOR/MessagePack library is now included in C++ for easier mapping of structured data into raw NetworkTables entries. This is a lightweight version of <https://github.com/nlohmann/json>, imported into the wpi namespace.
- API updated to provide handle-based access for improved performance. This also allows multiple simultaneous independent instances to be created (e.g. for unit testing purposes).
 - Added new classes and interfaces (NetworkTableInstance, NetworkTableEntry, NetworkTableValue, and others). In Java, these, along with a new version of the NetworkTable class, are now located in a different package (edu.wpi.first.networktables, rather than edu.wpi.first.wpilibj.networktables)
 - The NetworkTable class is still available, but many functions are now deprecated in favor of NetworkTableInstance and NetworkTableEntry functions.
 - In Java, many JNI functions changed; it's recommended users of the old JNI functions change to using NetworkTableInstance instead of JNI directly.
 - There is a "default" instance that is configured by the WPILib robot code during startup; this can be retrieved using NetworkTableInstance.getDefault(). You can then get NetworkTable's by calling NetworkTableInstance.getTable(String). This replaces the old NetworkTable.getTable() static function.
 - NetworkTableEntry's are returned by NetworkTableEntry(String). NetworkTableEntry provides functions for getting and setting values similar to the old NetworkTable string-based getters and setters, but is faster because it uses handles rather than string lookups.
- In Java, callbacks are now executed by a Java thread rather than a native thread.

SmartDashboard

- Brand new [Shuffleboard](#) dashboard added.
- SFX dashboard removed (will be provided as a separate download, but is no longer being actively maintained)

LiveWindow

LiveWindow now provides continuous telemetry (e.g. of motor and sensor values) for most WPILib classes via NetworkTables. Telemetry is sent each loop iteration of the IterativeRobot and

Getting Started with the Control System

TimedRobot templates (SampleRobot does not provide this functionality). Dashboards such as Shuffleboard provide ways to record this telemetry for later playback and analysis.

Most WPILib classes add themselves to the LiveWindow when they are constructed. While we recommend telemetry be left enabled, telemetry for specific instances can be disabled using `LiveWindow.disableTelemetry()`, or all telemetry can be disabled using `LiveWindow.disableAllTelemetry()`.

To implement this change, the Sendable interface now uses a property definition interface rather than multiple functions. The `LiveWindowSendable` and `NamedSendable` classes have been deprecated.

Subsystem now provides an `addChild()` function to use instead of `LiveWindow.addSensor()` and `LiveWindow.addActuator()`, which have been deprecated.

OutlineViewer

- Rewritten in JavaFX
- Enter team number along with server name
- Support for non-standard port numbers
- Checkboxes for boolean values

Eclipse Plugins

- Much faster deploy
 - Checks multiple network addresses in parallel
 - Uses WebDAV instead of SSH for transferring files
- Improved roboRio image version detection
- Automatically deploys Java JRE if not already installed
 - Java installer is no longer necessary and has been removed from the plugins
- Implement TCP netconsole (riolog) and show it automatically after successful deploy. The TCP netconsole is colorized like the DS console and provides additional filtering features. Note: it only displays output when a DS is connected (from any computer, not necessarily the Eclipse computer).

Getting Started with the Control System

Radio

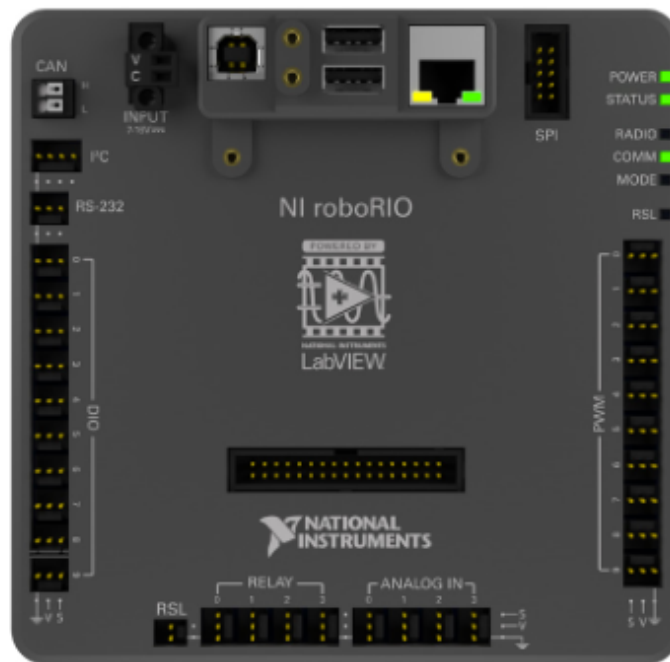
- Reserved DHCP Address added for roboRIO (10.TE.AM.2)
- DHCP enabled on Wired side even in bridge mode (and at events)
- 2nd Ethernet port issue on OM5P-AC believed to be resolved

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

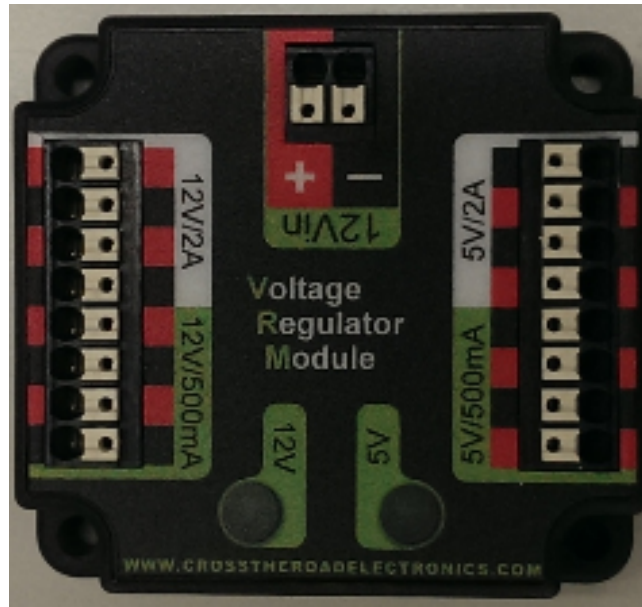
Pneumatics Control Module



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 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>

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/product/spark/>

Getting Started with the Control System

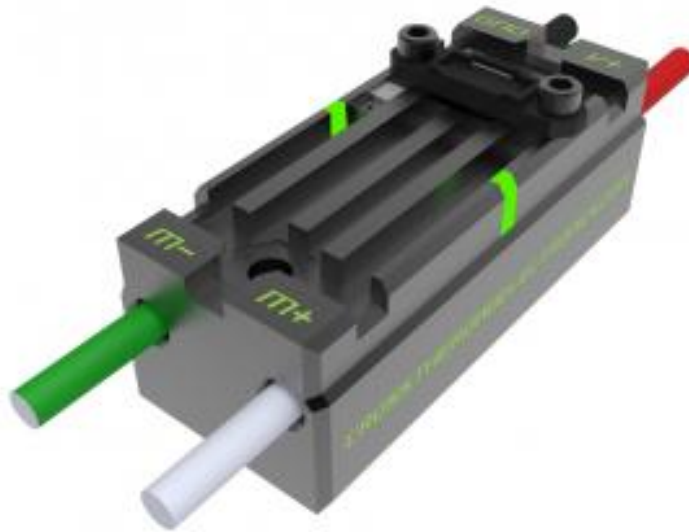
Talon Motor Controller



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

Getting Started with the Control System

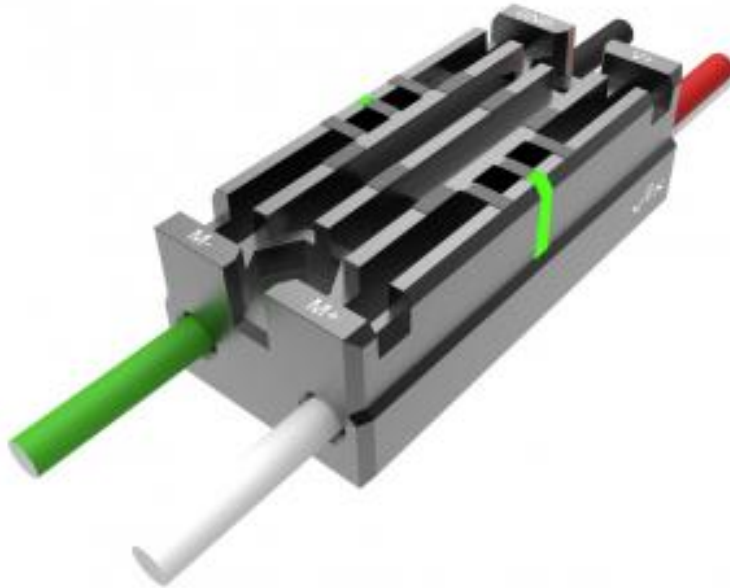
Victor 888 Motor Controller / Victor 884 Motor Controller



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

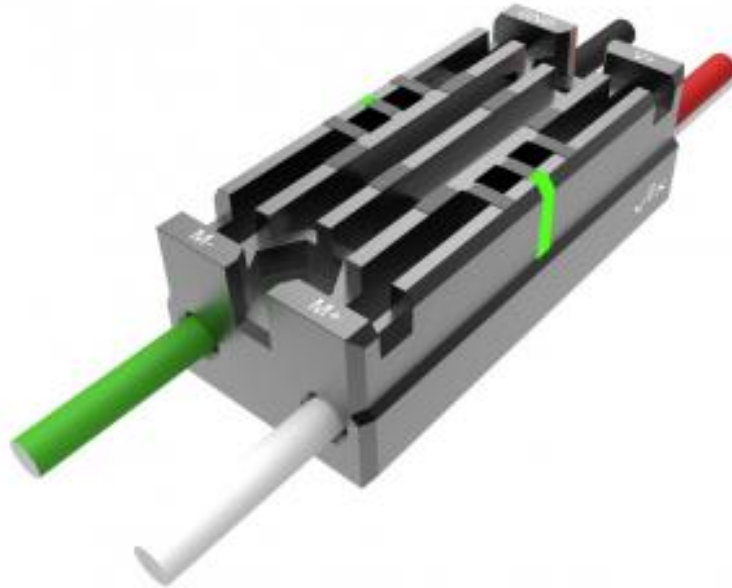
Victor SP



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

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](#) of this documentation.

Getting Started with the Control System

OpenMesh OM5P-AN or OM5P-AC Radio



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 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.

Getting Started with the Control System

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

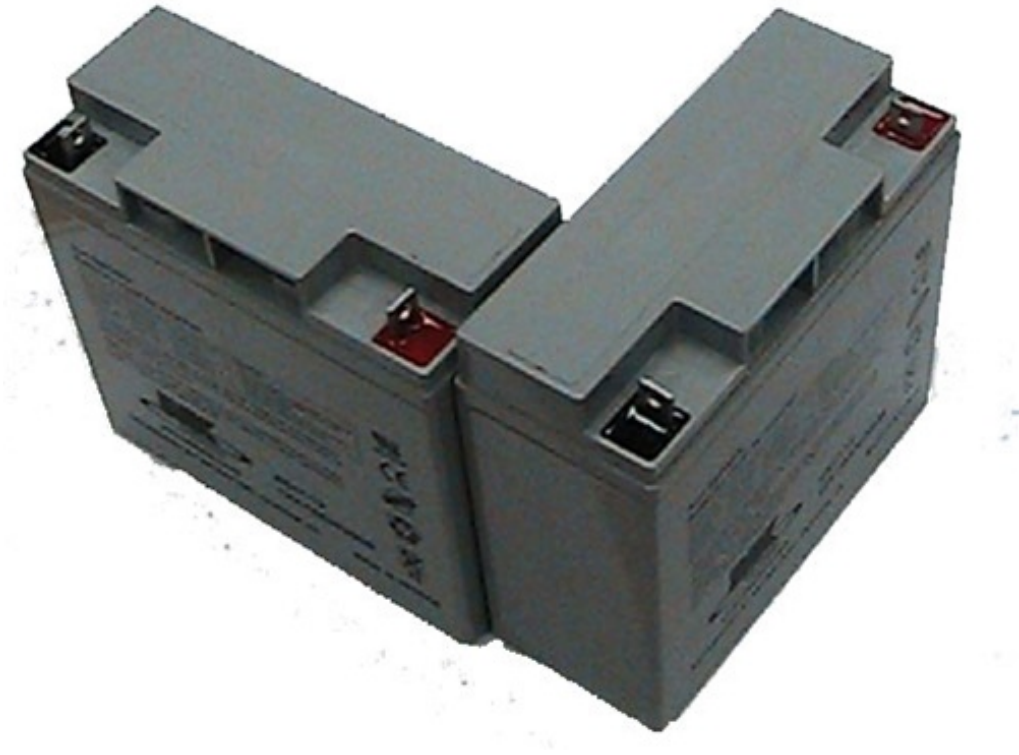
Snap Action Circuit Breakers



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 Datasheet 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 [Energys 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 [Diligent](#). 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.](#) 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 designed to assist you in the design, development and debugging of your robot code, control robot operation, and provide feedback to assist with 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.

OS Compatibility

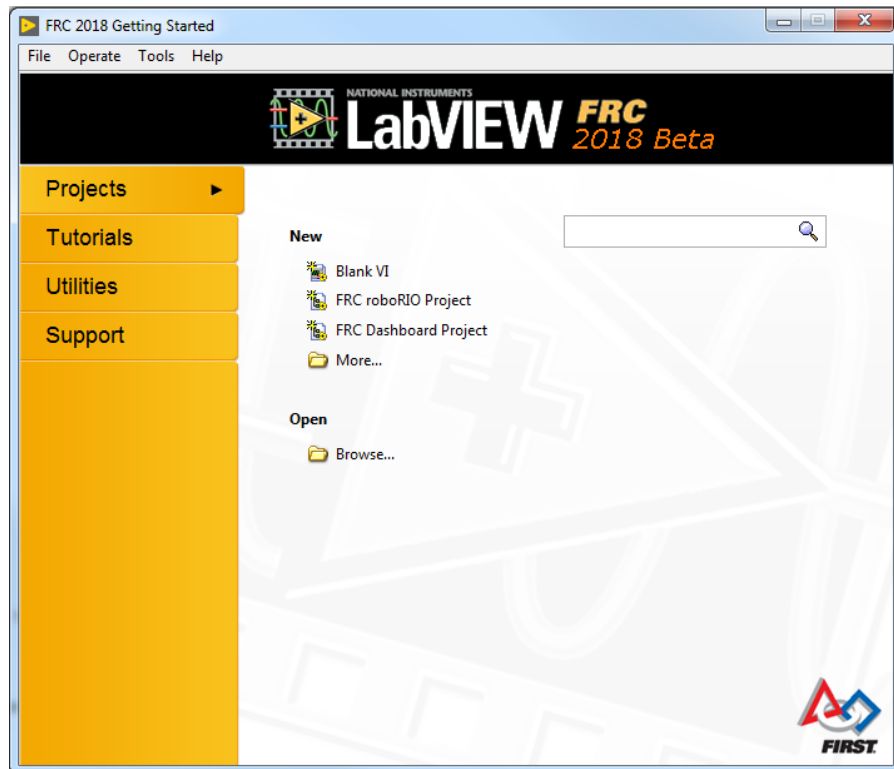
The primary supported OS for FRC components is Windows. All required FRC software components have been tested on Windows 7, 8 and 10. Windows XP is not supported. The only software that will not operate on Windows is the C++\Java FRCSim simulator, which requires specific versions of Linux.

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, 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

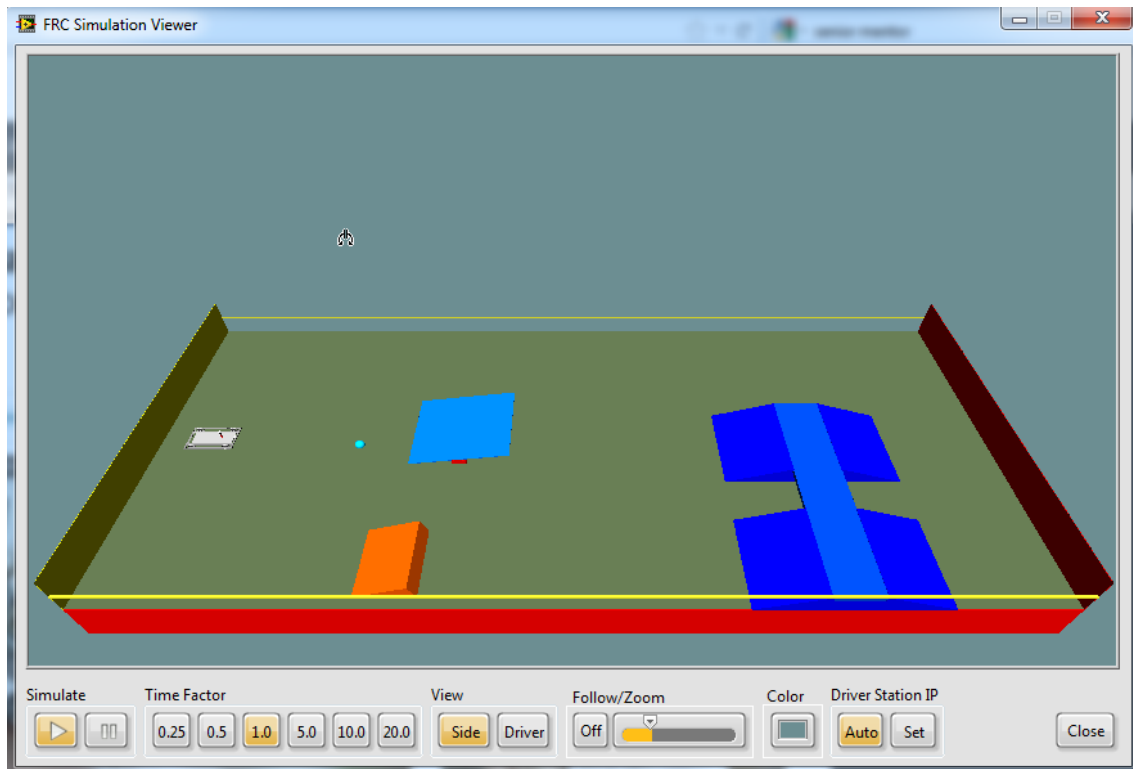
LabVIEW FRC



LabVIEW FRC, based on National Instruments' LabVIEW 2017, 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

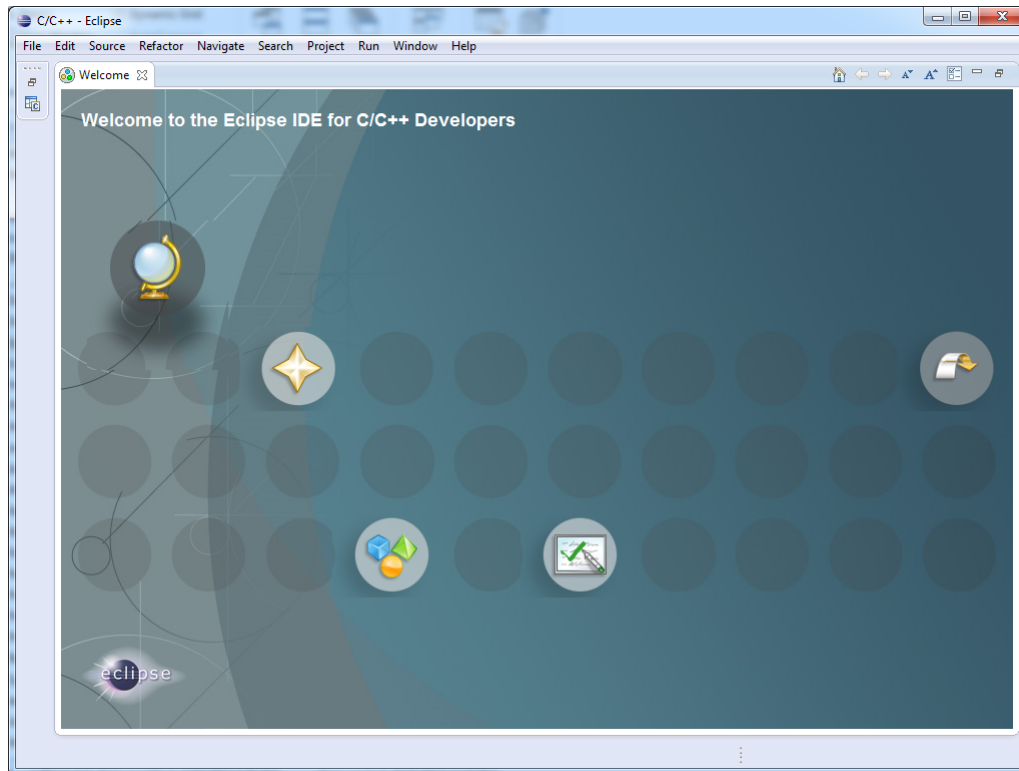
FRC Robot Simulator



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 LabVIEW FRC 2017. 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

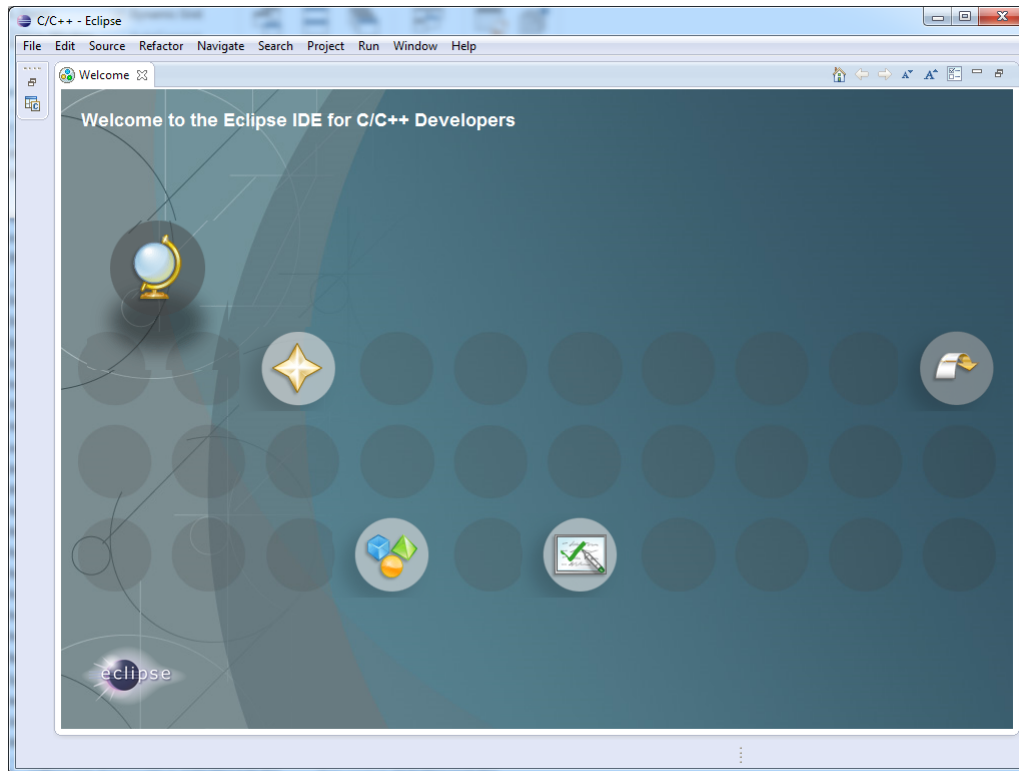
Eclipse IDE for C/C++ Developers*



Eclipse IDE for C/C++ Developers is the supported development environment for C++, one of the three supported languages used for programming an FRC robot. C++ is an object-oriented text based programming language. A program in C++ (for FRC) consists of a number of header (.h) and implementation (.cpp) files. It is recommended to install the language specific updates through Eclipse in order to be automatically notified of updates. A guide to getting started with C++ for FRC, including the installation and configuration of Eclipse IDE for C/C++ Developers can be found [here](#).

Getting Started with the Control System

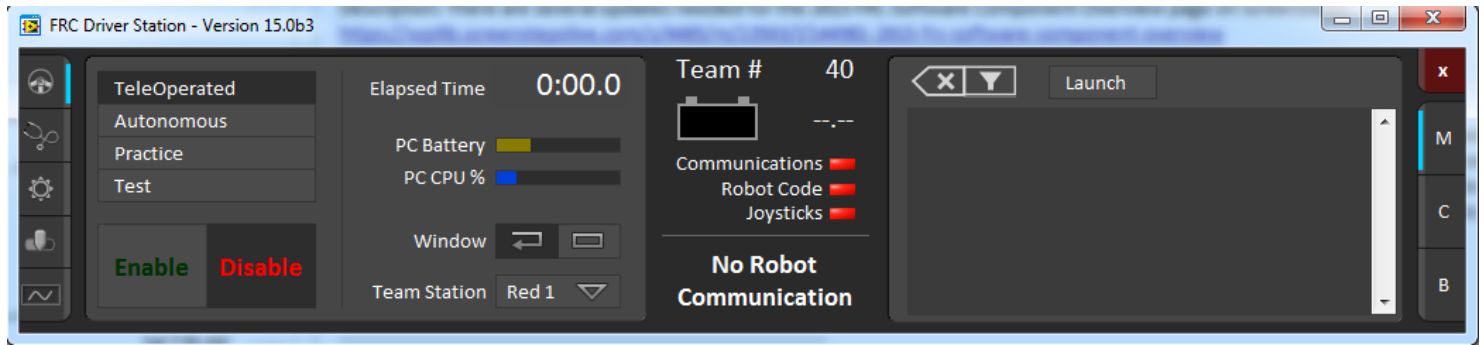
Eclipse IDE for Java Developers*



Eclipse IDE for Java Developers is the primary supported development environment for Java, one of the three supported languages used for programming an FRC robot. Java is an object-oriented text base programming language. A program in Java (for FRC) consists of one or more .java files contained in one or more packages. A guide to getting started with Java for FRC, including the installation and configuration of the Eclipse IDE can be found [here](#).

Getting Started with the Control System

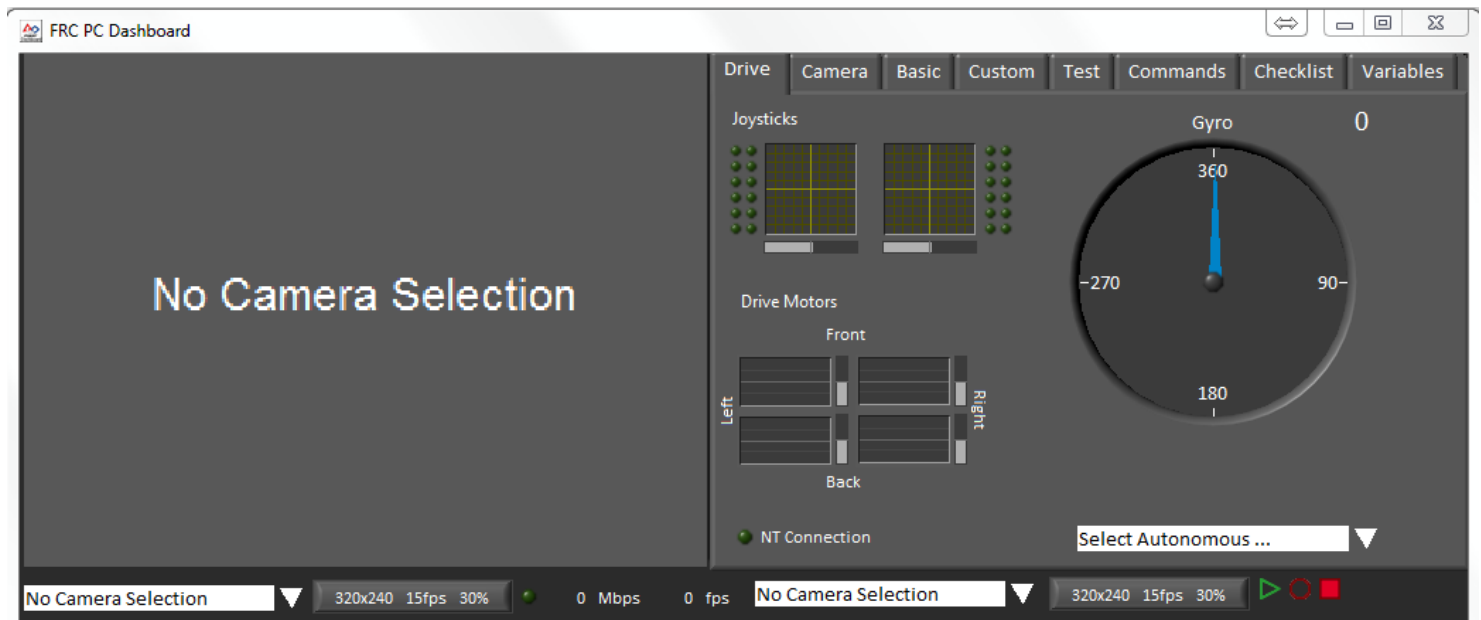
FRC Driver Station Powered by NI LabVIEW



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

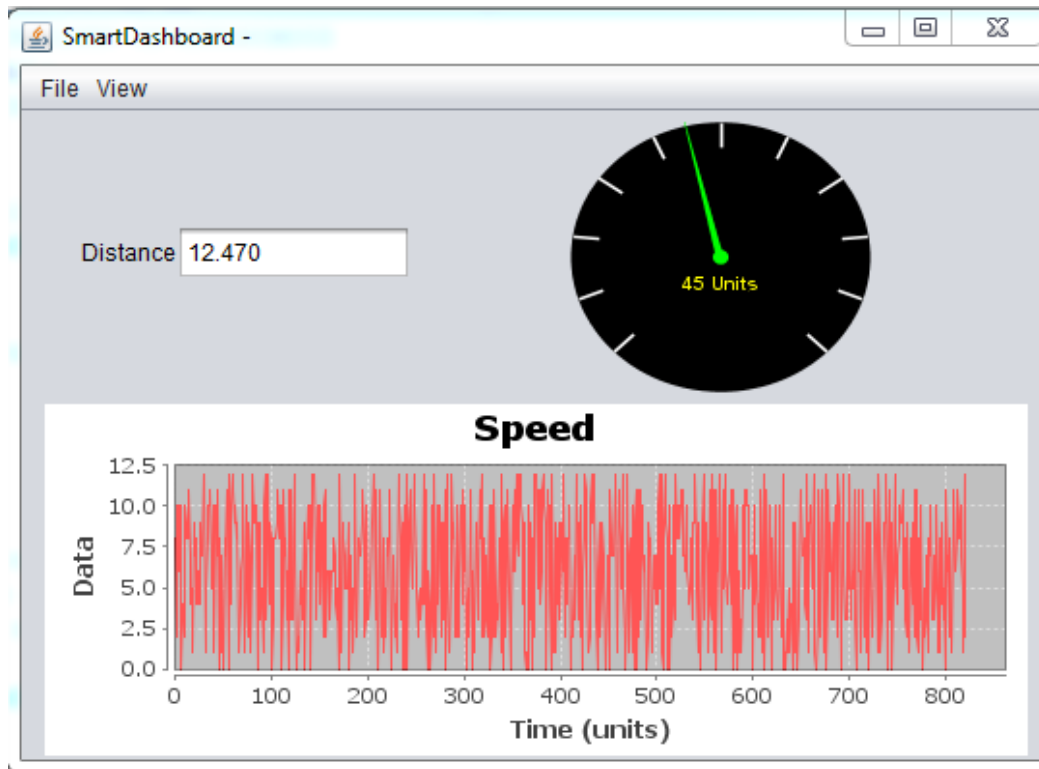
FRC LabVIEW Dashboard



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 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

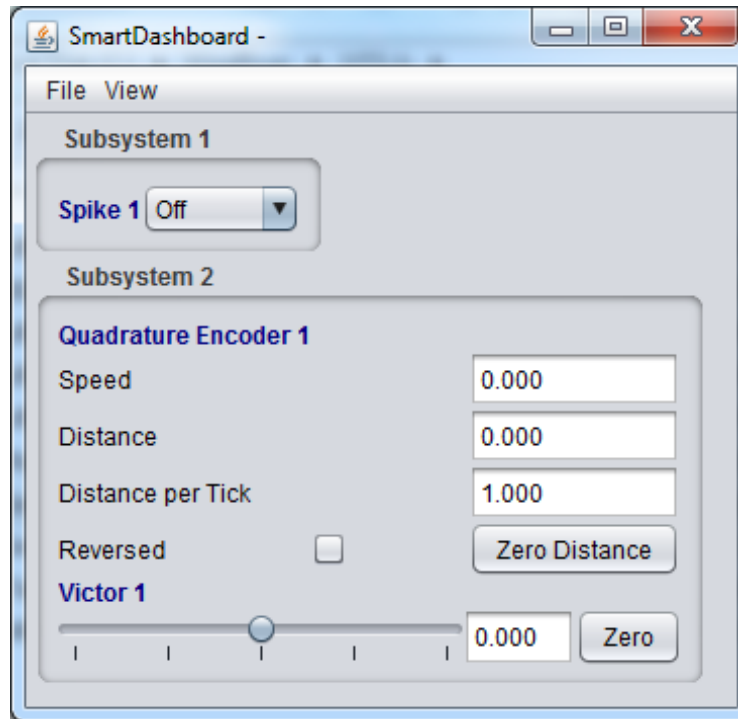
SmartDashboard*



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

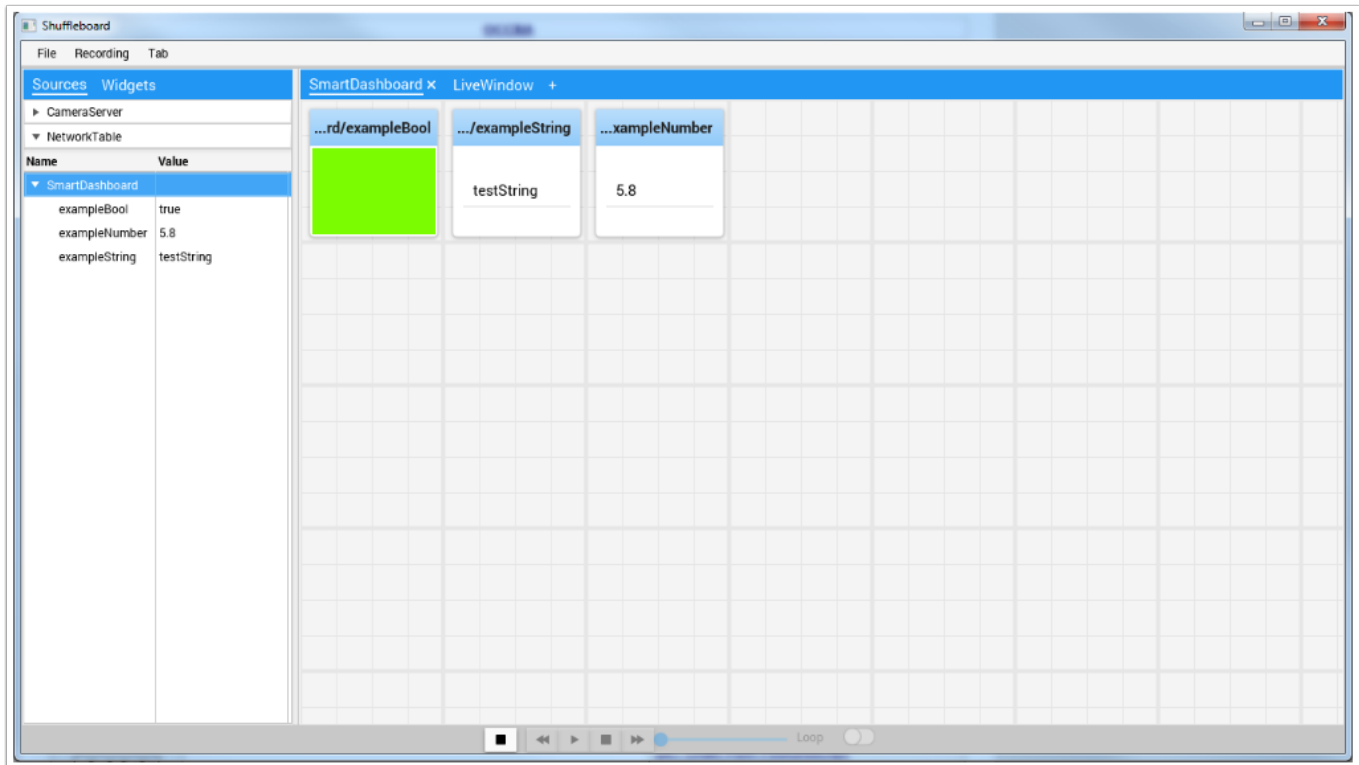
LiveWindow*



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

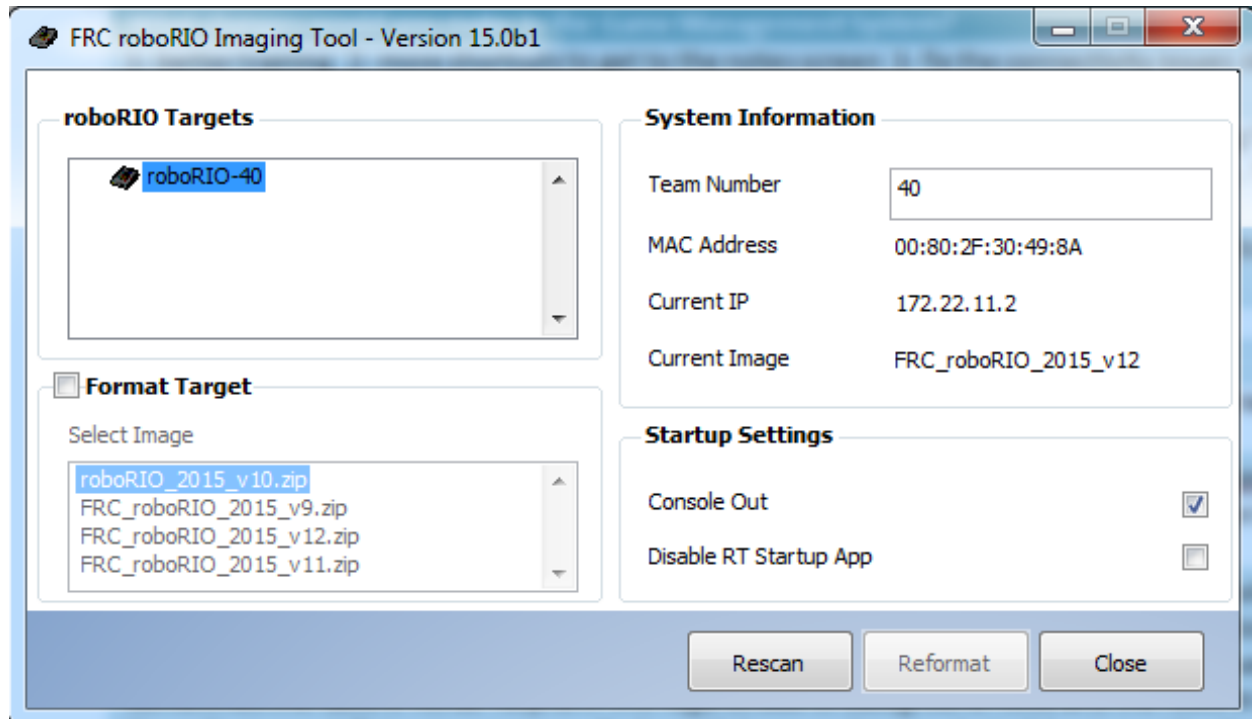
Shuffleboard*



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

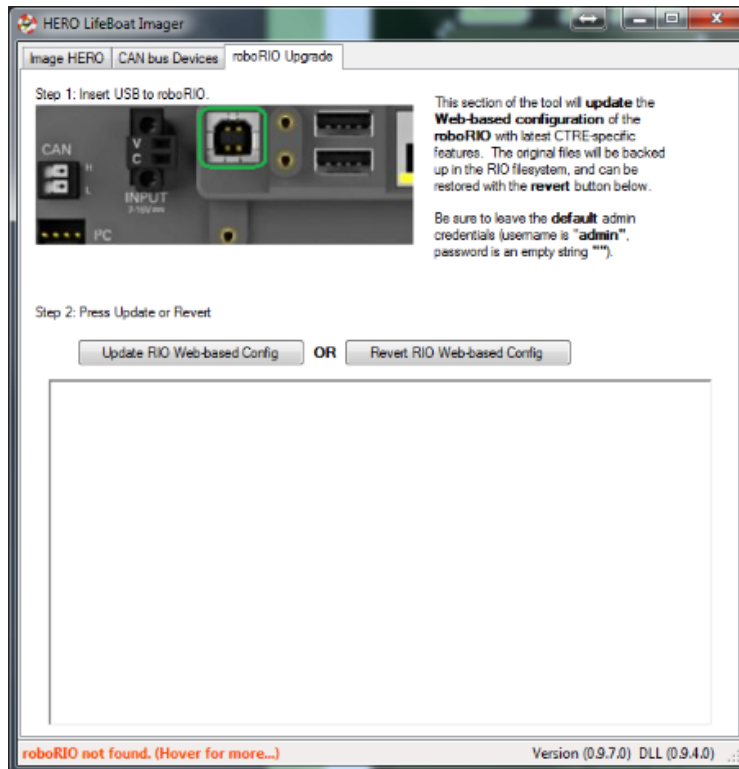
FRC roboRIO Imaging Tool



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*

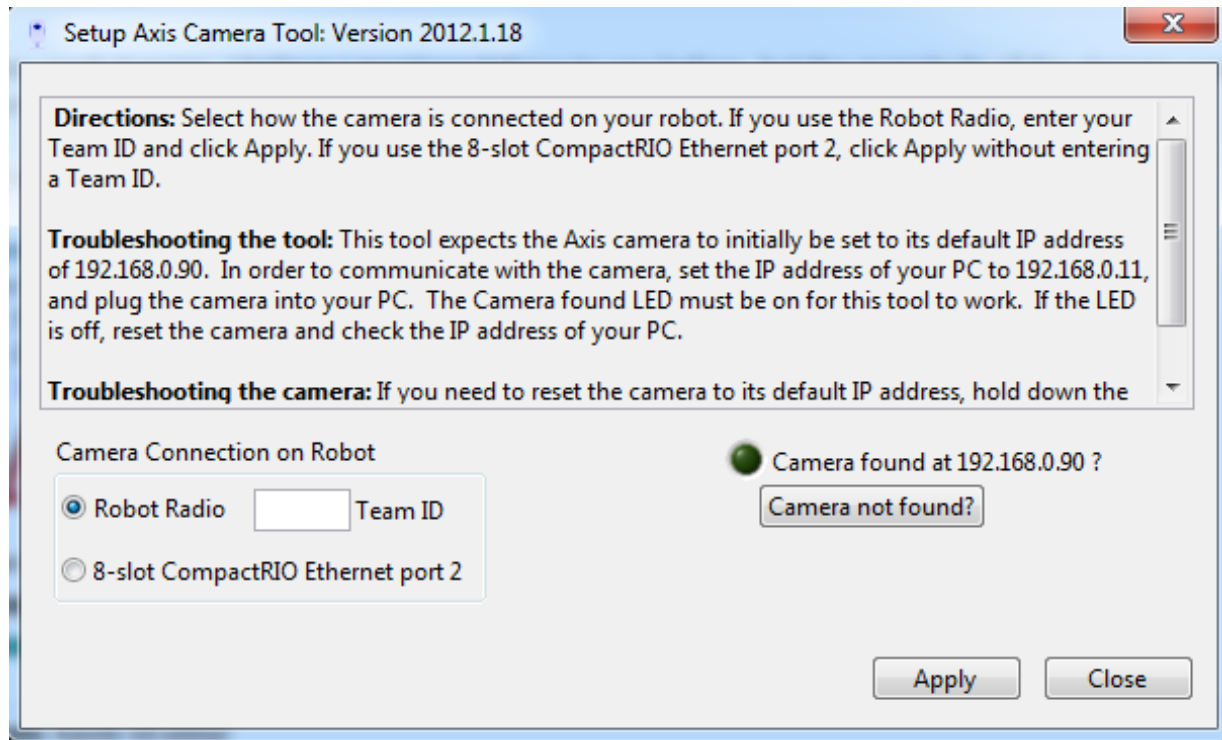


The CTRE Toolsuite installs the software libraries for Talon SRX (C++\Java\LabVIEW) as well as the HERO Lifeboat software which can be used to update the roboRIO web based CAN configuration with the latest CTRE-specific features. The installer can be found here: <http://www.ctr-electronics.com/control-system/hro.html>

Note on non-Windows: A separate package (zip) is provided to get the Talon SRX and Pidgeon 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

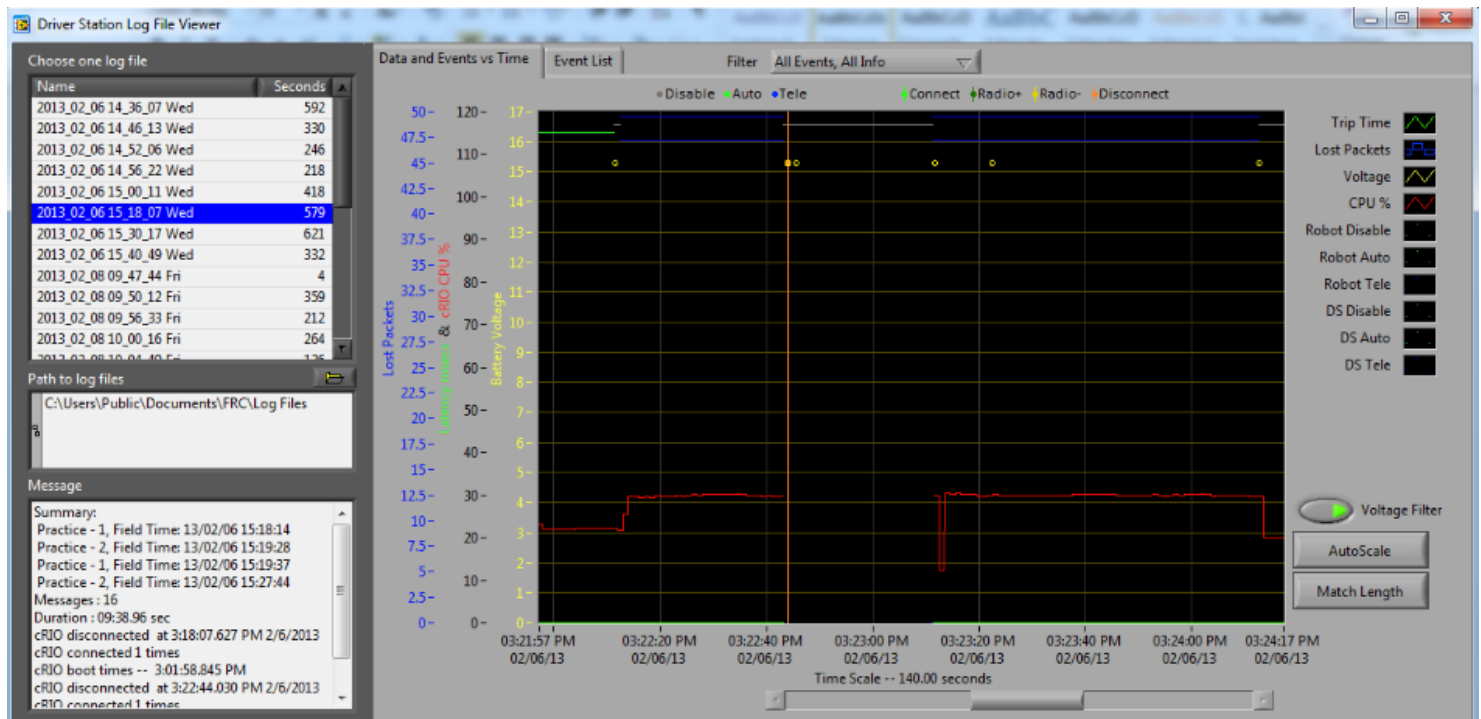
Setup Axis Camera



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

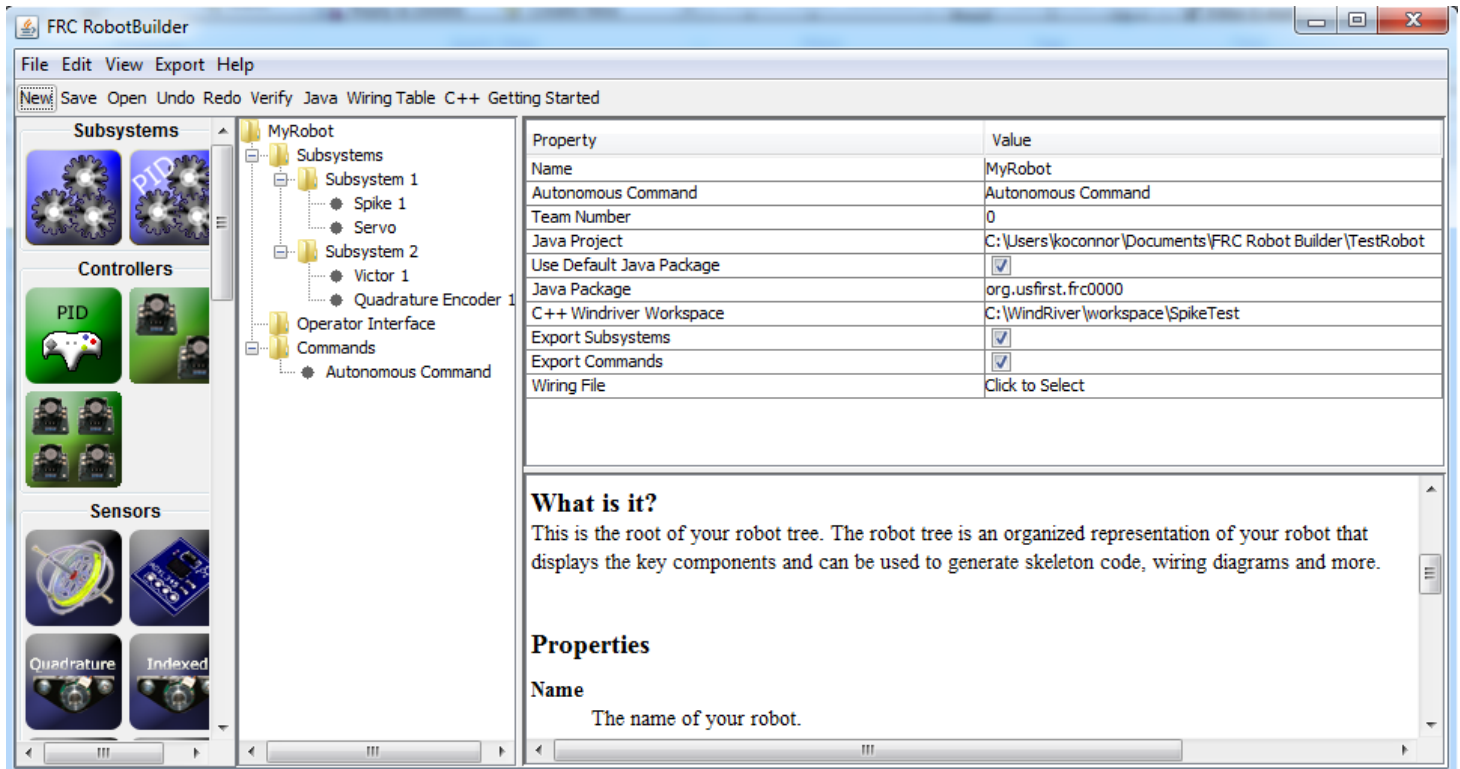
FRC Driver Station Log Viewer



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

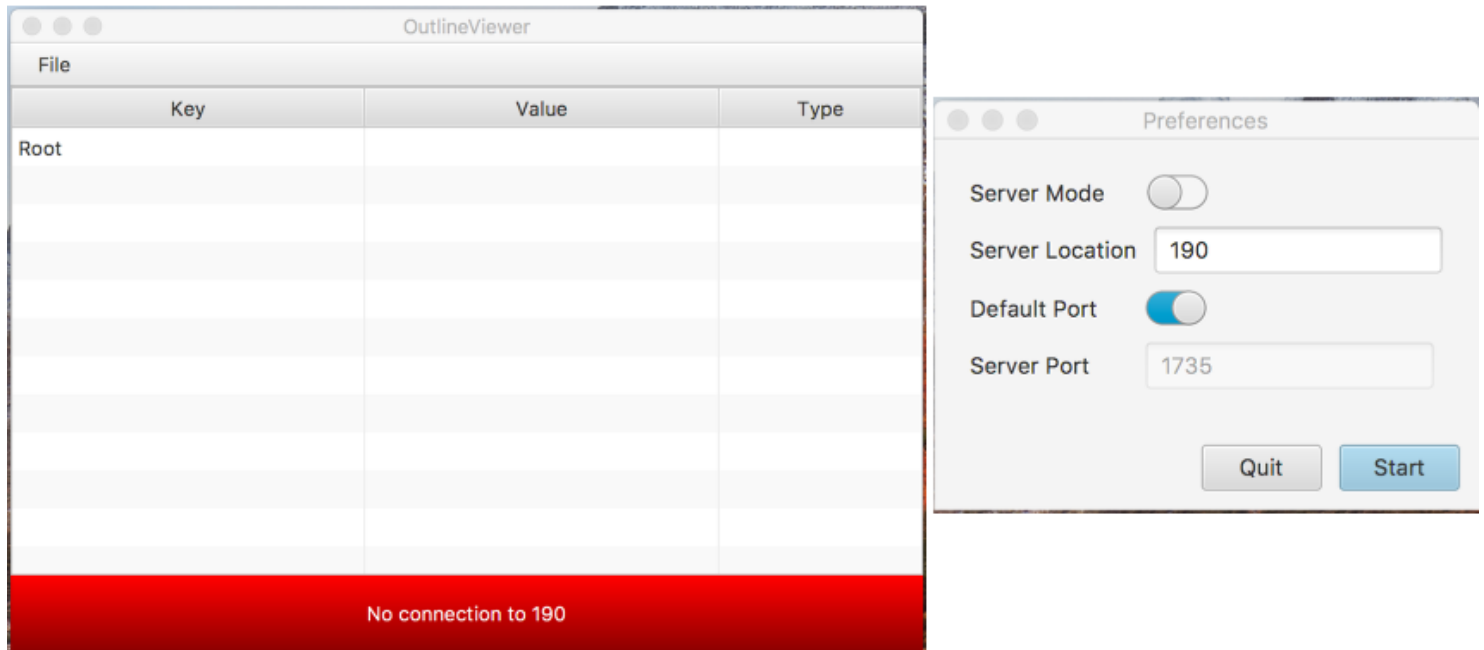
Robot Builder*



Robot Builder is a tool designed to aid in setup and structuring of a Command Based robot project for C++ or Java. Robot Builder 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. Robot Builder 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. Robot Builder is installed with the [C++ or Java language specific updates](#) (in the USER\wpilib\tools directory). Note that teams may need to install the Java Runtime Environment to use the Robot Builder on computers not set up for Java programming. 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*



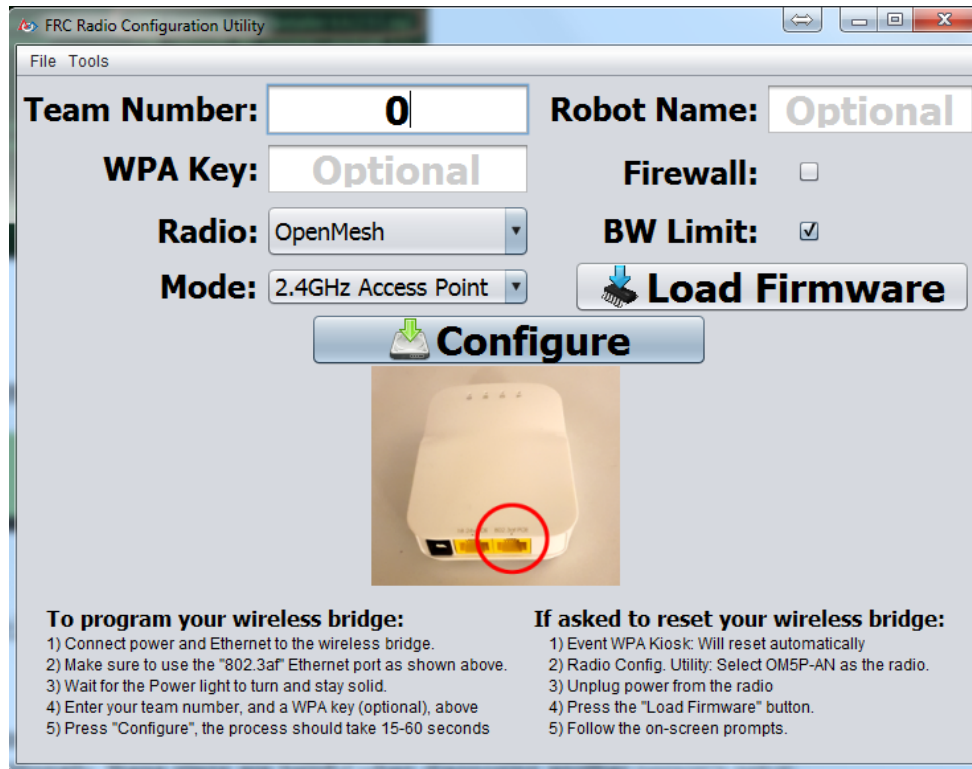
OutlineViewer is a utility used to view, modify and add to the contents of the NetworkTables for debugging purposes. It displays all key value pairs currently in the NetworkTables and can be used to modify the value of existing keys or add new keys to the table. OutlineViewer is included in the [C++ 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



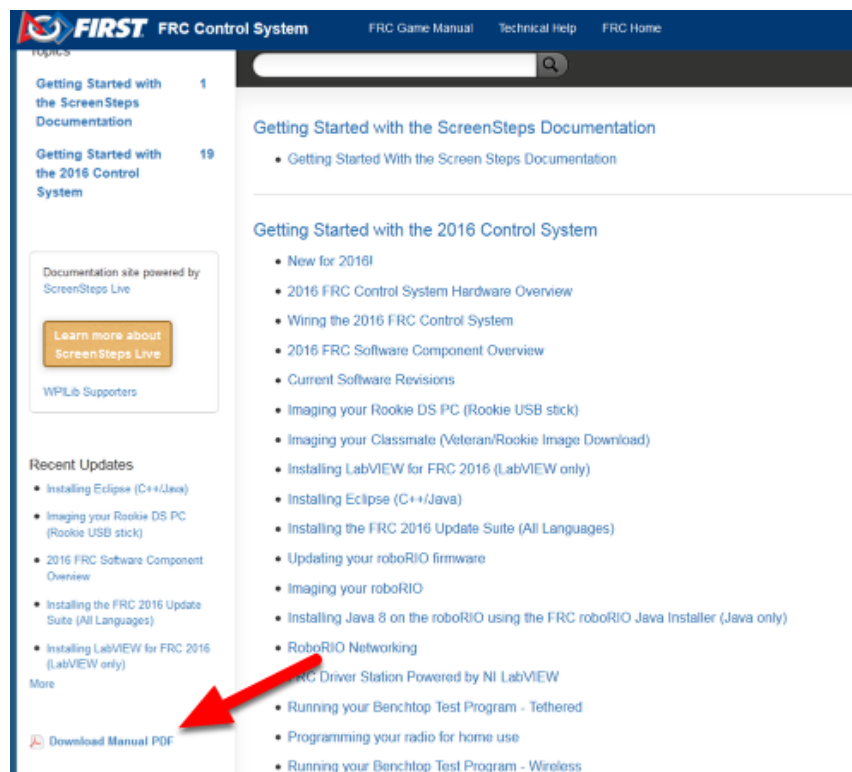
The FRC Bridge Configuration Utility is a tool used to configure the the OpenMesh OM5P-AN or OM5P-AC radio for practice use at home. This tool sets the appropriate IP, and network settings for proper network connection, as well as the QOS settings required to mimic the bandwidth limiting and packet prioritization 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

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 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:

- [2018 NI Update Suite](#) (Note: Requires decryption key from kickoff broadcast!)
- [2018 FRC Radio Configuration Utility](#) OR [2018 FRC Radio Configuration Utility Israel Version](#)
- (Optional - Veterans Only!) [Classmate/Acer PC Image](#)

LabVIEW Teams

- LabVIEW DVD or [Download](#) (Note: The DVD comes in your KOP)

C++ Teams

- [Java 8 SE Development Kit](#) (x86/x64 should match Eclipse)
- [Eclipse C++ version](#) (x86/x64 should match Java)
- [WPI Plugin Zip](#)
- [C++ Toolchain](#)

Java Teams

- [Java 8 SE Development Kit](#) (x86/x64 should match Eclipse)
- [Eclipse Java version](#) (x86/x64 should match Java)
- [WPI Plugin Zip](#)

3rd Party Libraries/Software

New for 2017 - A number of software components were broken out of WPILib for 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](#).

3rd Party Libraries

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

Libraries

[CTRE Toolsuite](#) - Contains TalonSRX Libraries and Lifeboat program for updating roboRIO webdashboard with latest CTRE features

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

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

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

C++ and Java teams should close Eclipse before running any 3rd party installer. If you do install with Eclipse open, make sure to restart Eclipse before trying to use the library.

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 mechanisms for C++ and Java are very similar but they have been split to describe exact usage and folders for each language.

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\WPI\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 VIs and they will be added automatically.

Getting Started with the Control System

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 VIs in one of these 3 locations and drag them into your project as you would with any other VI.

The Mechanism - C++

For C++ teams, third party libraries will be installed to a subdirectory of your WPILib install that is created by the Eclipse plugins. The plugins create a directory inside USER or USERHOME (C:\Users\USERNAME on Windows) called "wpilib". Third party libraries install inside this directory in "wpilib\user". Libraries may choose to put examples or documentation in this root location. C++ libraries install include files to \wpilib\user\cpp\include and and library files (.a or .so) to wpilib\user\cpp\lib. Files in the "include" folder are automatically added to the include path of all WPILib projects (including old projects that you open with new WPILib plugins) for you to include in your programs. Files in the "lib" folder are automatically added to the build command of all WPILib projects and any .so files are automatically copied to the roboRIO into /usr/local/frc/lib which allows the robot program to load them.

To use installed Third Party libraries, simply include the appropriate header in your C++ program. You can either locate the appropriate include in the documentation provided by the library provider or browse to USER\wpilib\user\cpp\include to see what third party includes are available.

Note that libraries may also choose to include C++ specific source, examples or documentation into either the root of \wpilib\user\cpp or into subdirectories

The Mechanism - Java

For Java teams, third party libraries will be installed to a subdirectory of your WPILib install that is created by the Eclipse plugins. The plugins create a directory inside USER or USERHOME (C:\Users\USERNAME on Windows) called "wpilib". Third party libraries install inside this directory in "wpilib\user". Libraries may choose to put examples or documentation in this root location. Java libraries install JAR files containing java code and any .so files containing native code used by the library to \wpilib\user\java\lib. JAR files in the "lib" folder are automatically added to the classpath of all WPILib projects (including old projects that you open with new WPILib plugins) for you to include in your programs and packaged into your robot program jar on deploy. Native library (.so) files in the "lib" folder are automatically copied to the roboRIO into /usr/local/frc/lib which allows the robot program to load them.

Getting Started with the Control System

To use installed Third Party libraries, simply import the appropriate package or class into your Java program.

Note that libraries may also choose to include Java specific source, examples or documentation into either the root of \wpilib\user\java or into subdirectories

Getting Started: Step-By-Step

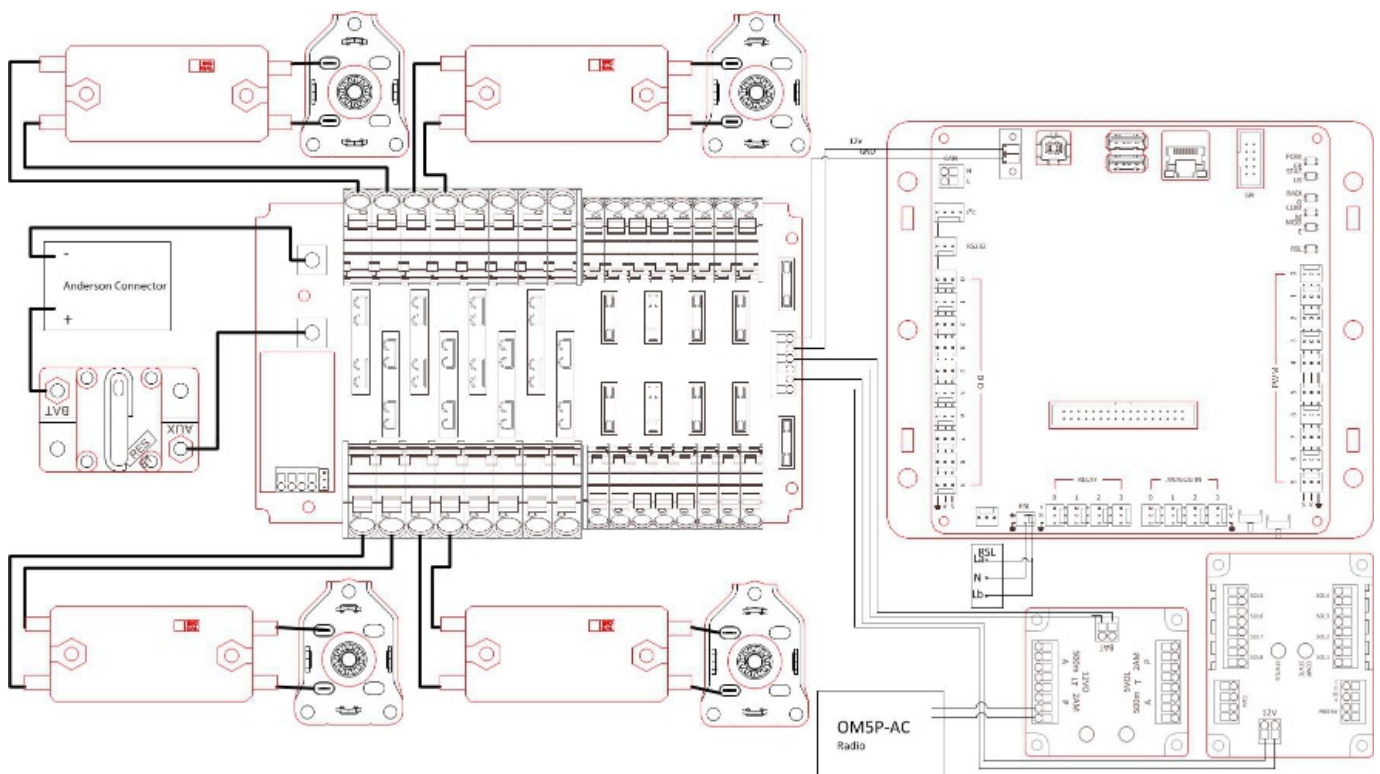
Getting Started with the Control System

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 Sparks 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

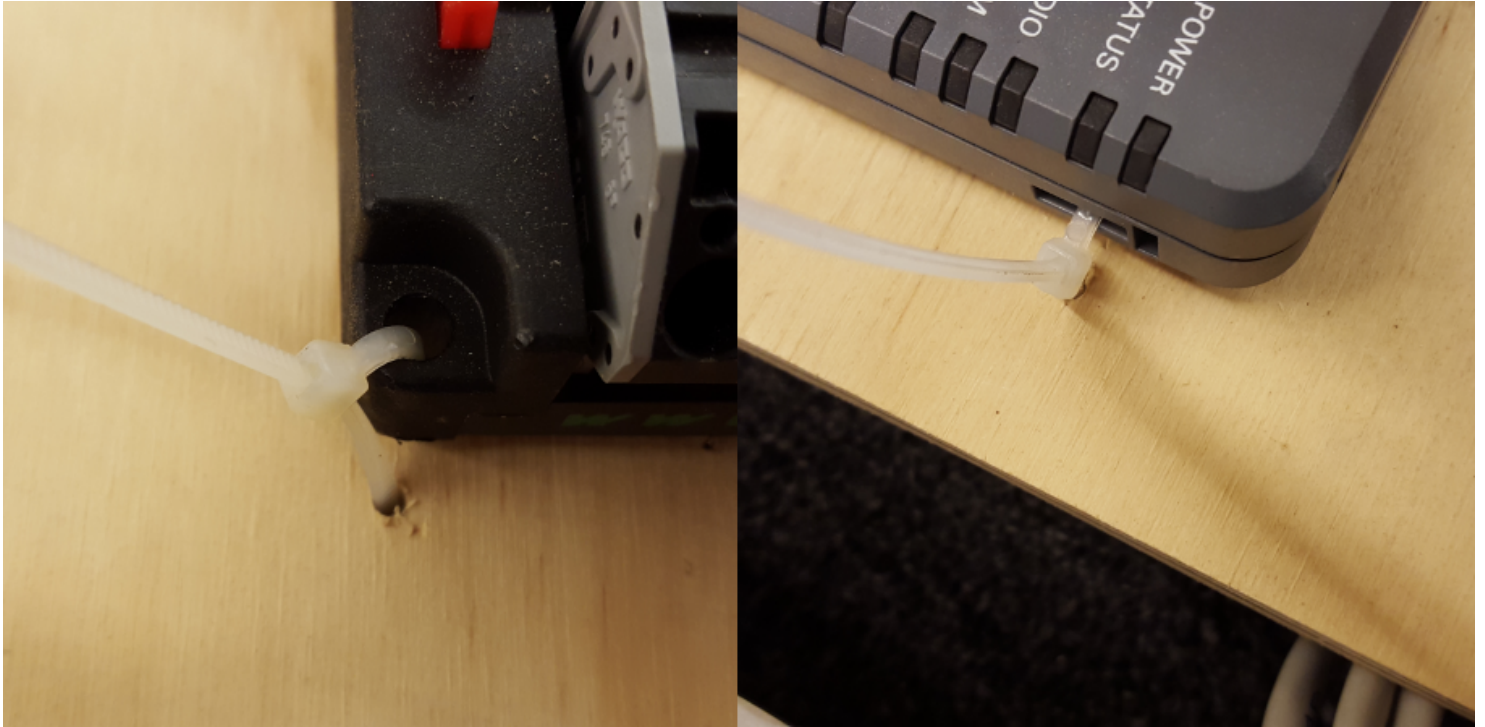


Layout the components on the board. One layout that should work is shown in the images above.

Note: If creating the board for a robot chassis, per the QuickBuild instructions for the **long orientation**, you may wish to turn the battery 90 degrees clockwise compared to the image above and spread the components on each side accordingly in order to accommodate building a box to retain the battery without hitting the CIM motors.

Getting Started with the Control System

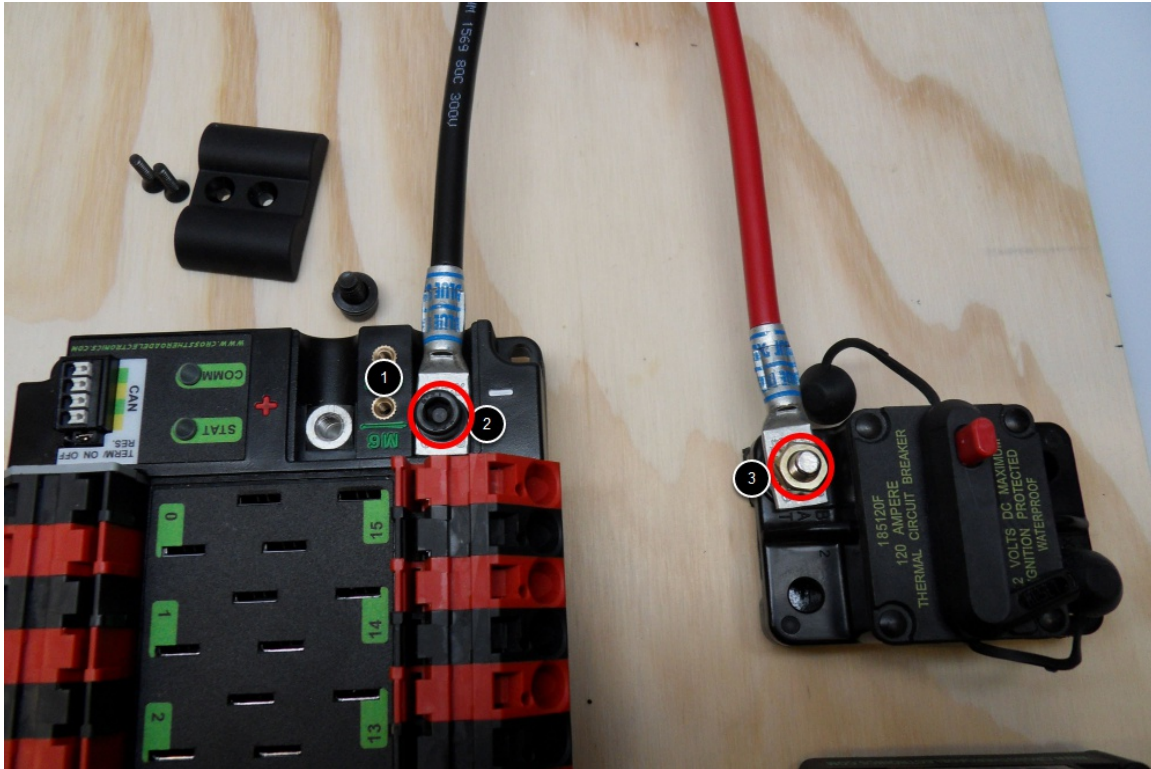
Fasten components



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

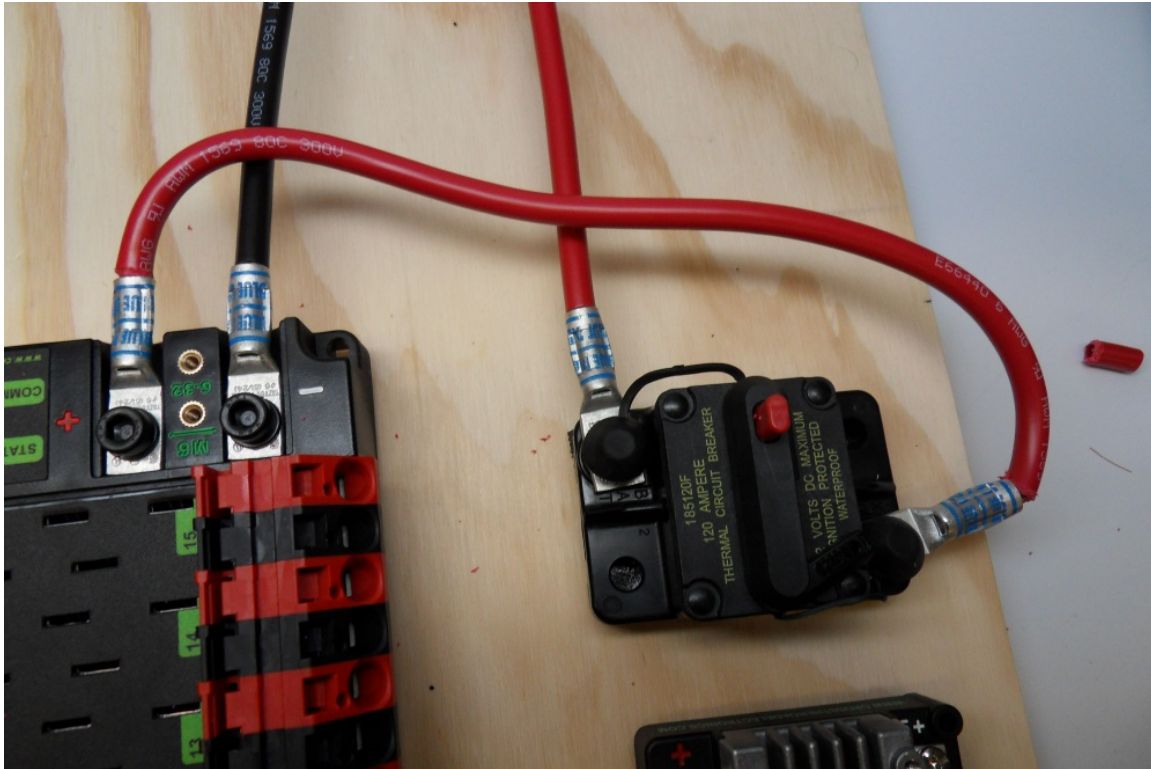


Requires: Battery Connector, 6AWG terminal lugs, 1/16" Allen, 5mm Allen, 7/16" Box end

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 connector

Getting Started with the Control System

Wire Breaker to PDP



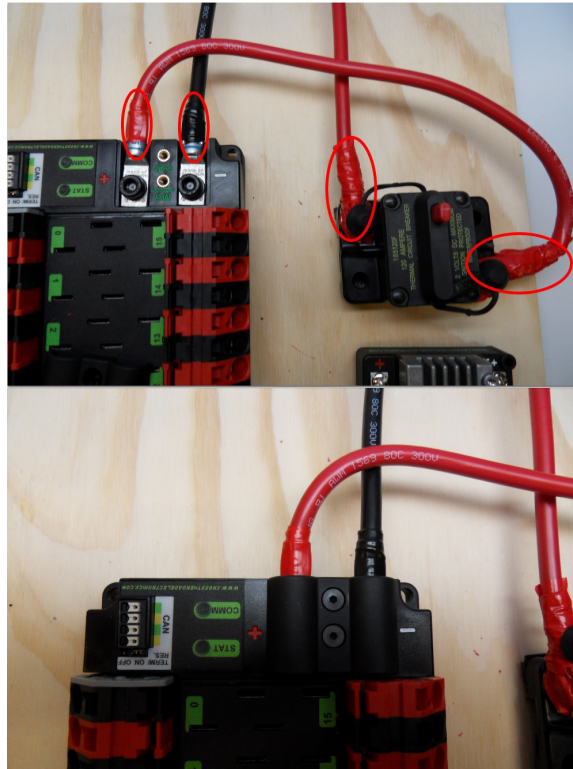
Requires: 6AWG red wire, 2x 6AWG terminal lugs, 5mm Allen, 7/16" box end

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

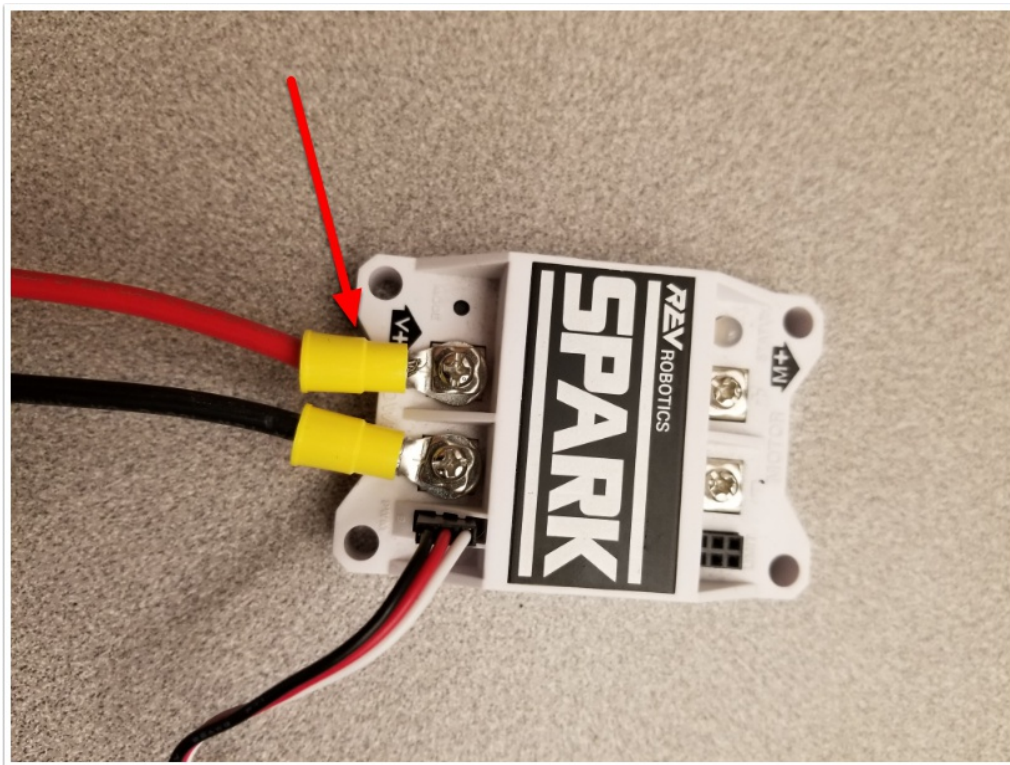
The next step will involve using the Wago connectors on the PDP. To use the Wago connectors, insert a small flat blade screwdriver into the rectangular hole at a shallow angle then angle the screwdriver upwards as you continue to press in to actuate the lever, opening the terminal. Two sizes of Wago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

Getting Started with the Control System

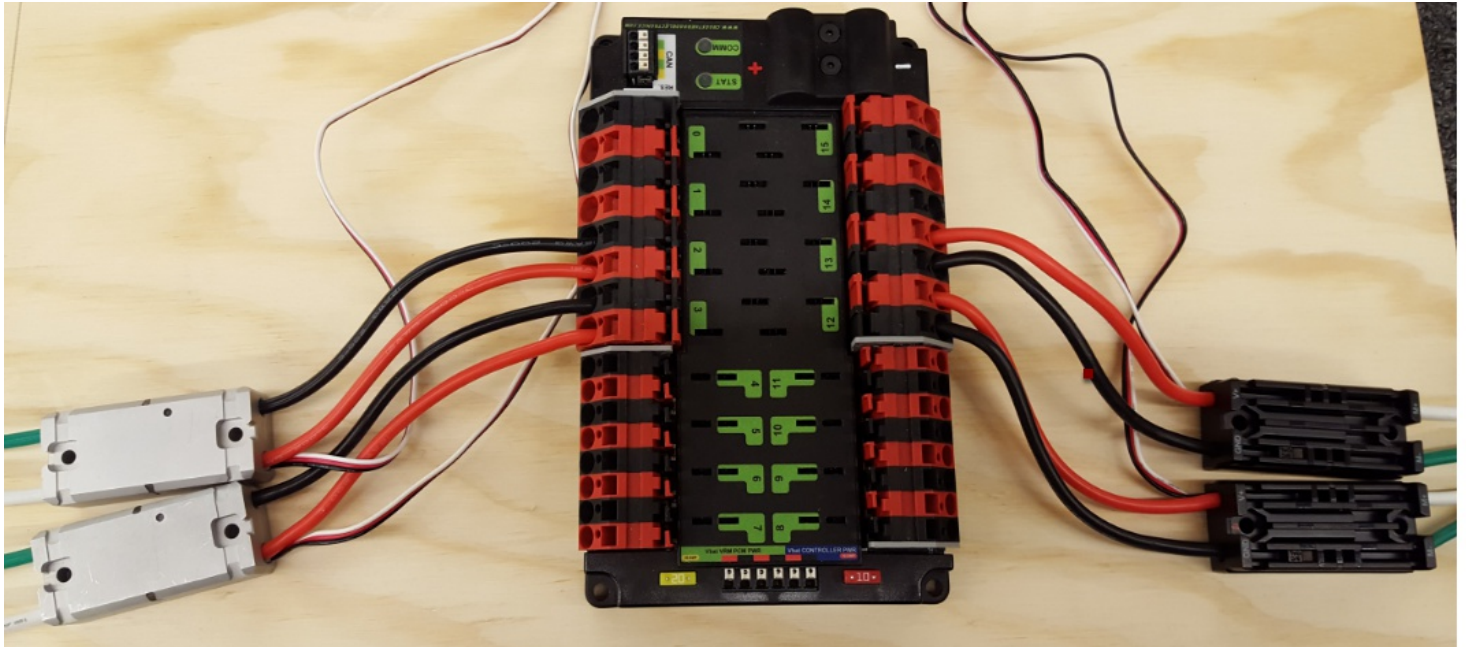
- Small Wago connector: Accepts 10AWG-24AWG, strip 11-12mm (~7/16")
- Large Wago connector: Accepts 6AWG-12AWG, strip 12-13mm(~1/2")

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, wire crimper

For each of the 4 Spark controllers (top 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 -)

For wire integrated motor controllers (bottom image):

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

Weidmuller Connectors

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

Getting Started with the Control System

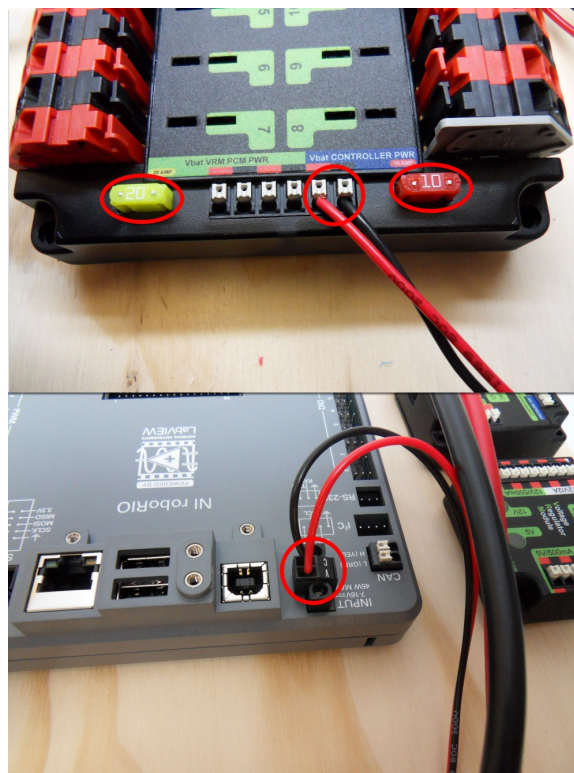
A number of the CAN and power connectors in the system use a Weidmuller LSF series wire-to-board connector. There are a few things to keep in mind when using this connector for best results:

- 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



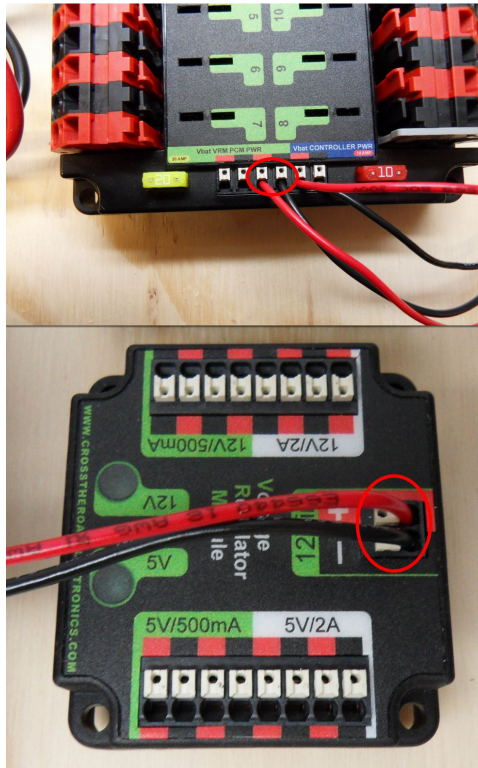
Requires: 10A/20A mini fuses, Wire stripper, very small flat screwdriver, 18AWG Red and Black

1. Insert the 10A and 20A mini fuses in the PDP in the locations shown on the silk screen (and in the image above)

Getting Started with the Control System

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



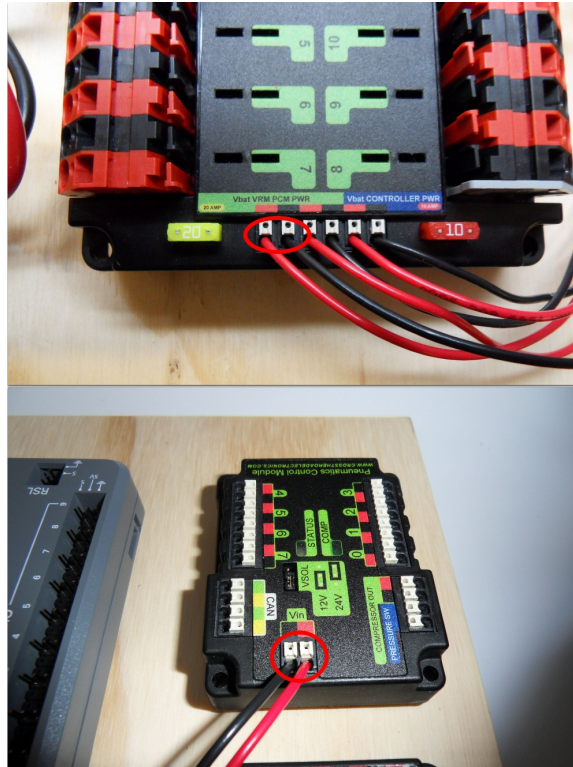
Requires: Wire stripper, small flat screwdriver (optional), 18AWG red and black wire

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.
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.

Getting Started with the Control System

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)



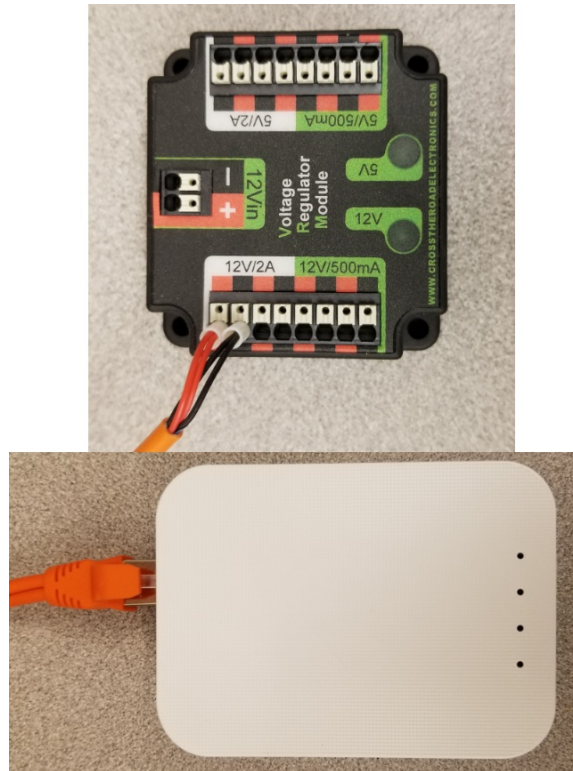
Requires: Wire stripper, small flat screwdriver (optional), 18AWG red and black wire

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

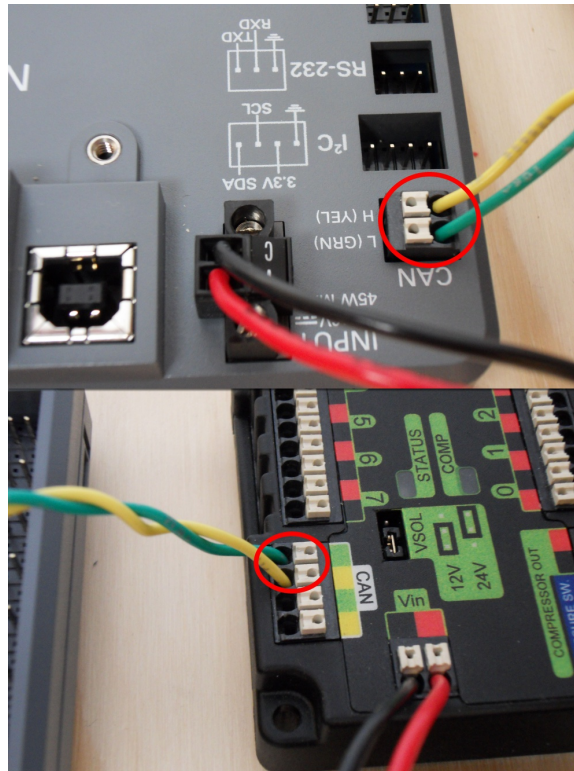


Requires: Ethernet cable

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



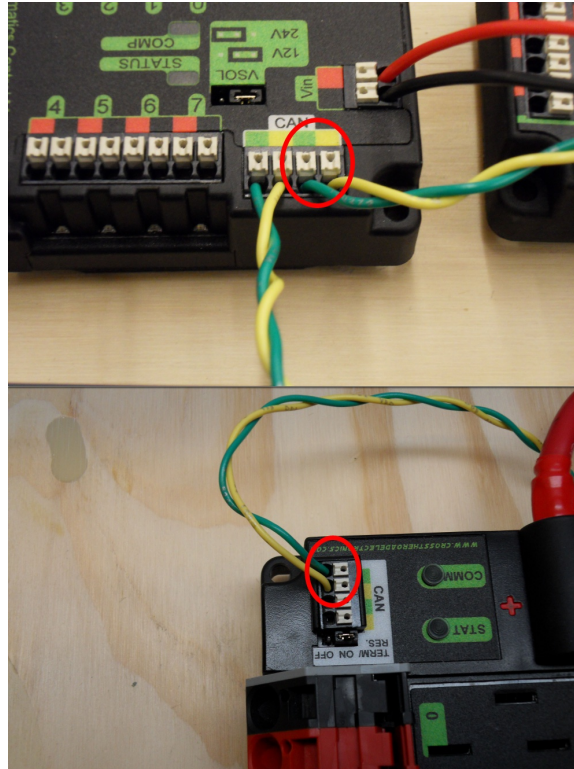
Requires: Wire stripper, small flat screwdriver (optional), yellow/green twisted CAN cable

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 (shown 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 (shown 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 (from Spark boxes), 2x PWM Y-cable (Optional)

Option 1 (Direct connect):

1. Connect the PWM cables from each Spark directly to the roboRIO. The black wire should be towards the outside of both the roboRIO and Spark. 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.

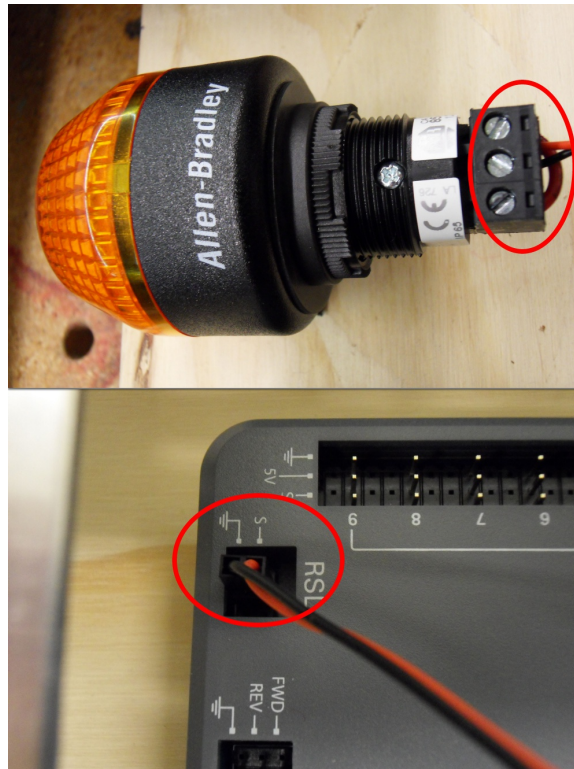
Option 2 (Y-cable):

1. Connect 1 PWM Y-cable to the PWM cables for the Sparks controlling one side of the robot. The brown wire on the Y-cable should match the black wire on the PWM cable.

Getting Started with the Control System

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



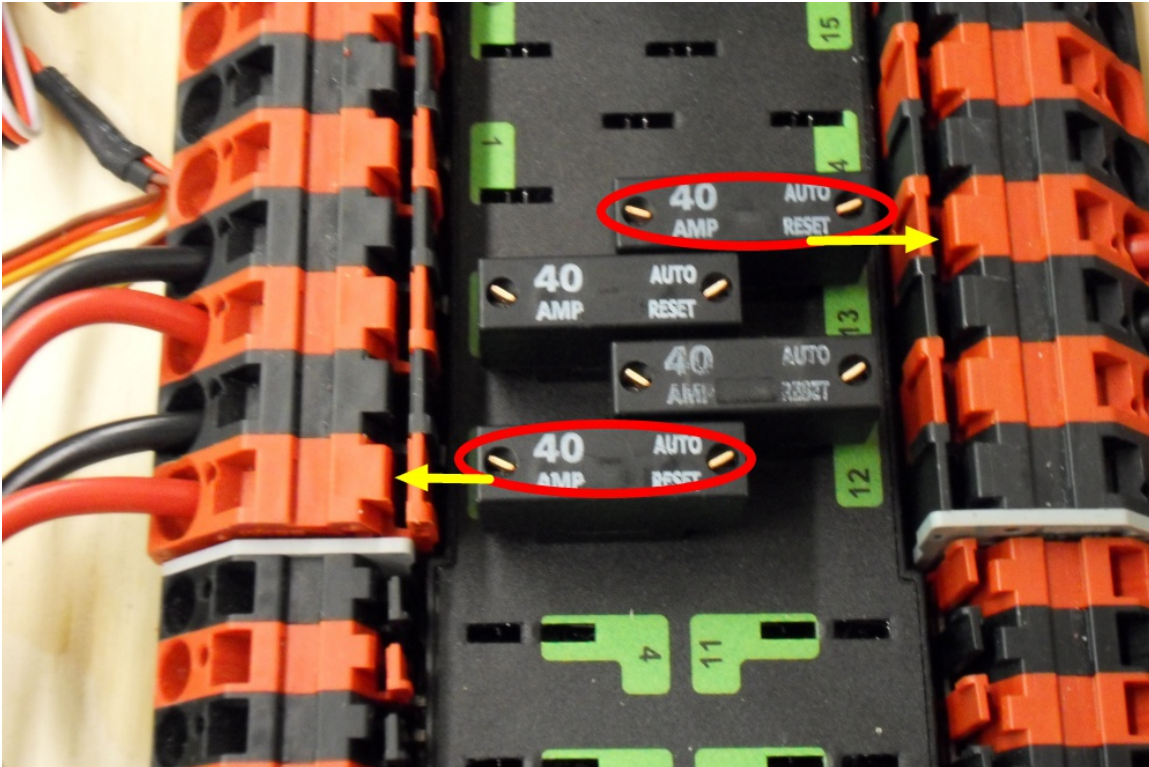
Requires: Wire stripper, 2 pin cable, Robot Signal Light, 18AWG red wire, very small flat screwdriver

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 zipties or Dual Lock (it is recommended to move the RSL to a more visible location as the robot is being constructed)

Circuit Breakers



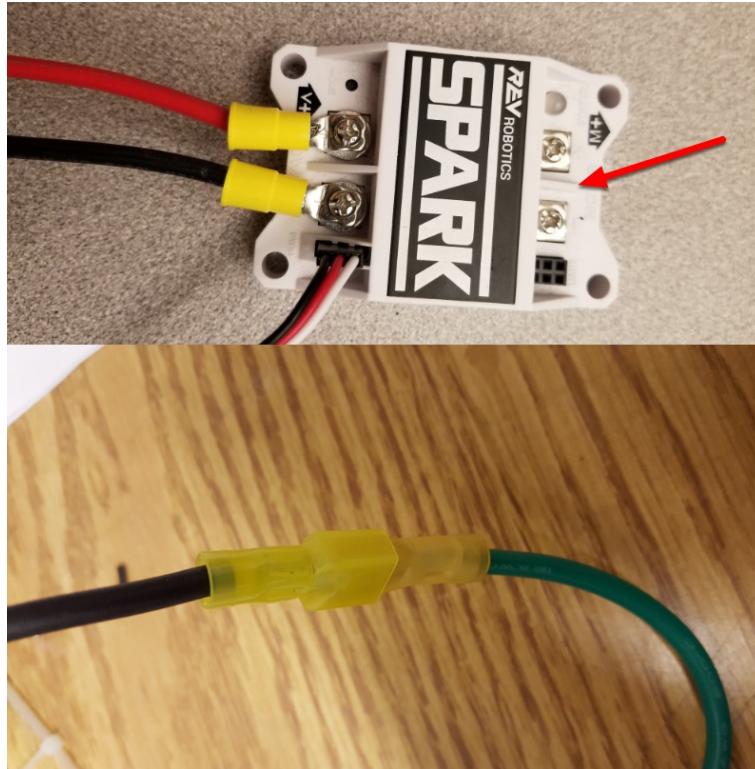
Requires: 4x 40A circuit breakers

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 Sparks or other non-integrated-wire controllers:

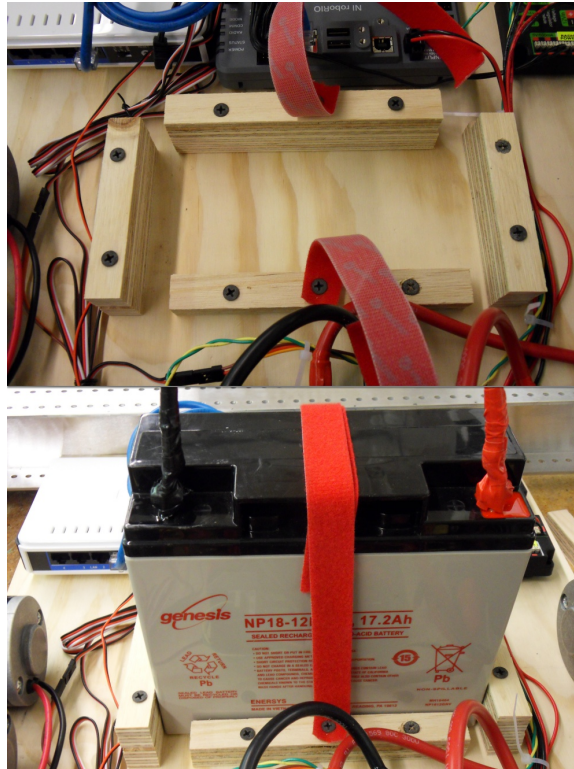
1. Crimp a ring/fork terminal on each of the motor wires.
2. Attach the wires to the output side of the motor controller (red to +, black to -)

For integrated wire controllers:

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

Getting Started with the Control System

Battery Box



Requires: Plywood Scraps, plywood cutting tool (e.g. saw), 10x 2" wood screws, drill, 1/8" drill bit, Philips head driver bit or philips head screwdriver, velcro wrap

Construct a battery box. the design shown uses scraps of plywood left over from cutting out the electronics board (4 pieces 4"x1.5" for the short sides of the battery stacked 2 high, 3 pieces 6"x1.5" for the front and back stacked 2 high in the back). Use the velcro wrap to make a pair of straps which will overlap to secure the battery.

Note: The battery box shown here is an example, sufficient for driving the robot. Teams should ensure that their battery will be securely held in their final design in the face of potentially violent robot-to-robot collision.

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

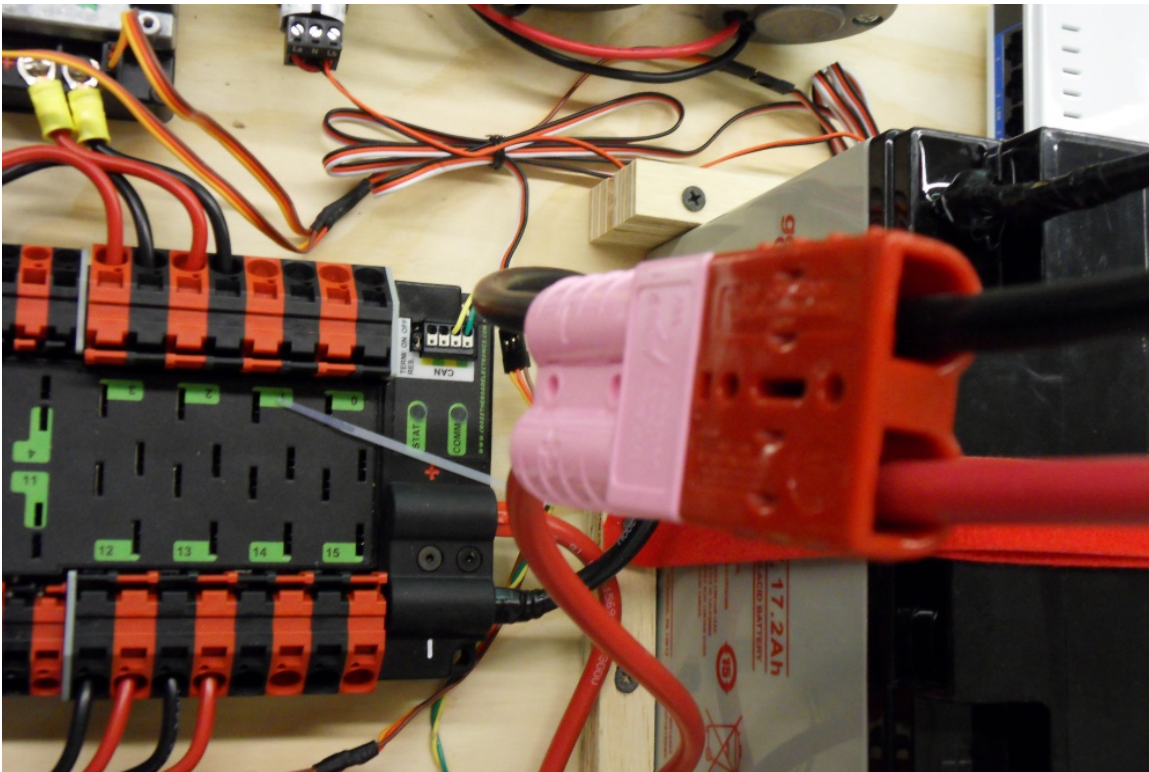
It is also recommended to put the robot on blocks so the wheels are off the ground before proceeding. This will prevent any unexpected movement from becoming dangerous.

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.

Getting Started with the Control System

Rookie USB stick

The computer provided in the Rookie Kit of Parts comes preloaded with an existing Windows install. The computer **should not be reimaged** and no image is provided for doing so. Veteran teams may not need to image their machines, but should see the section on [Image Download](#) if they wish to do so.


Rookie USB Stick

As the computer comes pre-loaded with Windows, the Rookie USB stick does not contain an image as it has in prior seasons. Instead it contains a copy of the LabVIEW DVD contents and pre-downloaded copies of Eclipse for C++, Eclipse for Java and Game Specific DLC.

Teams should follow the [Getting Started with the Control System](#) pages, and simply skip download steps for components they already have on the USB drive. Note: To use Eclipse, teams will need to download the Java JDK in addition to the FRC specific plugins.

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 you will need either a USB optical drive or to copy the contents of the LabVIEW DVD to a USB Flash drive then you can start with the appropriate document for [C++\Java](#), [LabVIEW](#), or [DS only](#).

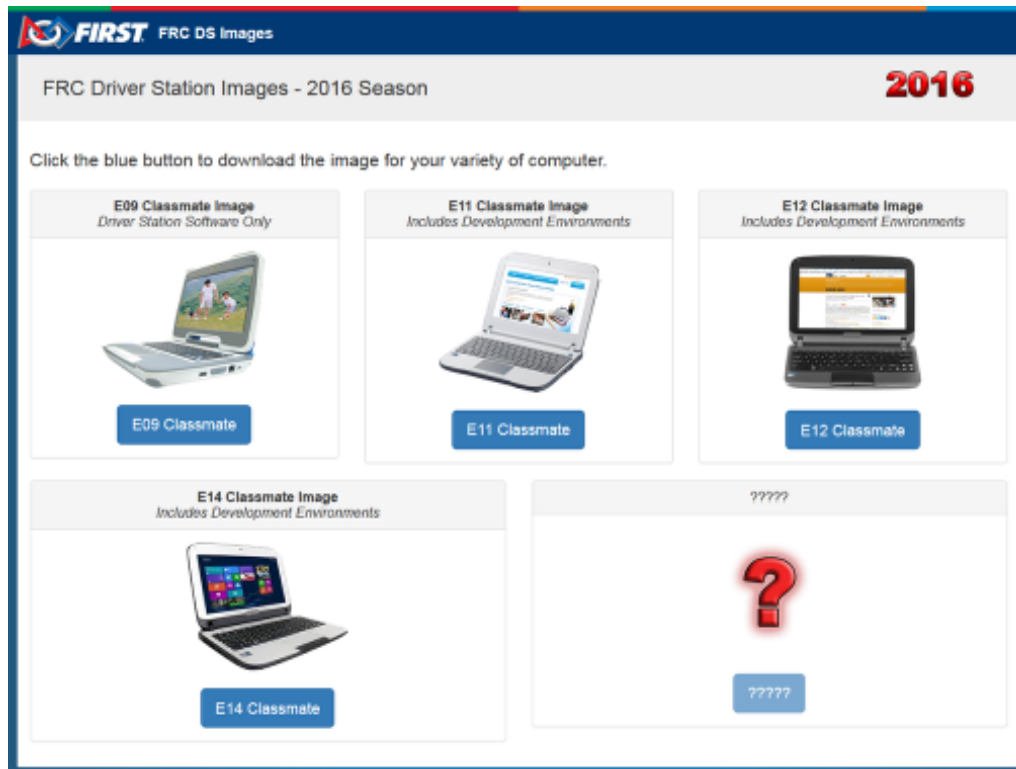
 Note: 2018 links are expected to be posted on or before Thursday, January 4th.

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 \(www.7zip.org\)](http://www.7zip.org) As of the writing of this document, the current released version is 9.20 (2010-11-18)
4. RMprepUSB software installed. [Download here \(http://www.rmrepusb.com/documents/release-2-0\)](http://www.rmrepusb.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 2.1.725

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 and Eclipse already downloaded.

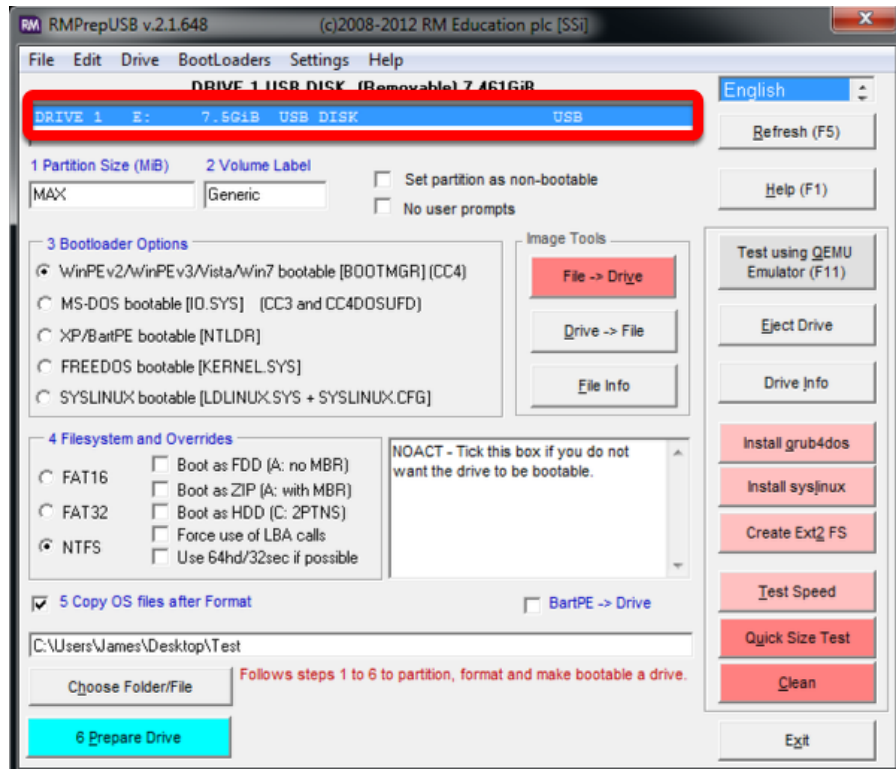
NOTE: These images only install the prerequisite core FRC software, it is still necessary to install the FRC specific updates. See the [Update Software](#) step for more information.

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.

Getting Started with the Control System

RMPrep

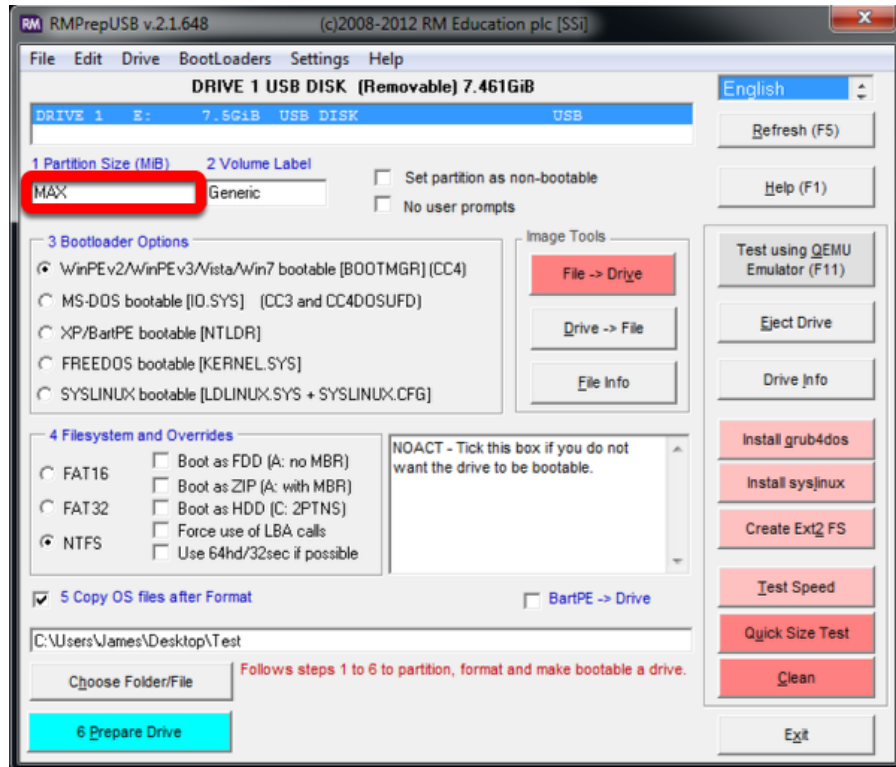


Start/Run RMPrepUSB

Select USB Drive

Getting Started with the Control System

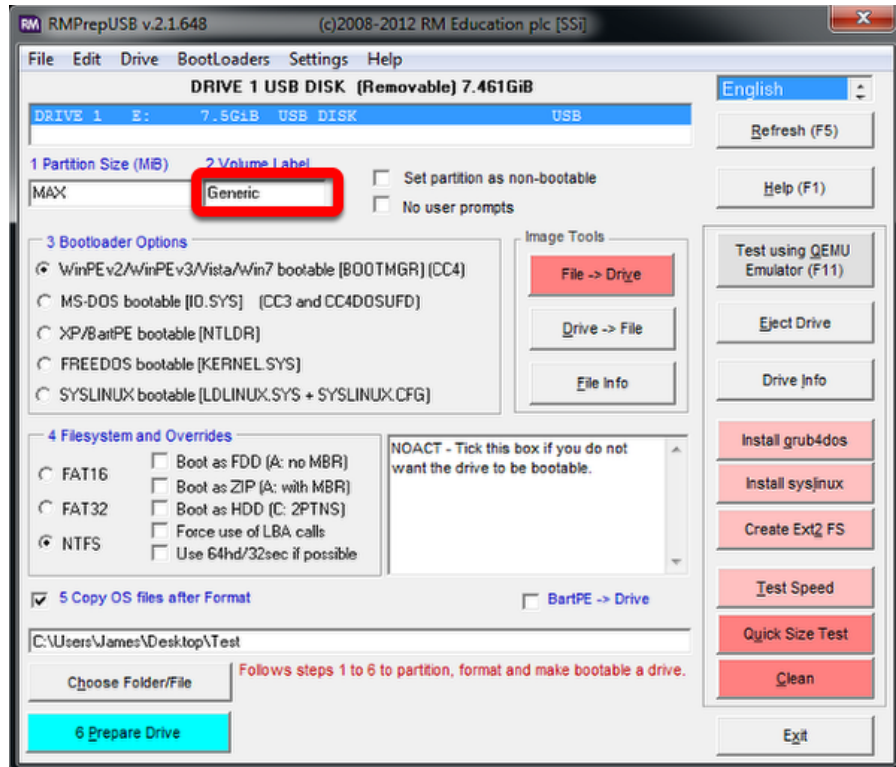
Set Partition Size



Set Partition Size to MAX

Getting Started with the Control System

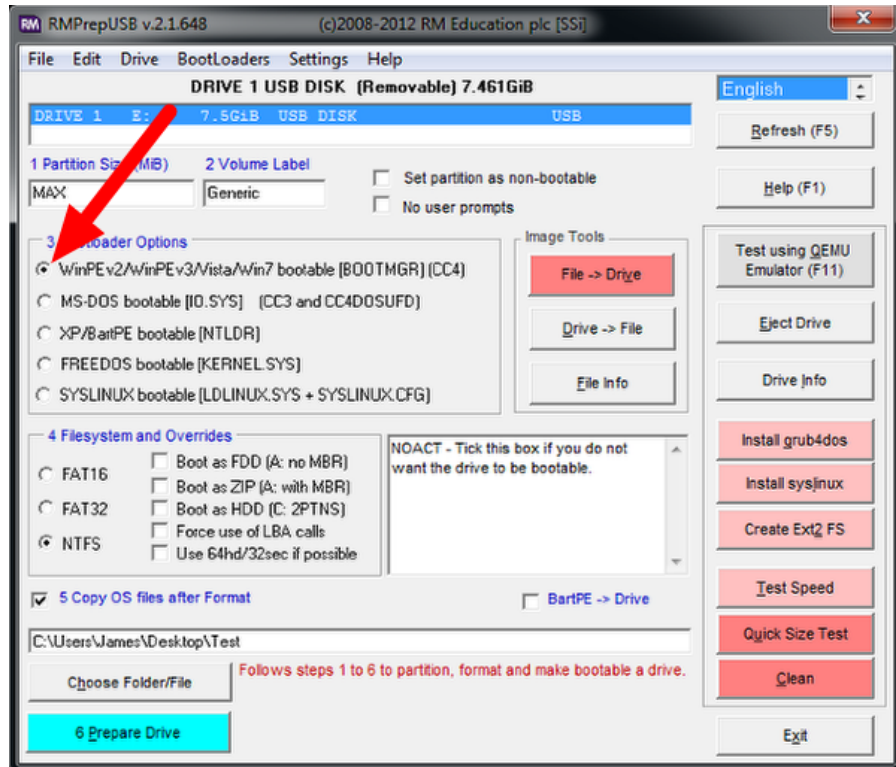
Set Volume Label



Set Volume Label to Generic

Getting Started with the Control System

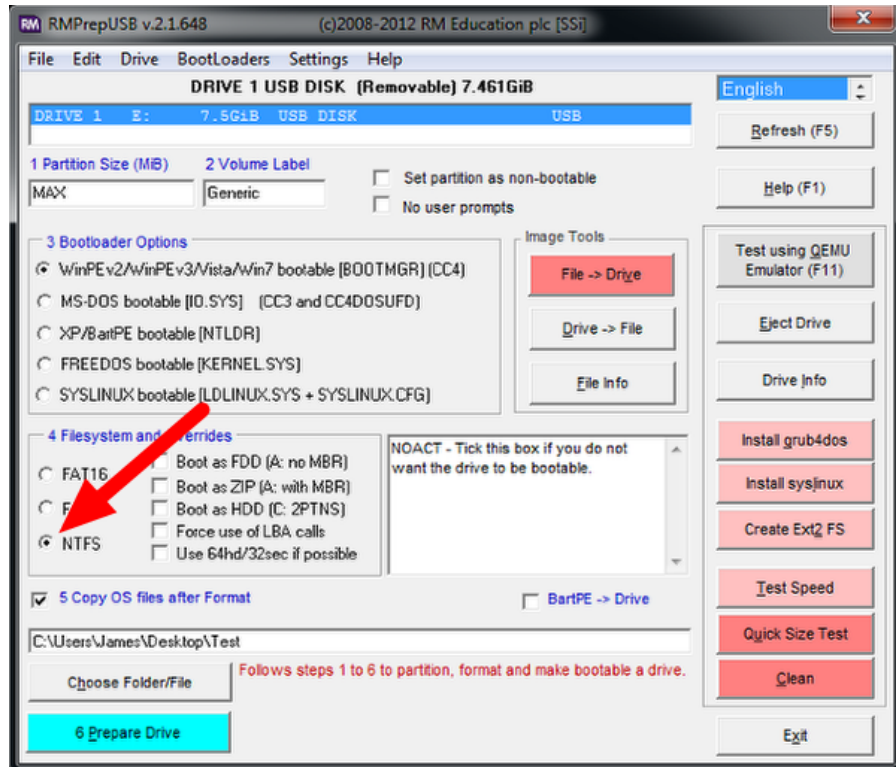
Set Bootloader Option



Select Bootloader Option “WinPE v2/WinPE v3/Vista/Win7 bootable”

Getting Started with the Control System

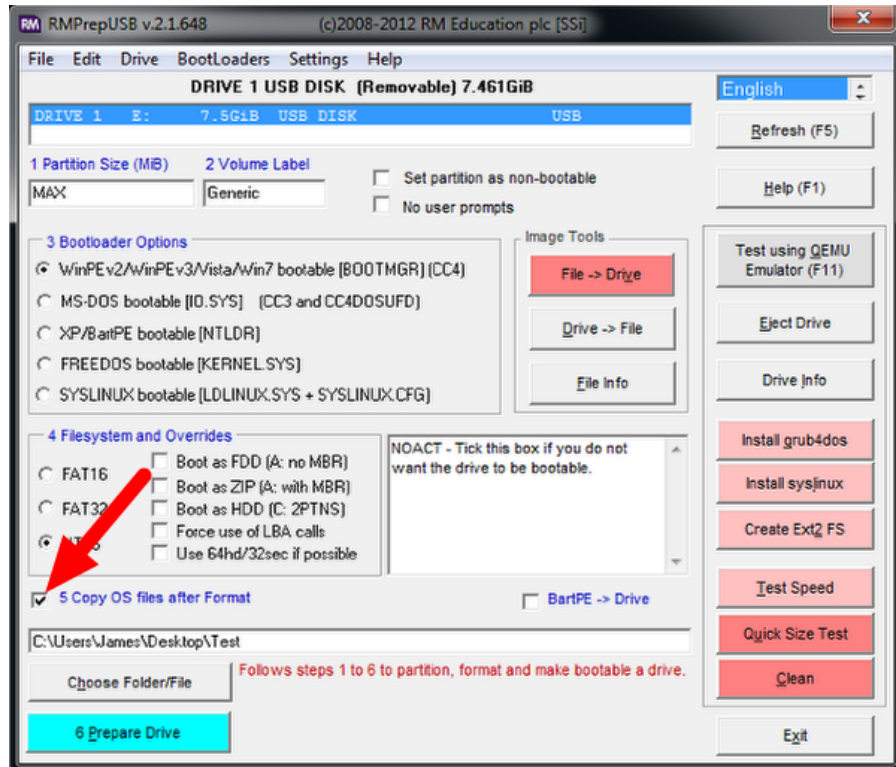
Select Filesystem



Select NTFS Filesystem

Getting Started with the Control System

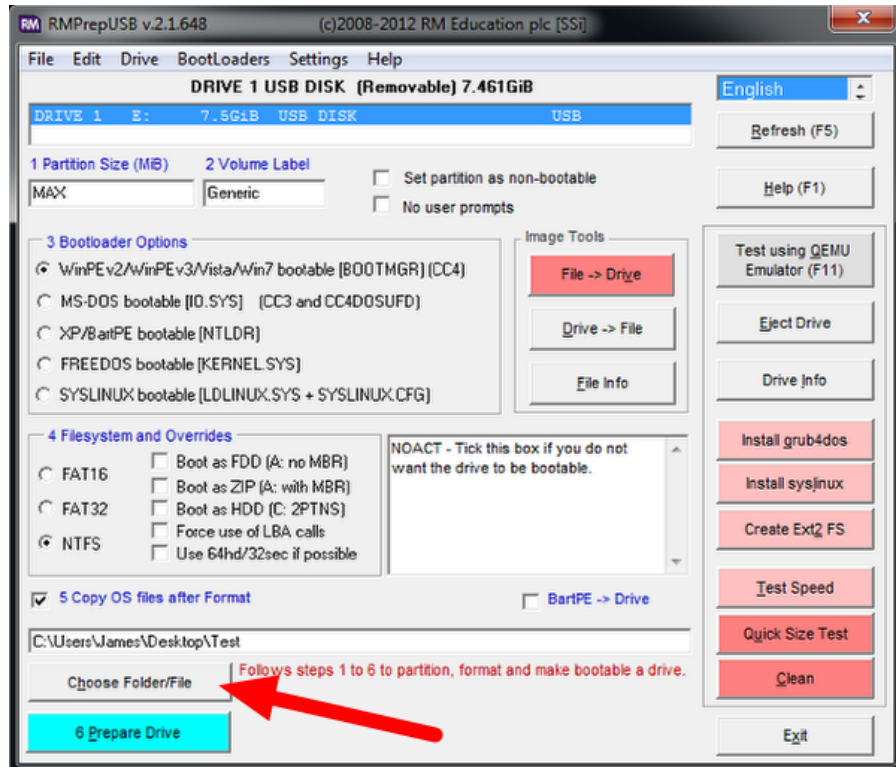
Copy OS Files Option



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

Getting Started with the Control System

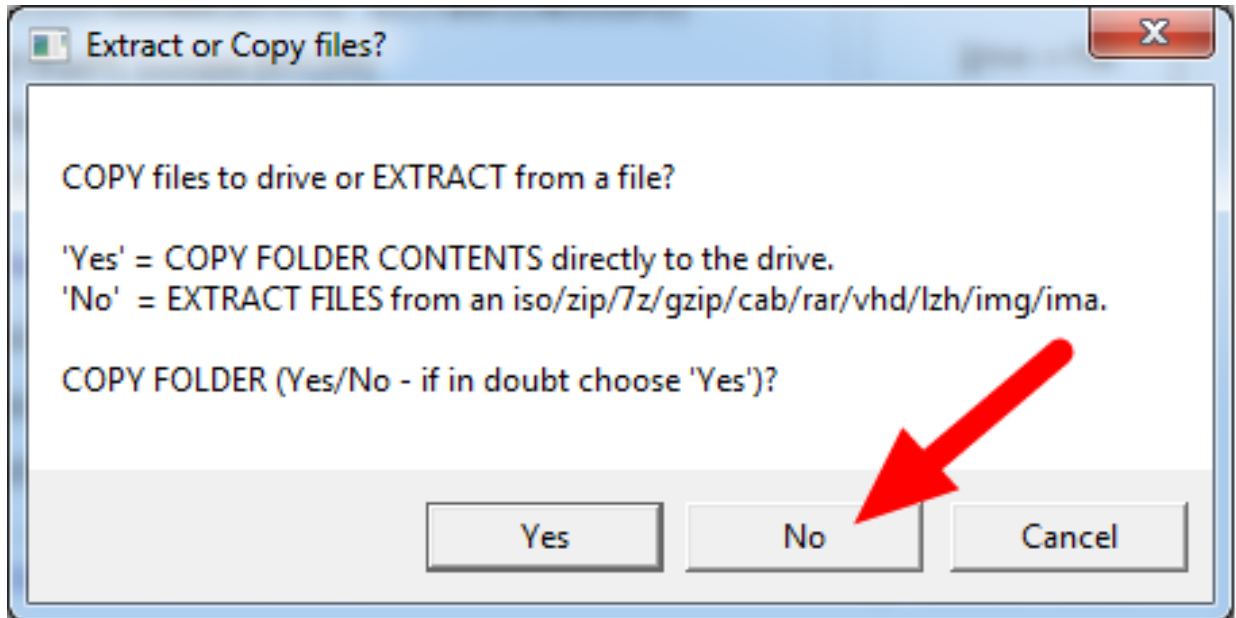
Locate Image



Select the “Choose Folder/File” button

Getting Started with the Control System

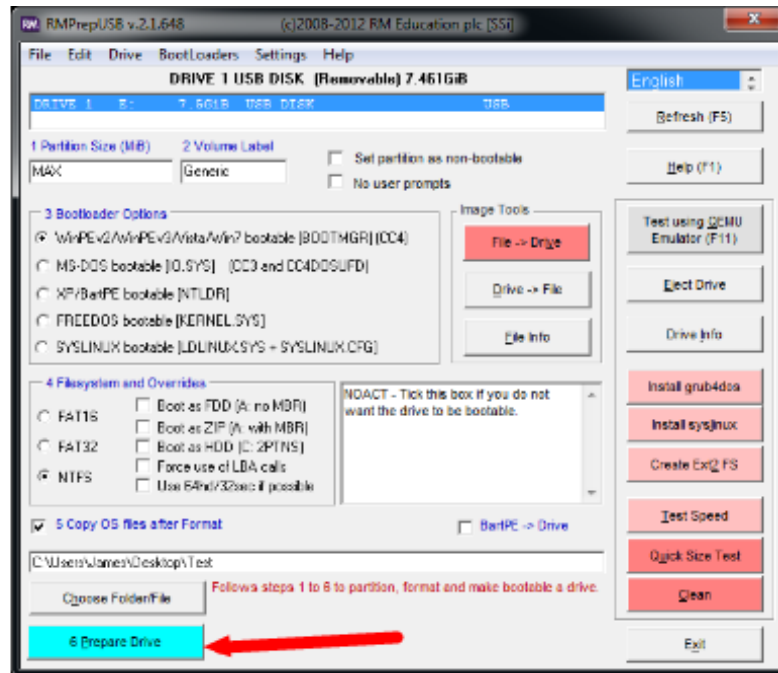
Copy Files Dialog



Choose "No" and select your .7z image

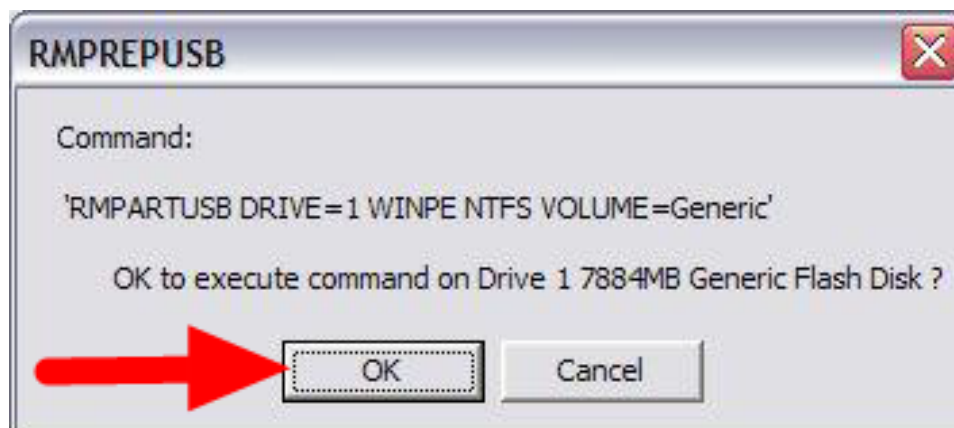
Getting Started with the Control System

Prepare Drive



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

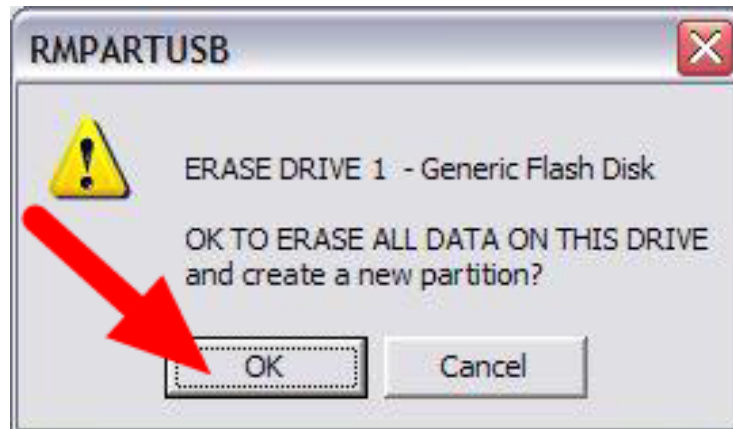
Confirmation Dialog 1



Getting Started with the Control System

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

Confirmation Dialog 2



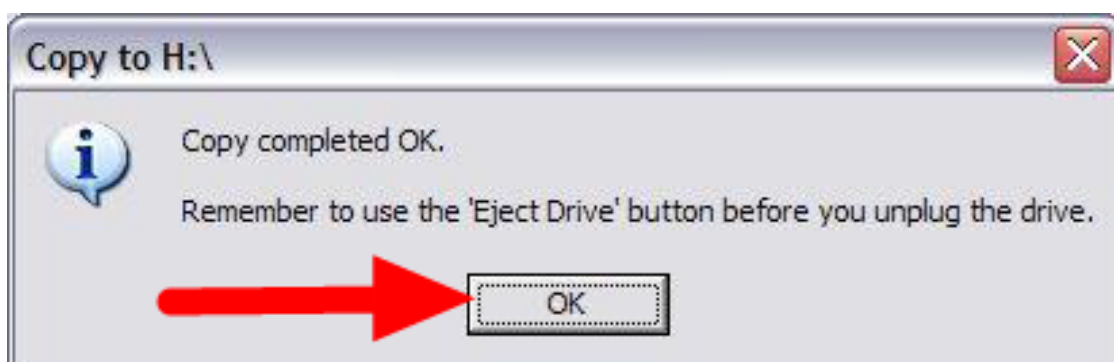
Click "OK" to format the USB drive

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.

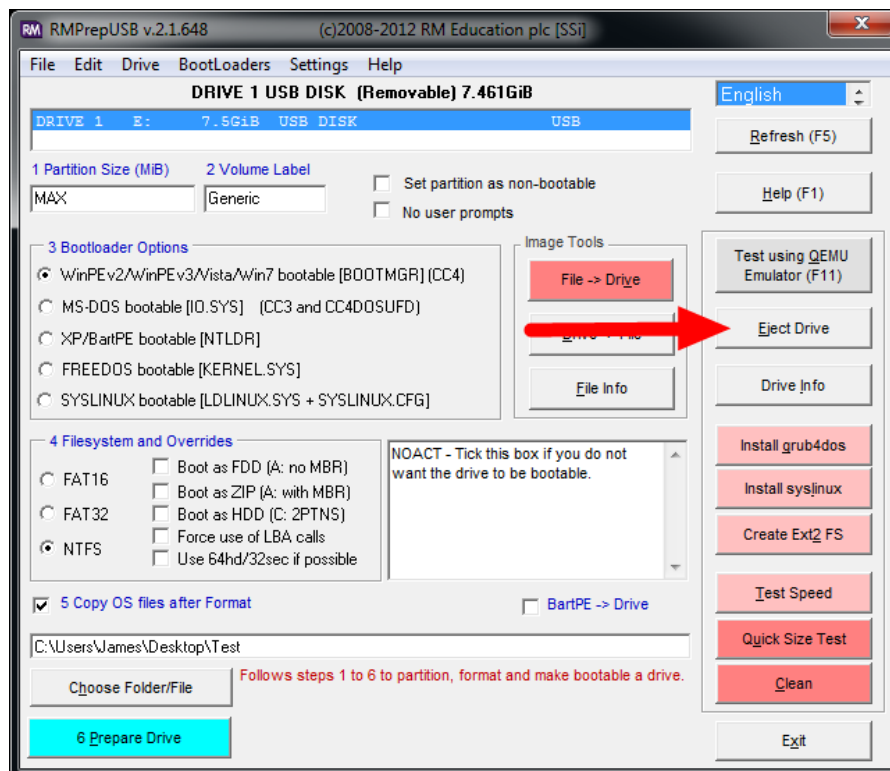
Copy Complete



Getting Started with the Control System

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



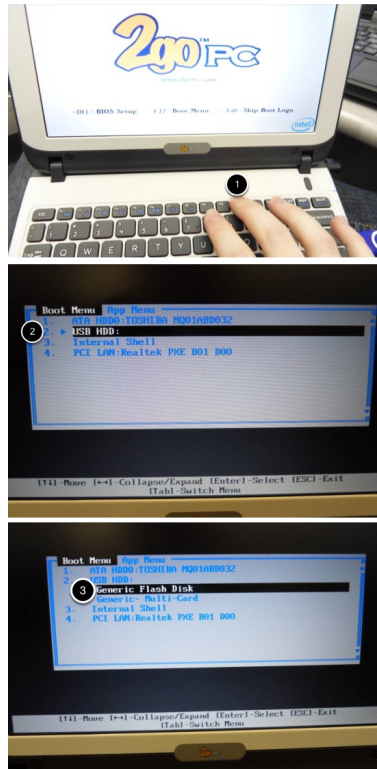
Press the “Eject Drive” button to safely remove the USB drive. The USB drive is now ready to be used to restore the image onto the PC.

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.

Getting Started with the Control System

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.

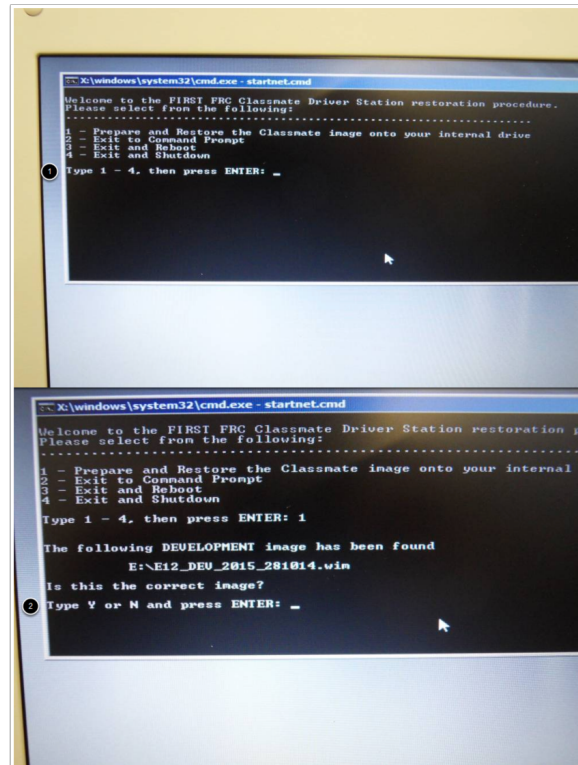
Acer ES1

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.

Getting Started with the Control System

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 reimage 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.

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.

Getting Started with the Control System

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

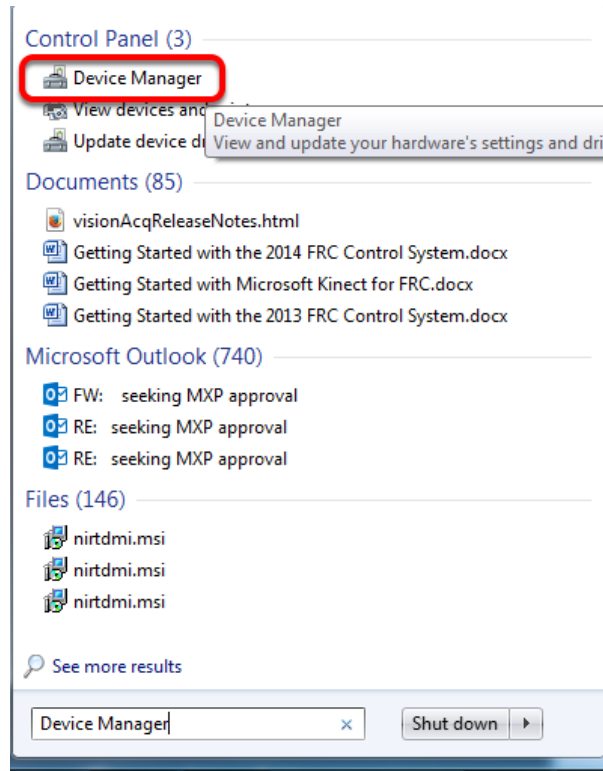
1. Establish an Internet connection.
2. Once you have an Internet connection, click the Start menu, right click "Computer" and click "Properties".
3. Scroll to the bottom section, "Windows activation", and Click "Activate Windows now"
4. Click "Activate Windows online now". The activation may take a 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

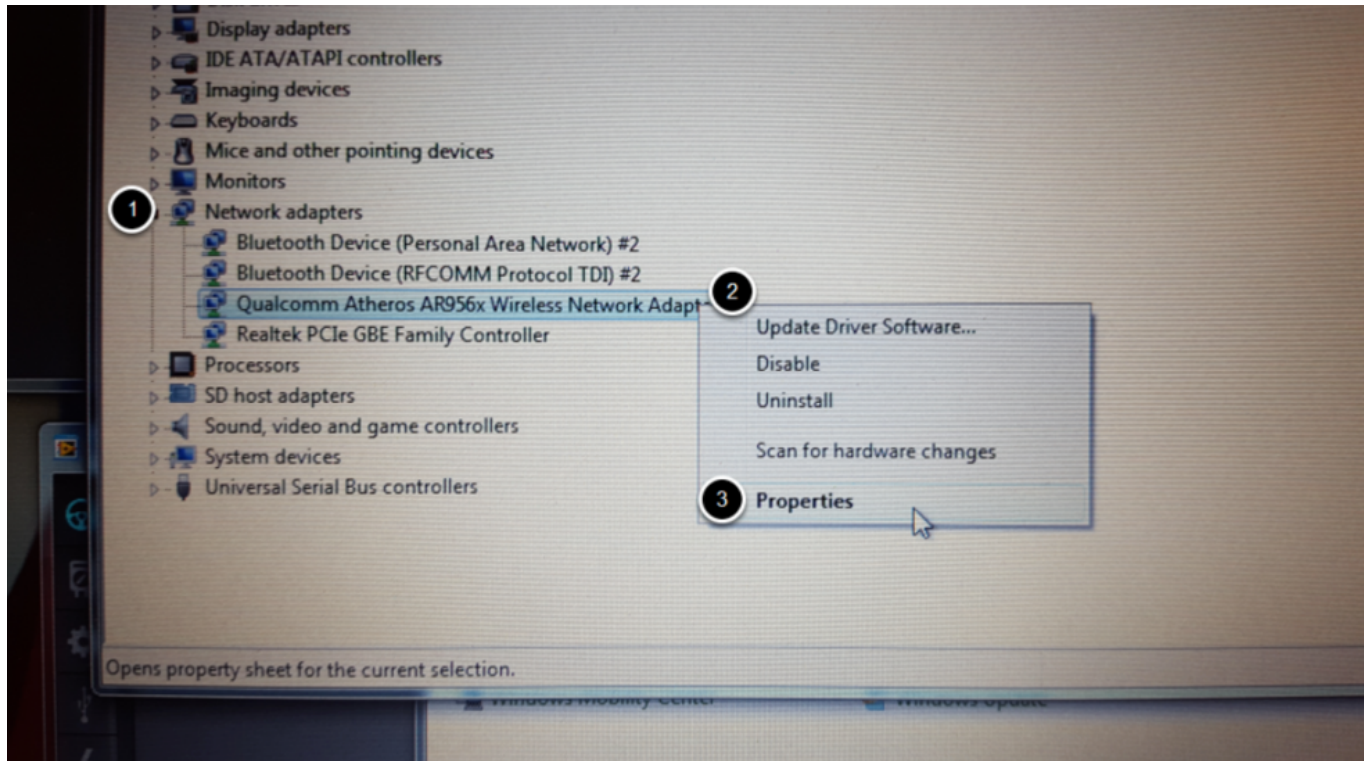


Acer ES1 PC only!

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

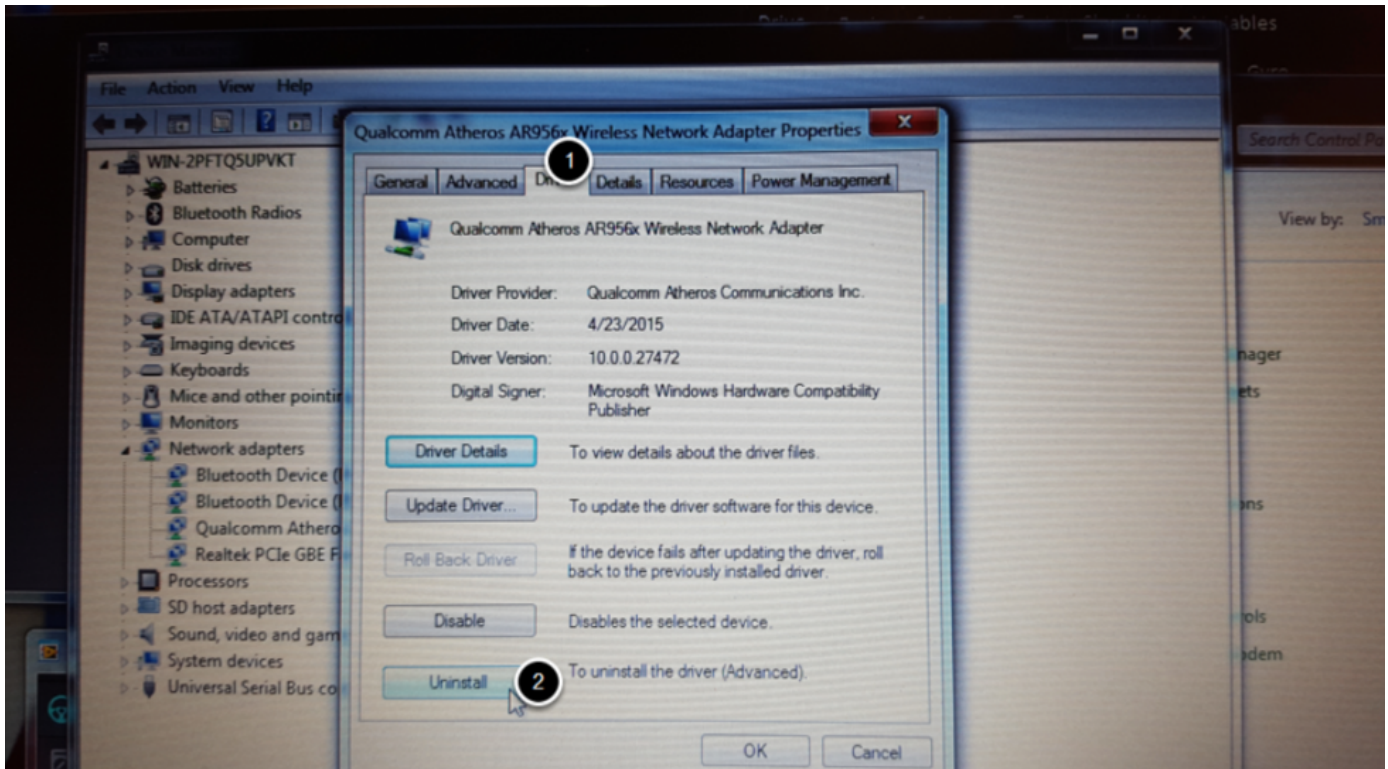
Open Wireless Device Properties



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

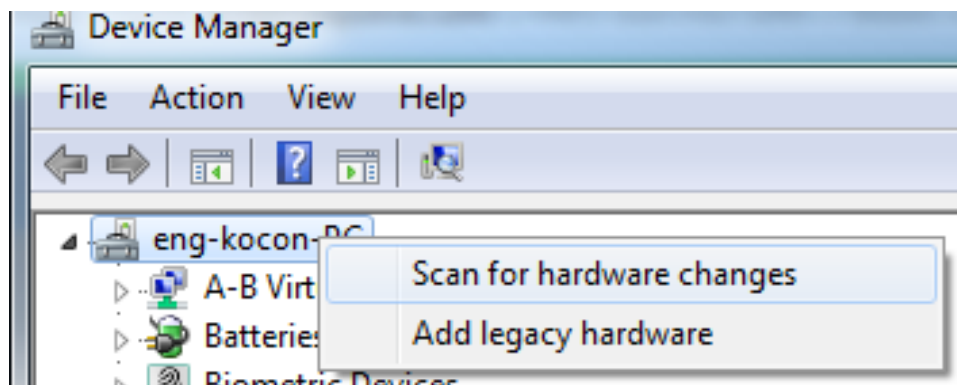
Getting Started with the Control System

Uninstall Driver



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

Scan for New Hardware



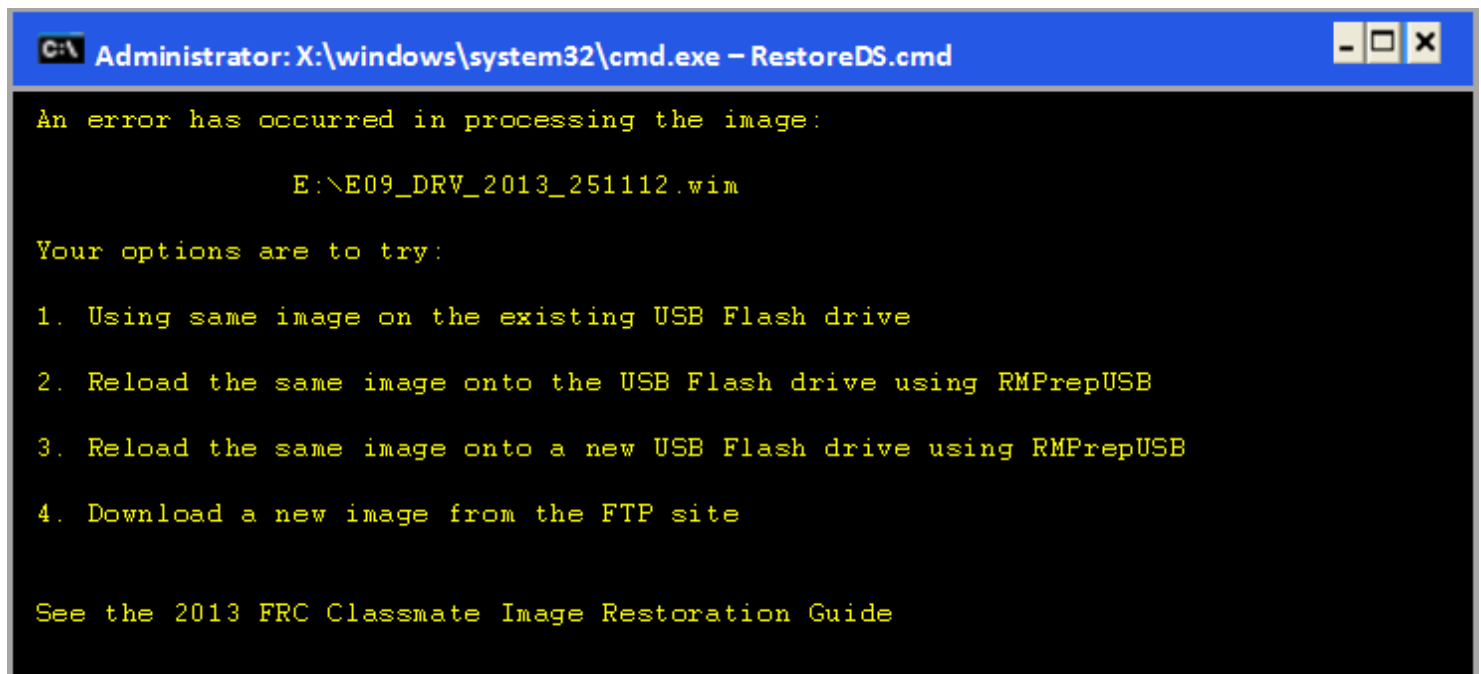
Getting Started with the Control System

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.

Update Software

In order for the Classmates to arrive at Kickoff locations in time, they were shipped before the final version of the software was ready. To use the software for FRC some additional components will need to be installed. LabVIEW teams should continue with the [Installing the FRC 2016 Update Suite \(All Languages\)](#) article. C++ or Java teams should continue with [Installing Eclipse \(C++/Java\)](#) but can skip to configuring Eclipse and installing the FRC plugins.

Errors during Imaging Process



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

Getting Started with the Control System

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](#).

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

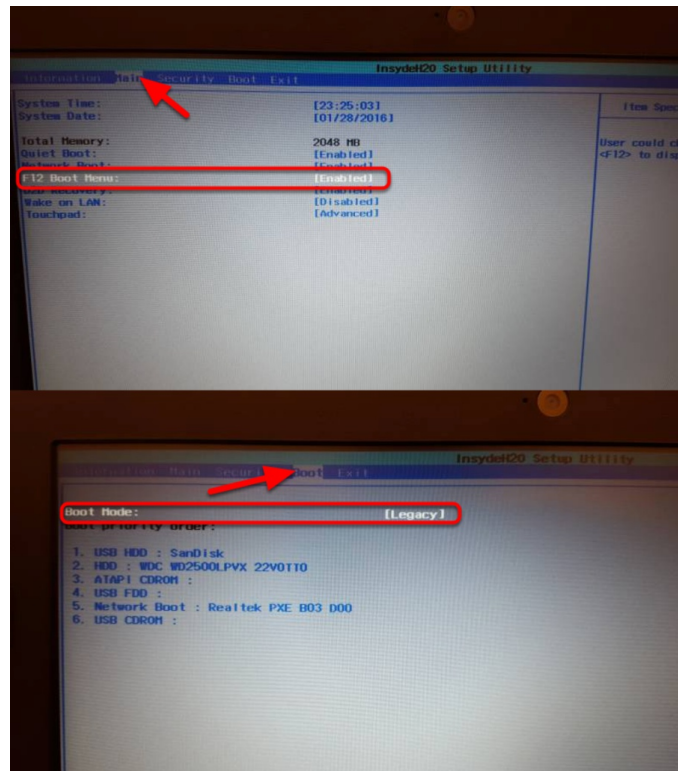
Reload the same image onto a new USB Flash drive using RMPrepUSB: The error message displayed may also be caused by an error with the USB Flash drive itself. Press any key to return to the main menu and select #4 to safely shutdown the Classmate. Select a new USB Flash drive and follow the steps starting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

Installing LabVIEW for FRC 2018 (LabVIEW only)

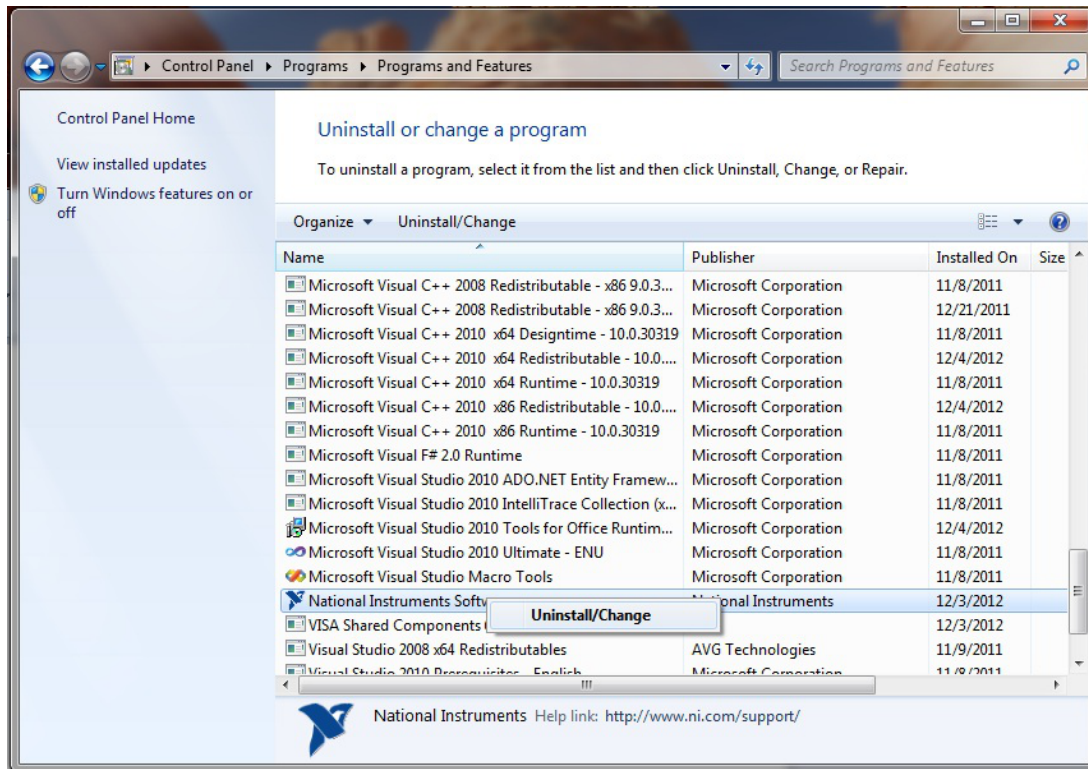


Note: This installation is for teams programming in LabVIEW or using NI Vision Assistant only. C++ and Java teams not using these features do not need to install from the DVD.

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)

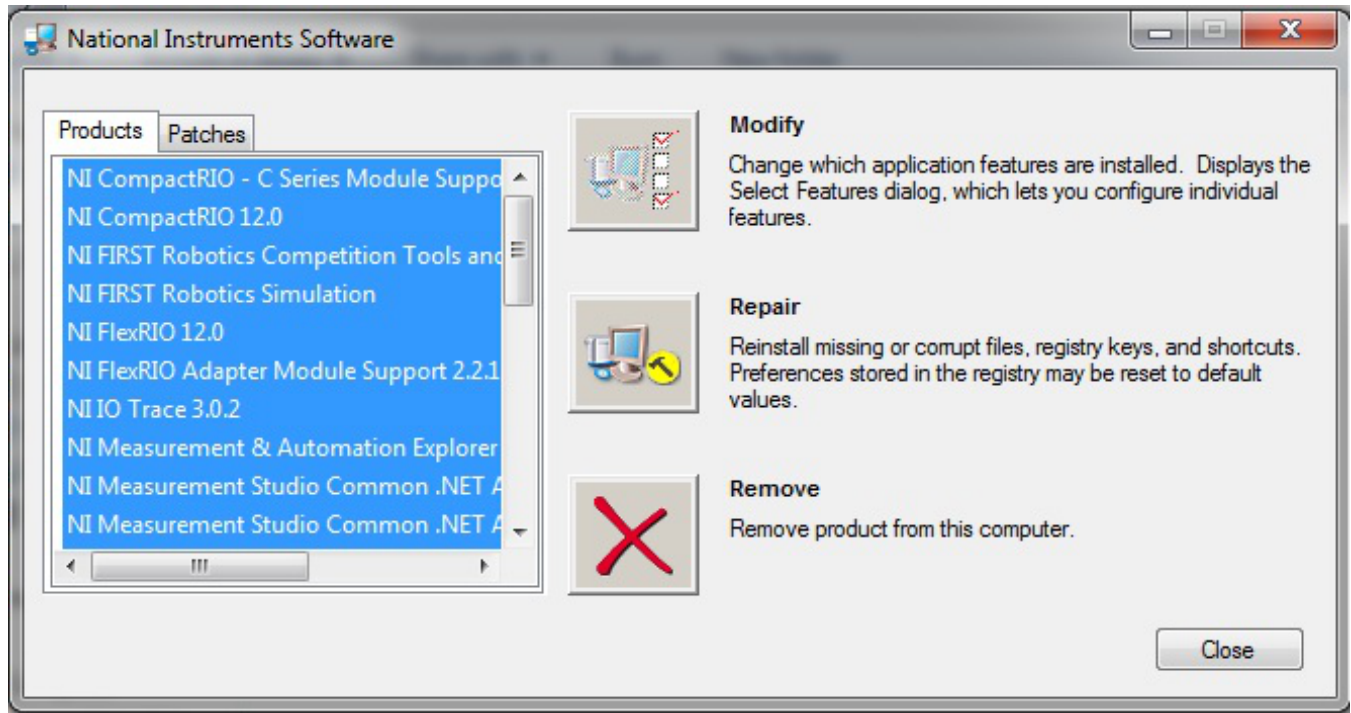


NOTE: If you wish to keep programming cRIOs you will need to maintain an install of LabVIEW for FRC 2014. The LabVIEW for FRC 2014 license has been extended. While these versions should be able to co-exist on a single computer, this is not a configuration that has been extensively tested.

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 2017 only. Make sure to back up any team code located in the "User\LabVIEW Data" directory before uninstalling. 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 DVD or download the LabVIEW 2018 installer from <http://www.ni.com/download/labview-for-frc-17.0/7185/en/>

If downloaded, right click on the downloaded file (NI_FRC2018.zip) and select Extract All.

Note: This is a large download (~4GB). 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 2018 season expires on January 31, 2019)

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.

Getting Started with the Control System

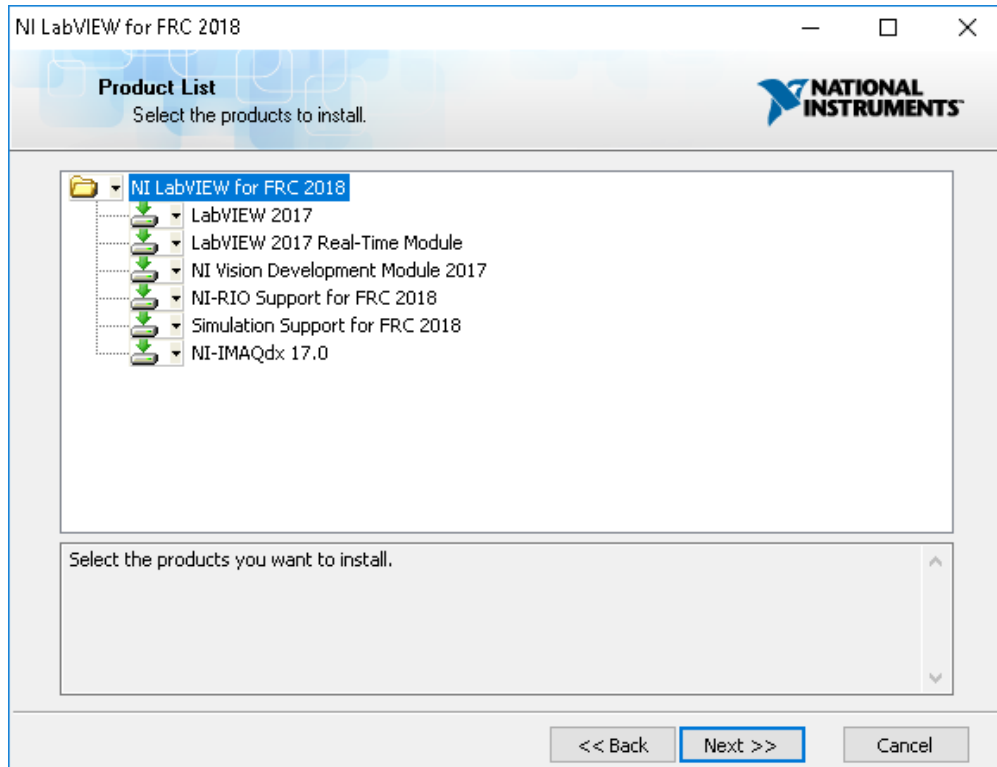
Warnings



Click "Next"

Getting Started with the Control System

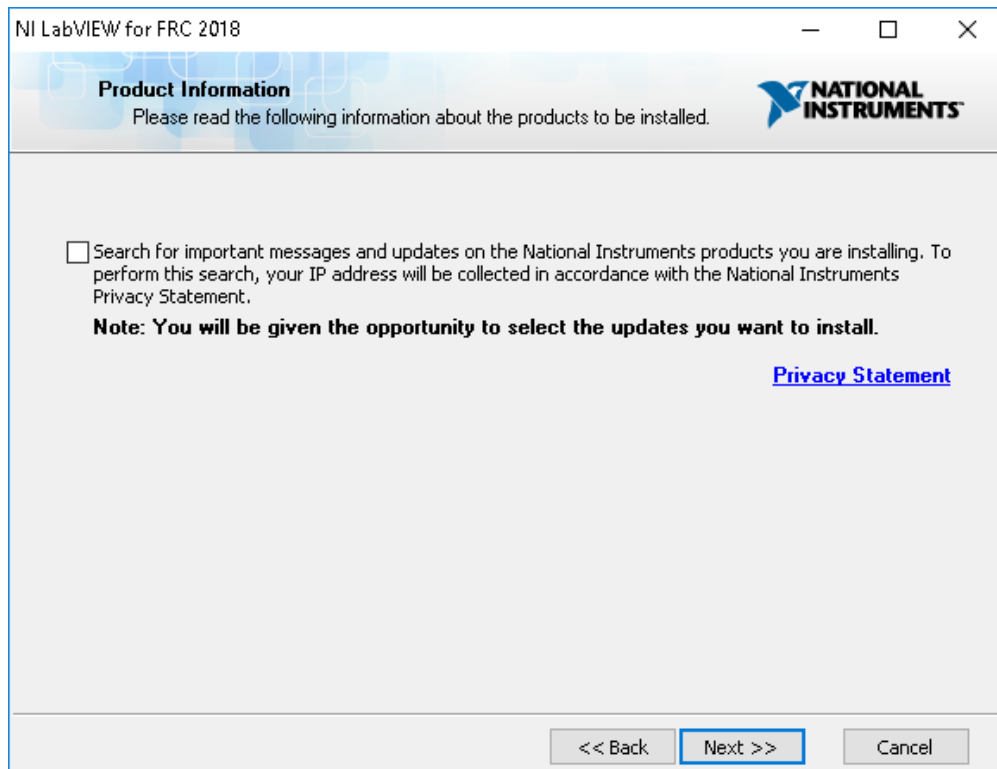
Product List



Click "Next"

Getting Started with the Control System

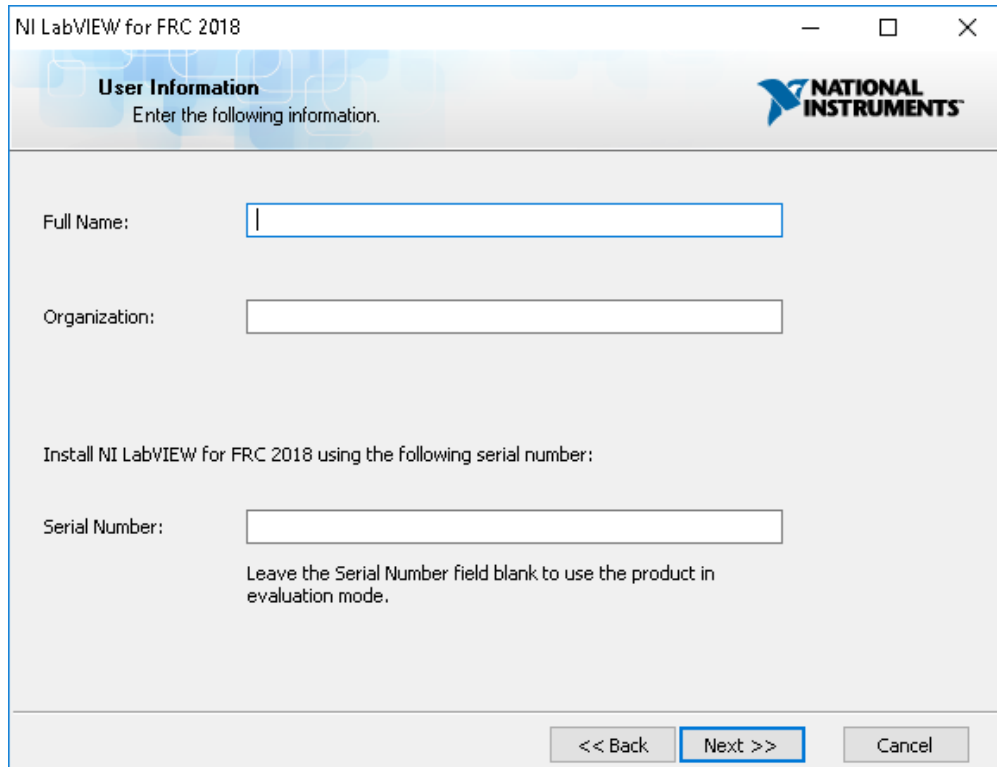
Product Information



Un-check the box, then click "Next"

Getting Started with the Control System

User Information

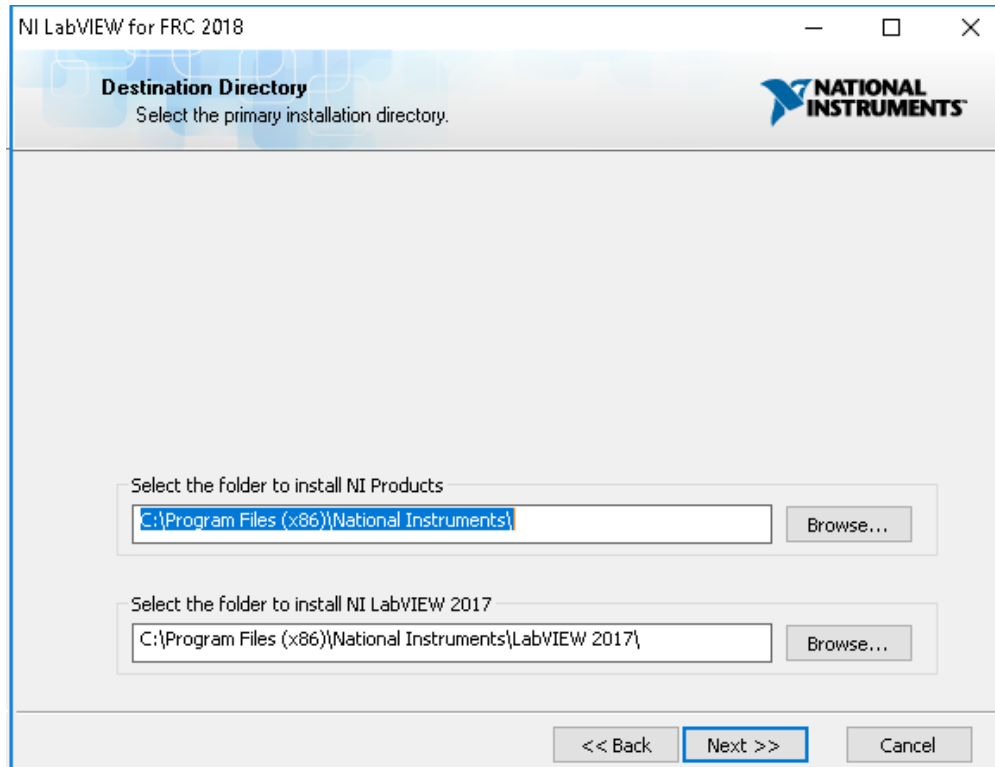


The image shows a Windows-style dialog box titled "NI LabVIEW for FRC 2018". The dialog has a header bar with the title and standard window controls (minimize, maximize, close). Below the header, the text "User Information" is displayed in bold, followed by "Enter the following information." and the National Instruments logo. The main area of the dialog contains three input fields: "Full Name:" with a text box, "Organization:" with a text box, and "Serial Number:" with a text box. Below the "Serial Number" field, there is a note: "Leave the Serial Number field blank to use the product in evaluation mode." At the bottom of the dialog, there are three buttons: "<< Back", "Next >>" (which is highlighted with a blue border), and "Cancel".

Enter name, organization, and the serial number from the LabVIEW packet in your KOP. Click "Next". If you cannot find your serial number, please reach out to National Instruments at www.ni.com/frc/needhelp.

Getting Started with the Control System

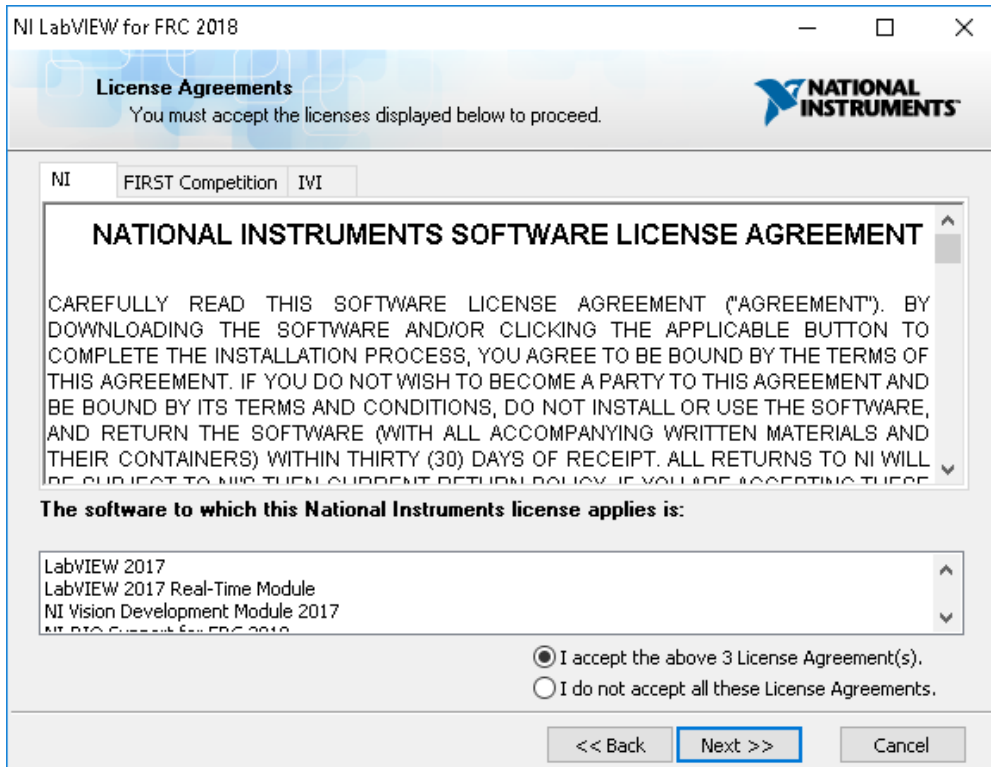
Destination Directory



Click "Next"

Getting Started with the Control System

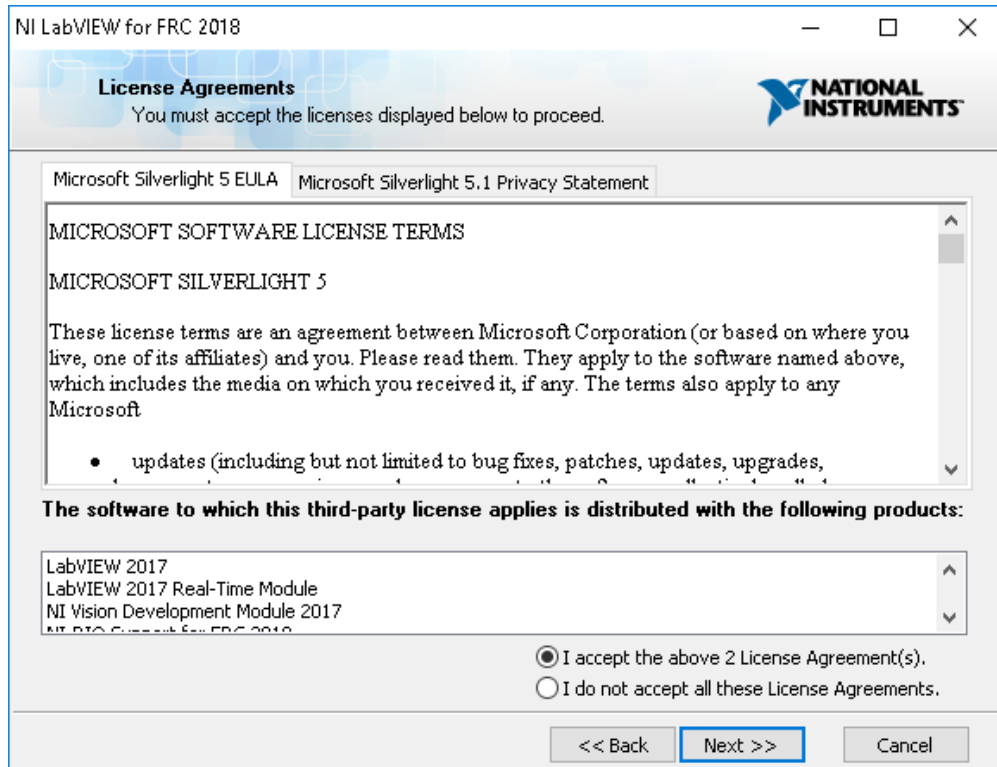
License Agreements (1)



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

Getting Started with the Control System

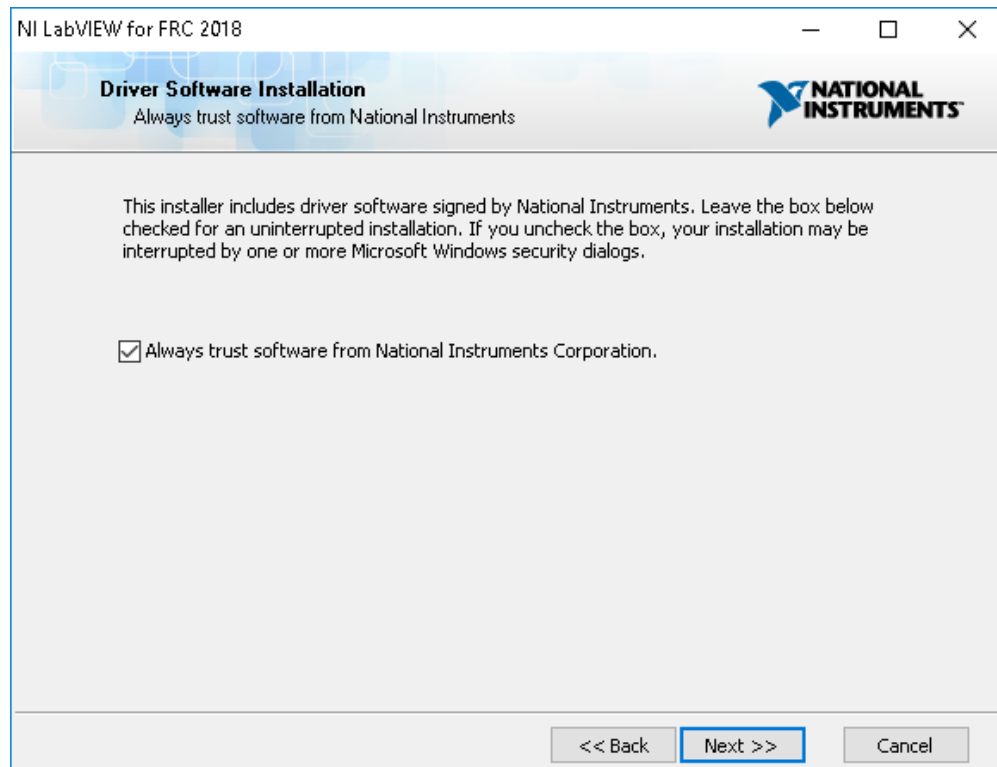
License Agreements (2)



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

Getting Started with the Control System

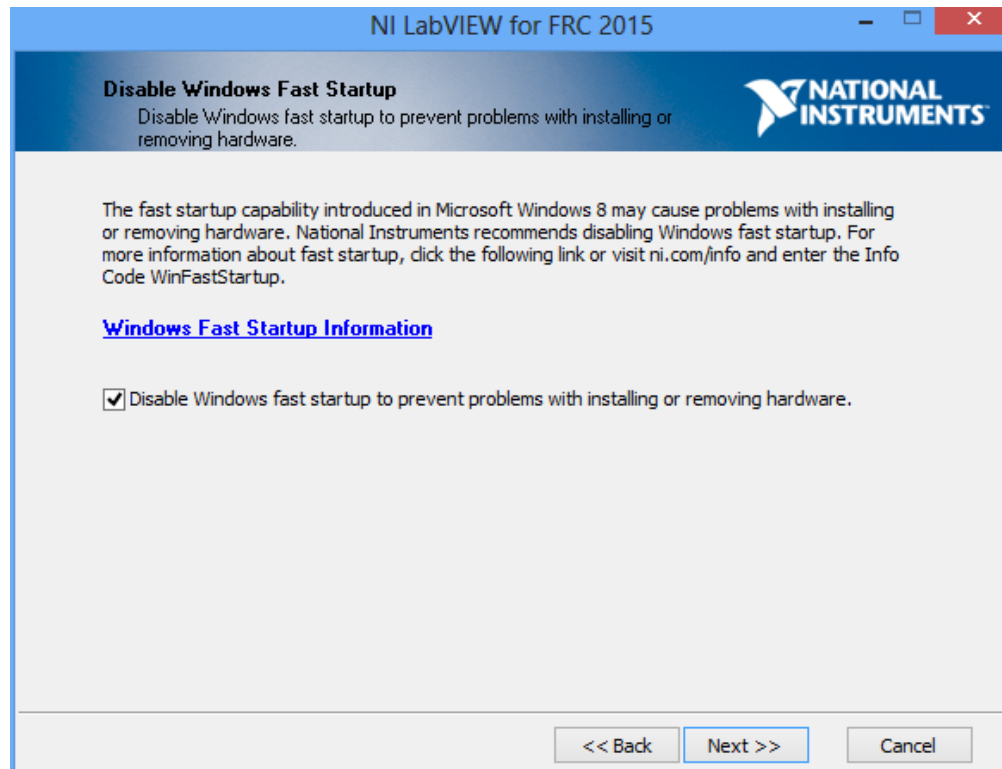
Driver Software Installation



If you see this screen, Click "Next"

Getting Started with the Control System

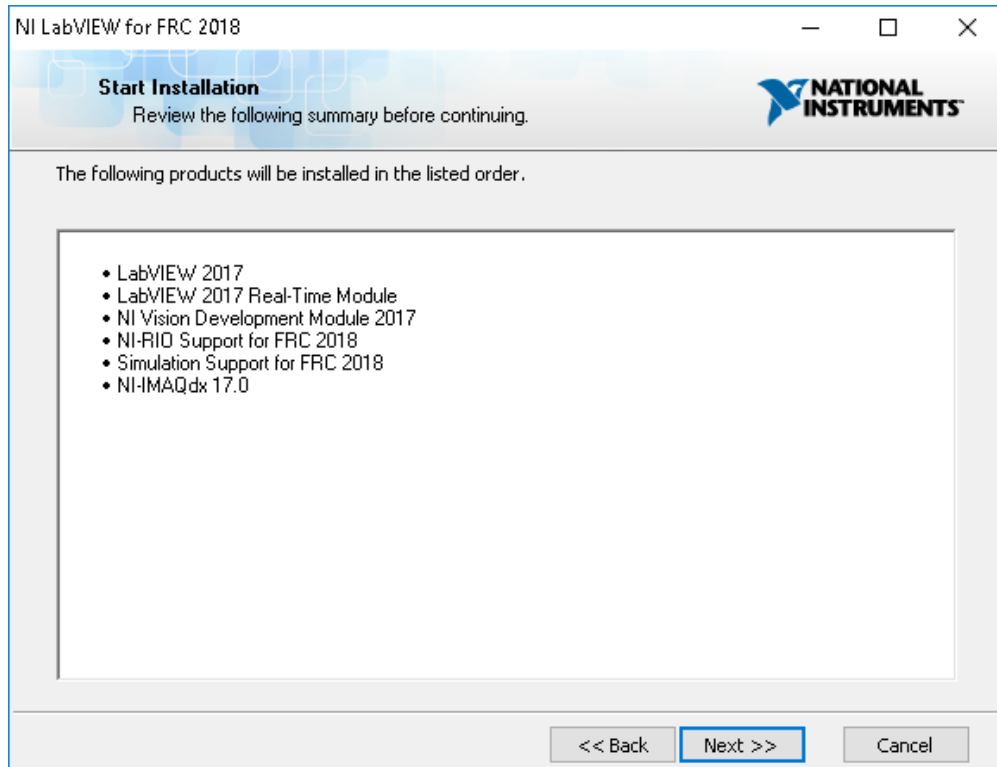
Disable Windows Fast Startup



If you see this screen, click "Next"

Getting Started with the Control System

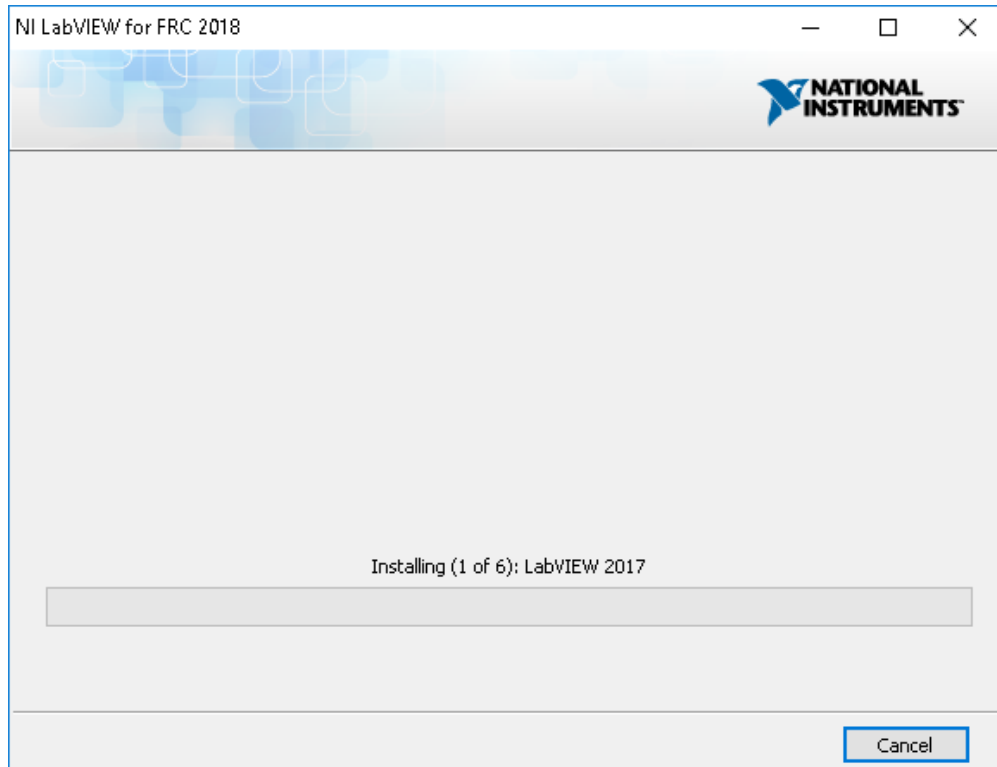
Start Installation



Click "Next"

Getting Started with the Control System

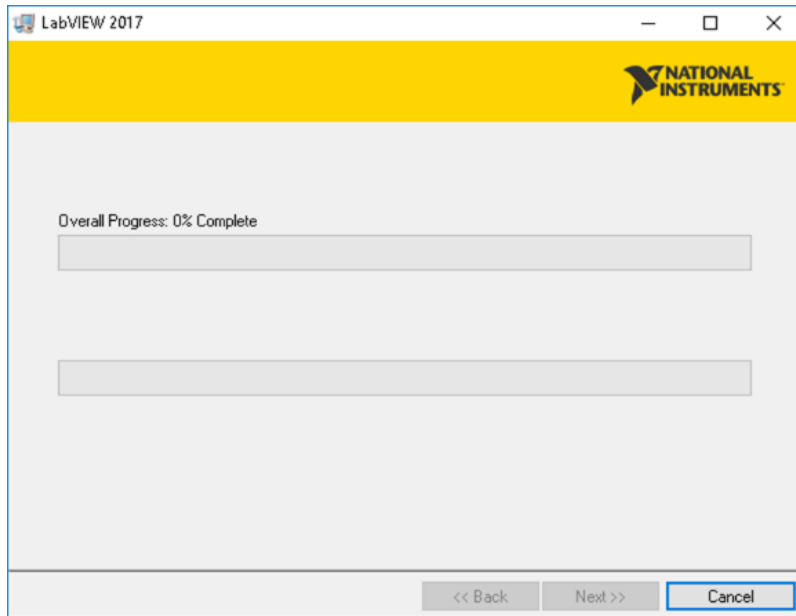
Overall Progress



Overall installation progress will be tracked in this window

Getting Started with the Control System

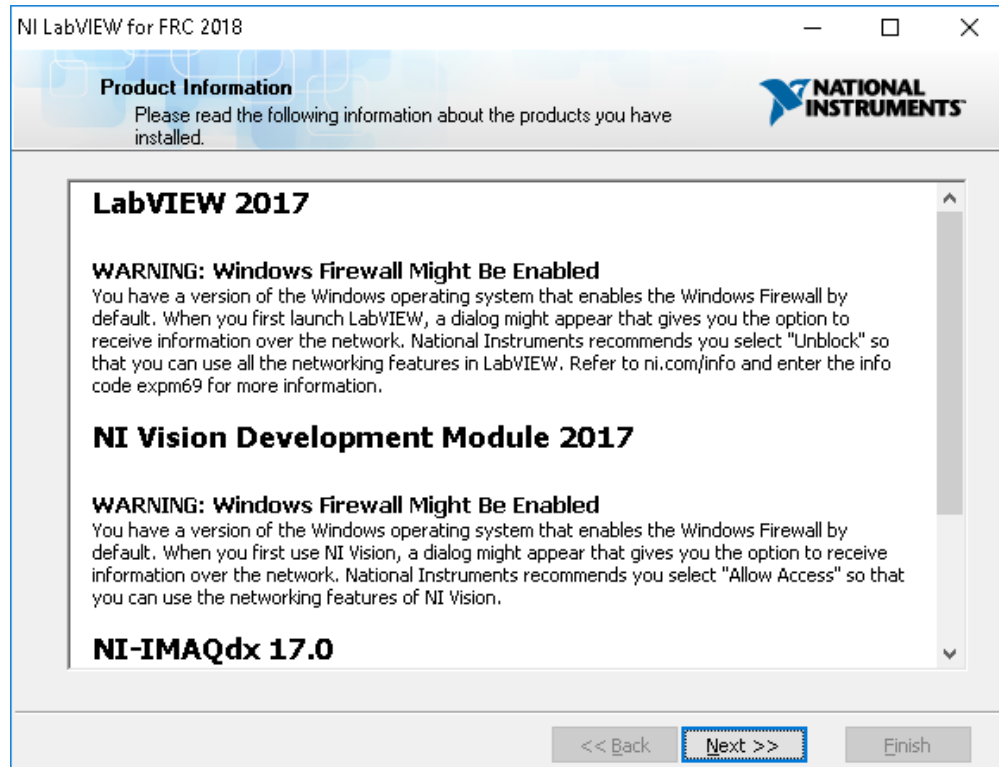
Individual Product Progress



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

Getting Started with the Control System

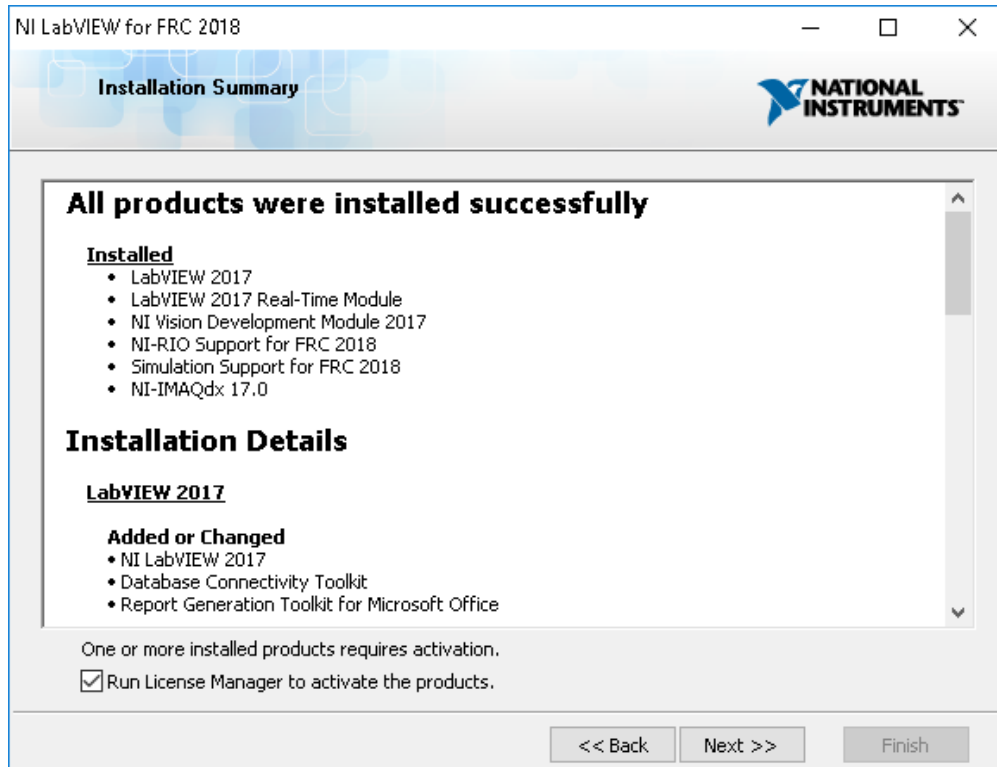
Product Information



Click "Next"

Getting Started with the Control System

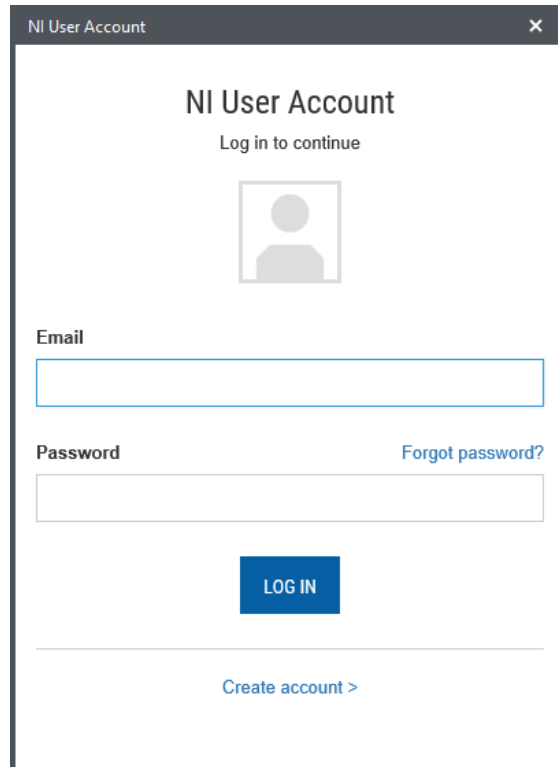
Installation Summary



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

A screenshot of the 'NI User Account' login window. The window has a dark title bar with the text 'NI User Account' and a close button. The main content area is white and contains the following elements: the title 'NI User Account', the instruction 'Log in to continue', a placeholder for a user profile picture, an 'Email' label above a text input field, a 'Password' label above a text input field, a 'Forgot password?' link, a blue 'LOG IN' button, and a 'Create account >' link at the bottom.

NI User Account

Log in to continue



Email

Password [Forgot password?](#)

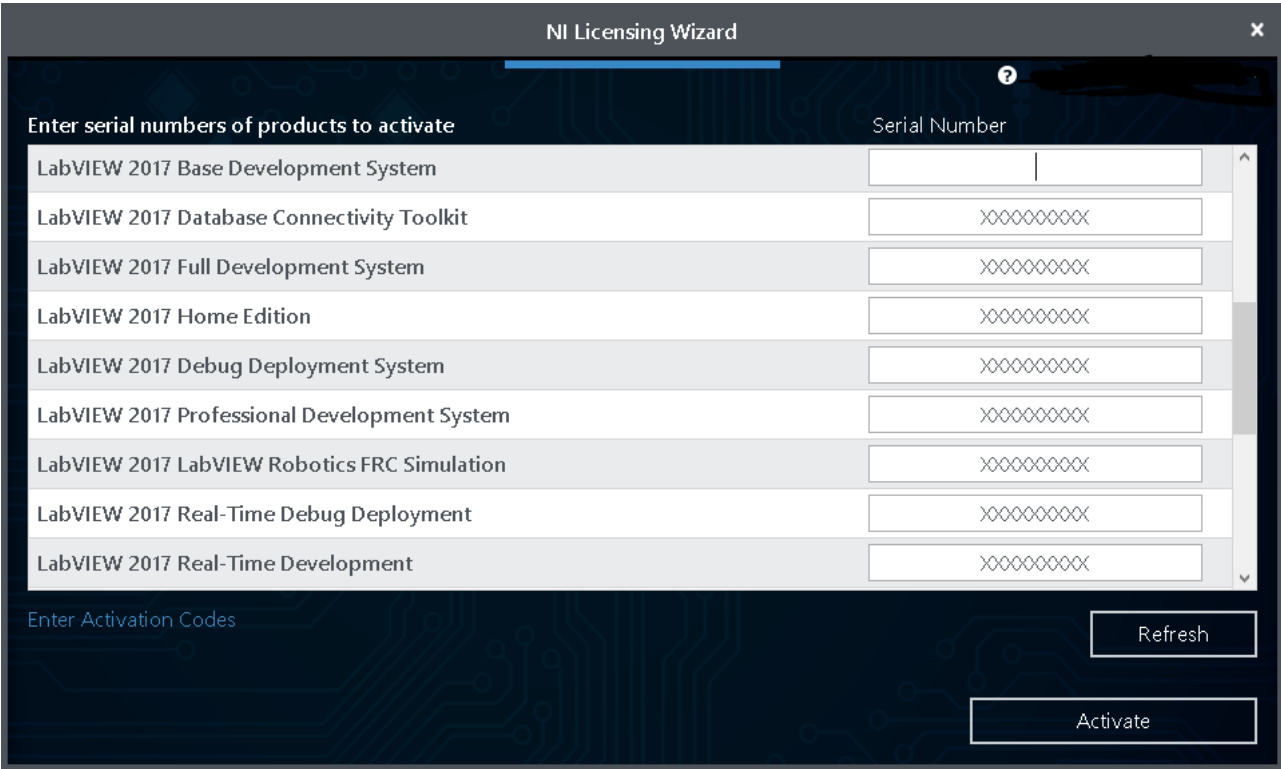
[LOG IN](#)

[Create account >](#)

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

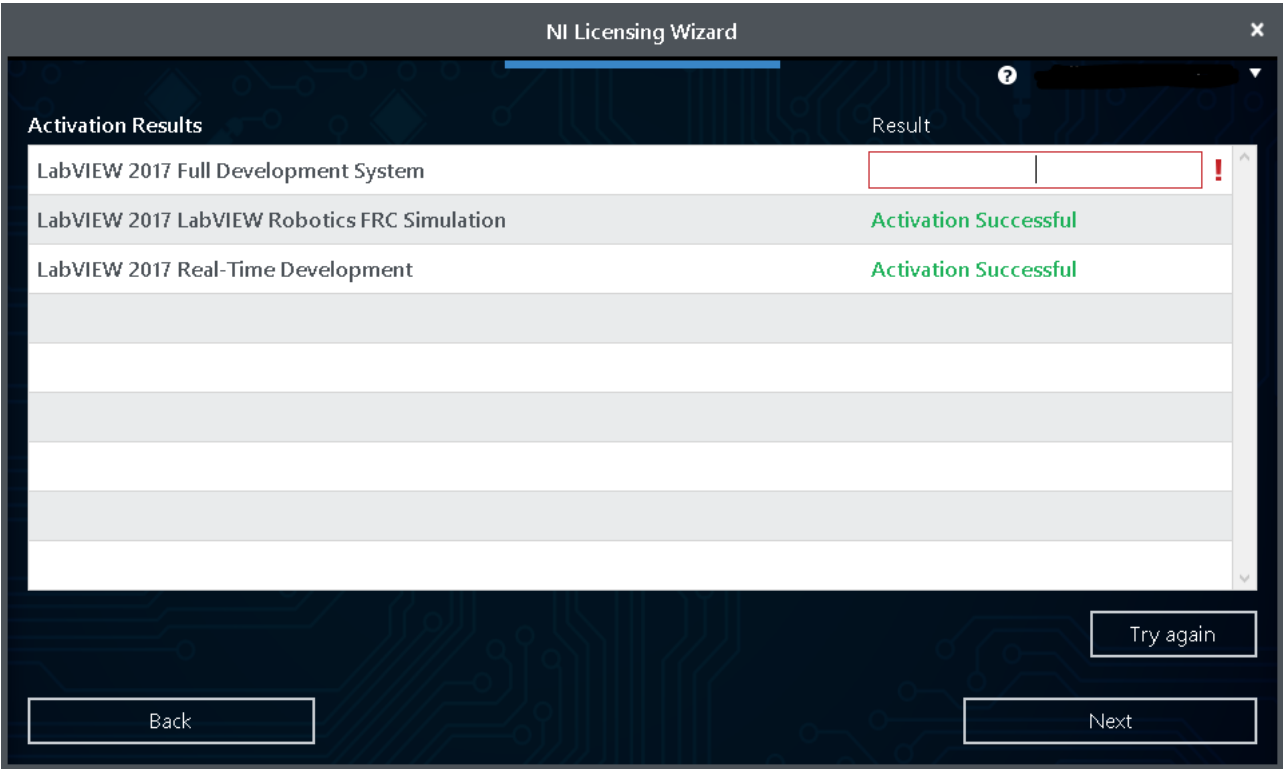
NI Activation Wizard (2)



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".

Getting Started with the Control System

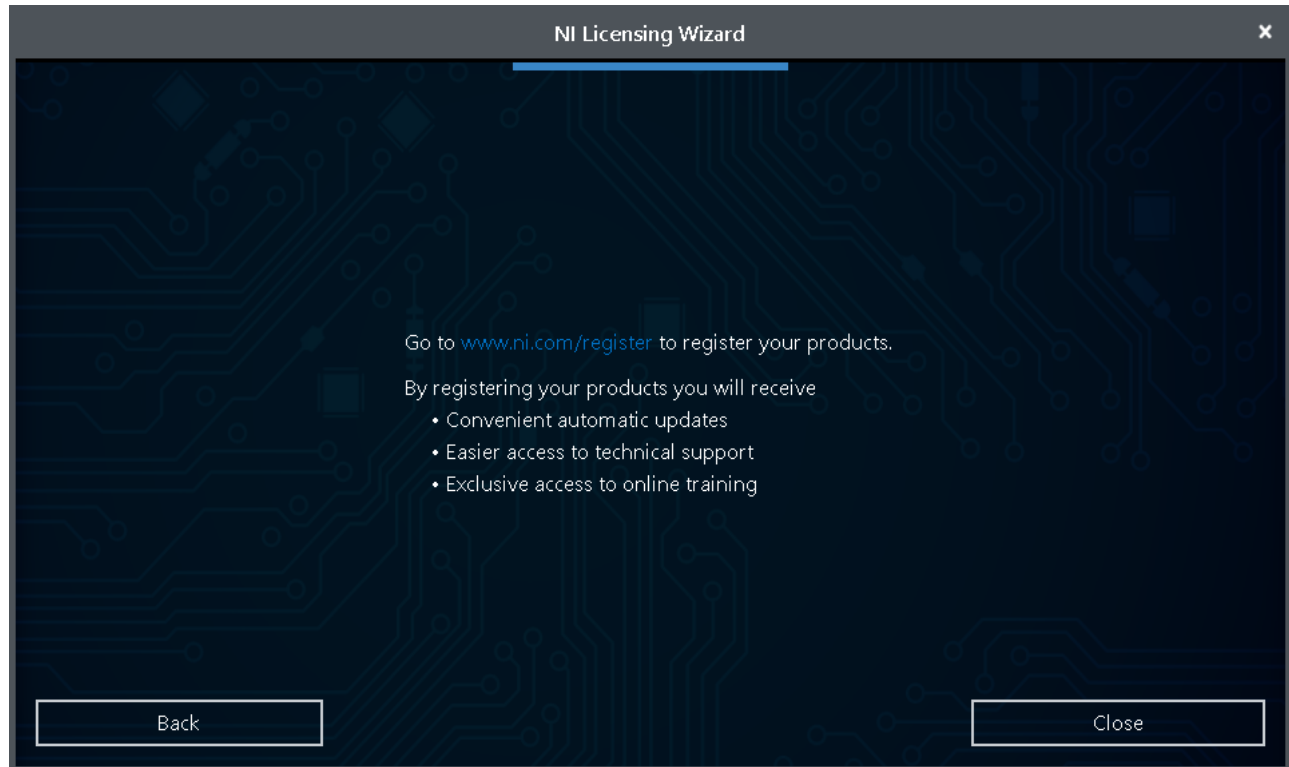
NI Activation Wizard (3)



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”.

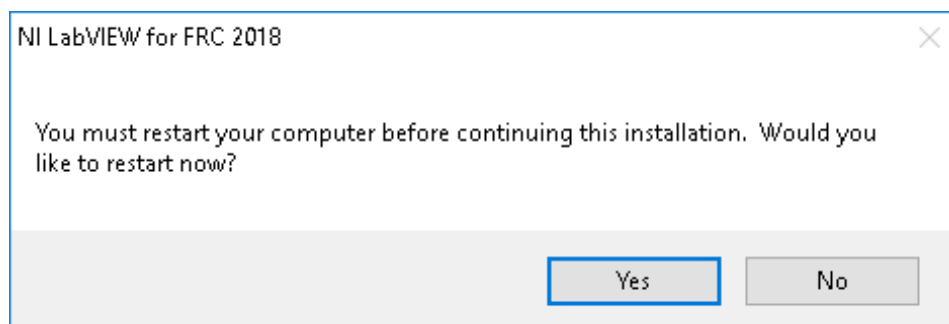
Getting Started with the Control System

NI Activation Wizard (4)



Click "Close".

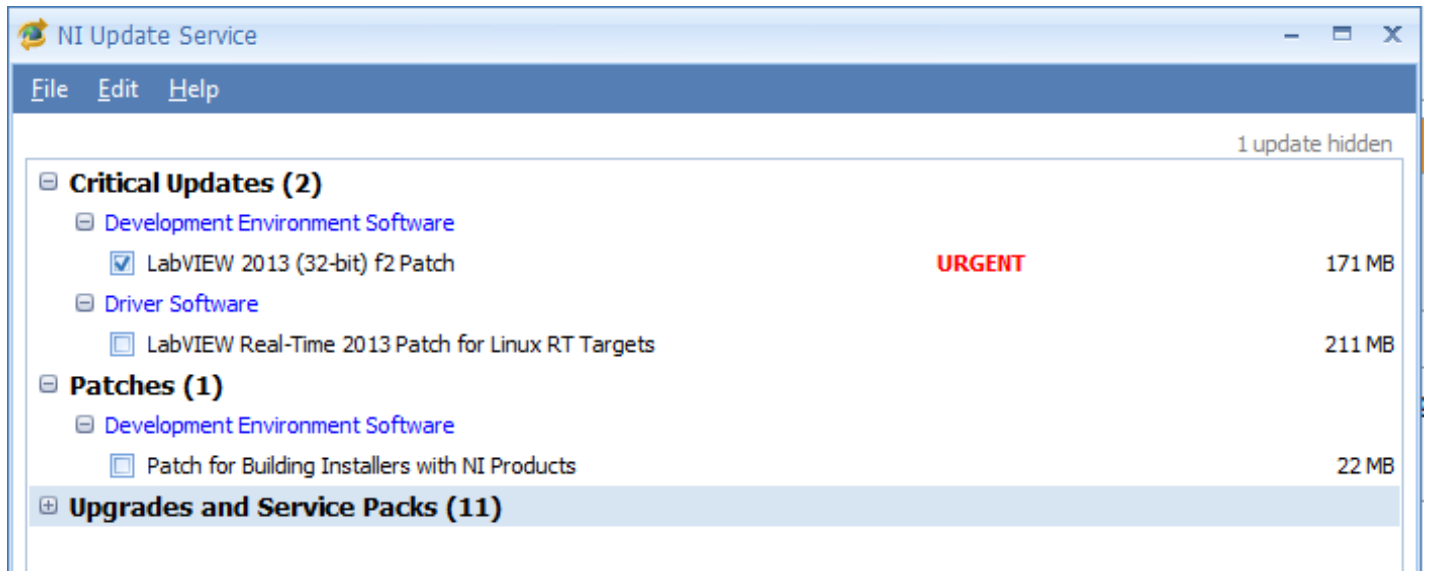
Restart Message



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 Eclipse (C++/Java)

The current suite of text-based languages, Java and C++, utilize the current version of Eclipse as a development environment. The FRC specific tools for the chosen language are installed as Eclipse plugins. You can install both the Java and C++ development tools into the same installation of Eclipse to allow programs to be written with either language using a common set of tools and user interface.

The Eclipse plugins have been tested with Eclipse Luna, Eclipse Mars, Eclipse Neon, and Eclipse Oxygen. Teams with existing installs from 2017 can update their installations to 2018 by following the updating the plugins when prompted by opening Eclipse (if automatic update is enabled) or following the Updating the plugins manually instructions below. C++ teams should also install the new toolchains ([Installing the C++ Toolchains](#)).

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

Note: The C++ and Java tools and environment are available for Windows, Mac OSX and Linux, though the Windows version is the one that has been the most heavily tested. You should be able to use any of the three for your development platform, however you should keep in mind that you will need a Windows computer to run the Driver Station software and roboRIO Imaging tool.

⚠ Warning: Java 9 is not currently supported by the FRC tools. Java 9 introduces many breaking changes, and is not provided for 32-bit systems which we need to support. Java 9 will not be supported for 2018.

Getting Started with the Control System

Getting Java

Java SE Development Kit 8u151

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.9 MB	jdk-8u151-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.85 MB	jdk-8u151-linux-arm64-vfp-hflt.tar.gz
Linux x86	168.95 MB	jdk-8u151-linux-i586.rpm
Linux x86	183.73 MB	jdk-8u151-linux-i586.tar.gz
Linux x64	166.1 MB	jdk-8u151-linux-x64.rpm
Linux x64		
macOS		
Solaris SPARC 64-bit	140.00 MB	jdk-8u151-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.32 MB	jdk-8u151-solaris-sparcv9.tar.gz
Solaris x64	140.65 MB	jdk-8u151-solaris-x64.tar.Z
Solaris x64	97 MB	jdk-8u151-solaris-x64.tar.gz
Windows x86	198.04 MB	jdk-8u151-windows-i586.exe
Windows x64	205.95 MB	jdk-8u151-windows-x64.exe

x86/64 should match Eclipse version

To use Eclipse you must have a Java 8 JDK installed on your system. You can get Java from the web site: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>.

Do not select Java 9!

Scroll down the page to "Java SE Development Kit". Accept the license agreement and download the Java SDK for your platform. The version (either x86 or x64) should match the version of Eclipse that you have installed or plan to install on your computer. This has been tested with Java SE 8u151 but will probably work with later versions as well.

Java 8 is installed on the RoboRIO and to take advantage of all the features it offers, it is suggested that you use Java 8 on your development system. You may use an earlier version, however it should be noted that the rioLog Eclipse plugin which is used to view console output requires Java 8, and other WPILib tools (dashboards, etc.) may as well.

Note: Java is required to be installed even if you are doing C++ development since Eclipse, the development environment, is a Java program. Also, the Oracle web page might change over time, so the images shown here might not exactly match what you see.

Getting Started with the Control System

Installing the C++ Toolchains (C++ teams only)



! If you have toolchains installed from a prior season, you should uninstall them using Add/Remove and install the new toolchains

Download the appropriate C++ Toolchains installer for your platform from <http://first.wpi.edu/FRC/roborio/toolchains/>

Windows: Double click on the downloaded file to launch it. If you receive a Security Warning, click Run. Check the box to accept the License Agreement, then click **Install**. When the install completes, click Finish.

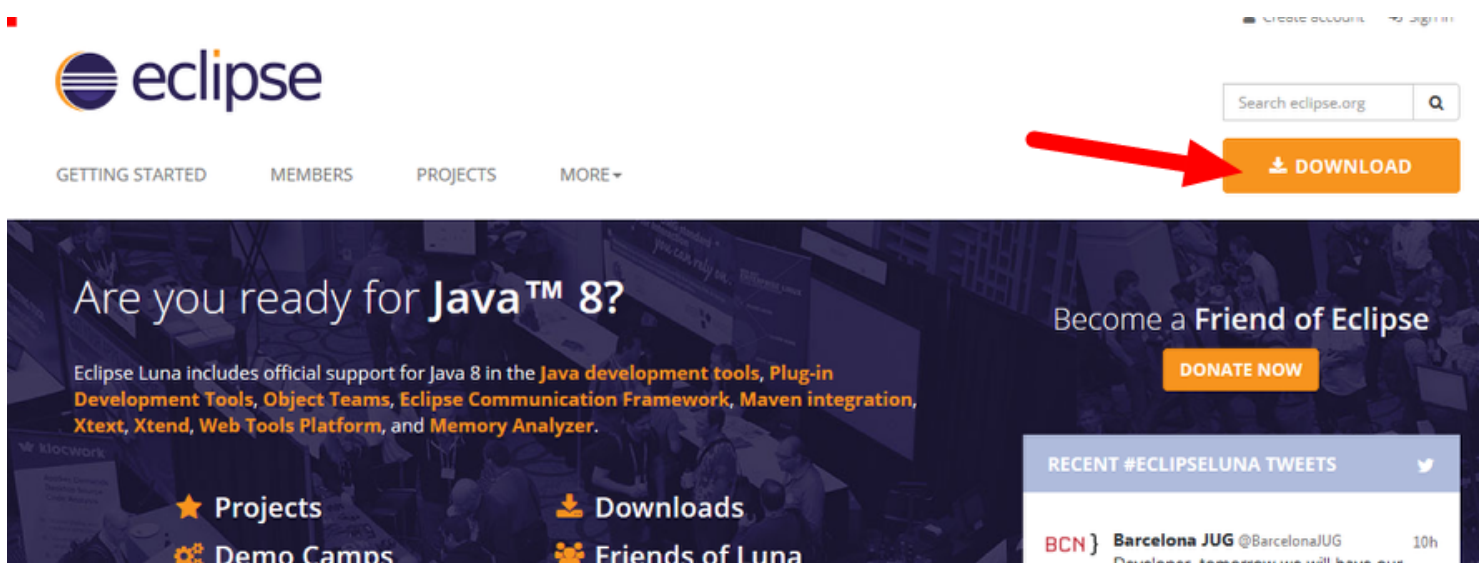
Mac OSX: Double-click on the downloaded file in Finder to unzip it. In Finder, right-click on the "FRC ARM Toolchain.pkg" file, then press the option key on your keyboard, and click "Open". Follow the steps to install the package.

Getting Started with the Control System

Linux: See the instructions in the text file on the toolchains page.

! The Windows toolchains will always install to the root of the system drive.

Getting Eclipse



Getting Started with the Control System

Download Eclipse



The screenshot shows the Eclipse IDE download page. It features two main sections: 'Eclipse IDE for Java Developers' and 'Eclipse IDE for C/C++ Developers'. Each section includes a download icon, a box for 'Windows' with '32 bit' and '64 bit' options, and a red box with the text '32/64 should match Java version'. Arrows point from this red box to the 32 bit and 64 bit options in both sections.

Eclipse IDE for Java Developers
166 MB | 918,911 DOWNLOADS
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven integration and WindowBuilder...

Eclipse IDE for C/C++ Developers
176 MB | 377,192 DOWNLOADS
An IDE for C/C++ developers with Mylyn integration.

You can get eclipse from the web site: <https://www.eclipse.org/downloads/eclipse-packages/>

Select the version of eclipse that matches your desired programming language (there are instructions below for adding C++ to Java or Java to C++). You should choose the version of eclipse that matches your operating system and version of Java from above.

At the time of this writing the current version is Oxygen (4.7) and that is what we've been using for development of the tools. On the next screen choose a download site and start the download. Choose a location such as the downloads folder for the zip file.

Note: on 64 bit Linux systems it might be necessary to install 32 bit version of libc. For example, on Ubuntu Linux, the command would be:

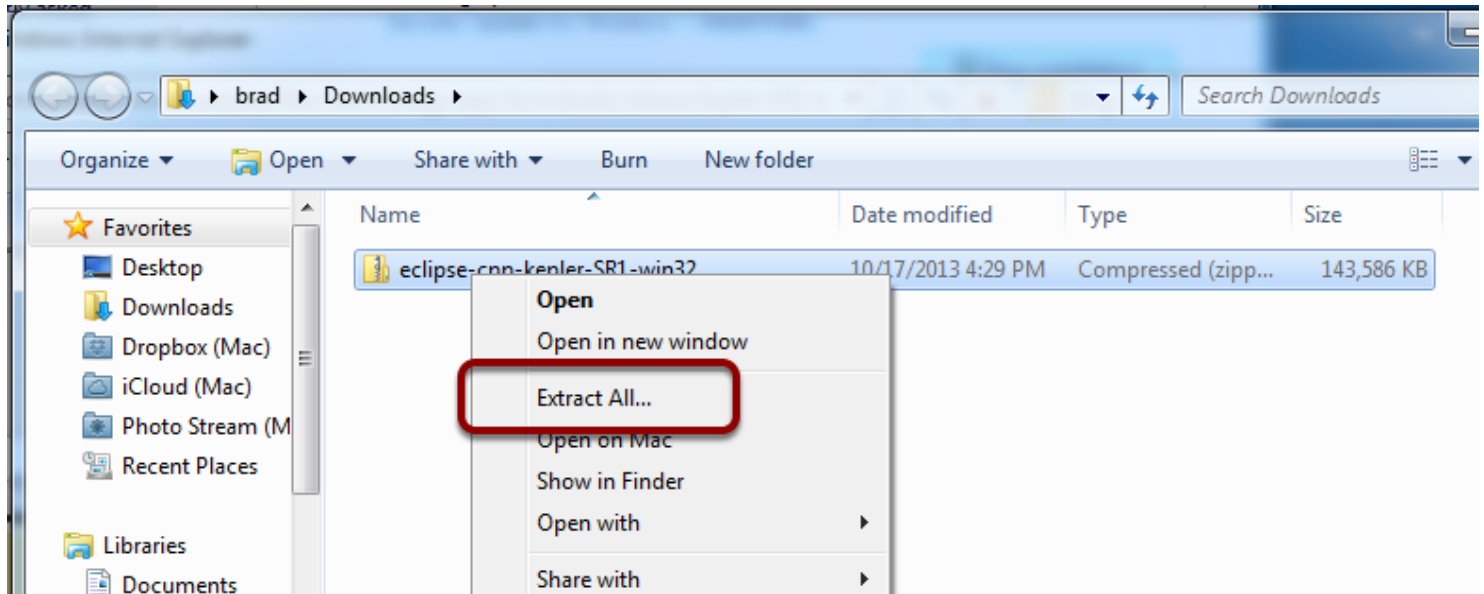
```
sudo apt-get install libc6-i386
```

This is necessary to run the gcc binaries as part of the plugins since they are compiled for 32 bit linux.

💡 Eclipse also provides .exe installers now. Other than changing the next few instructions, these should work fine as well.

Getting Started with the Control System

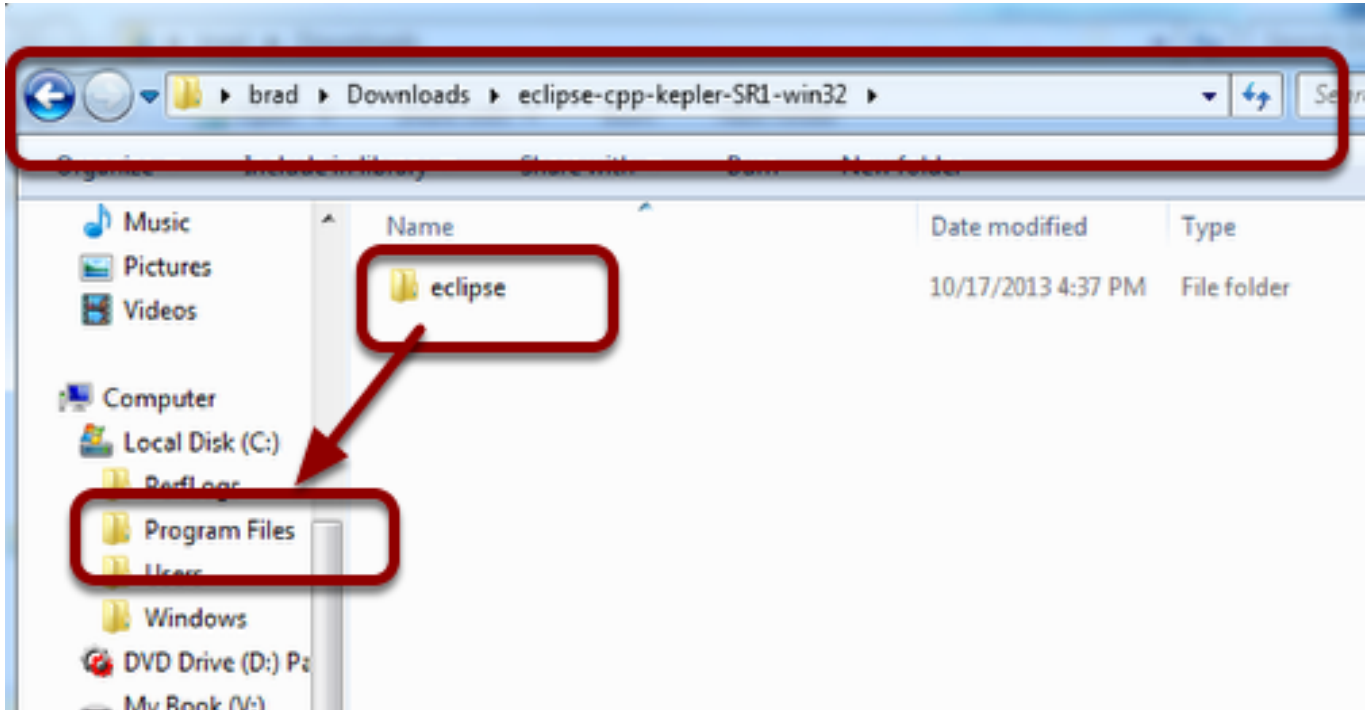
Unpack the eclipse folder and move it to Program Files



Extract the contents of the zip file by right-clicking on the .zip file in a windows explorer window and selecting "Extract All..." and taking the default for the location to extract it.

Getting Started with the Control System

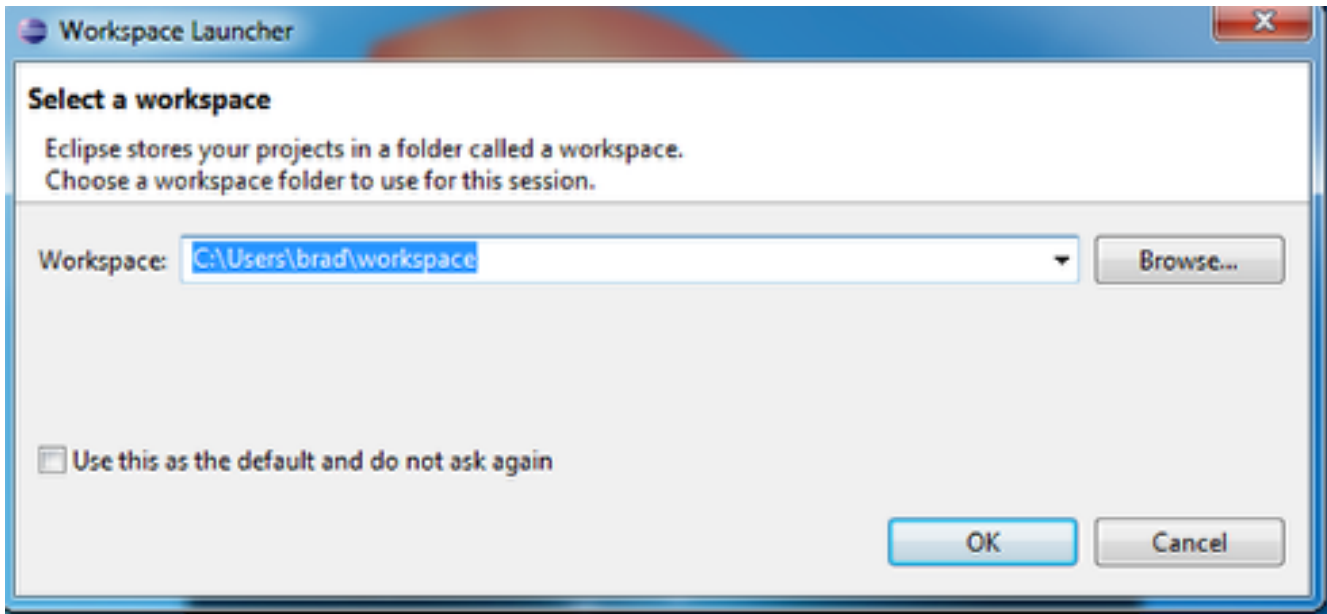
Move the extracted eclipse folder to Program Files



Move the extracted folder to Program Files or some other convenient location from which to easily run it. Within the eclipse folder you'll see the file "eclipse.exe". You can right-click on "eclipse.exe" and select "Pin to start menu" to make it easier to run eclipse without having to find the installation location.

Getting Started with the Control System

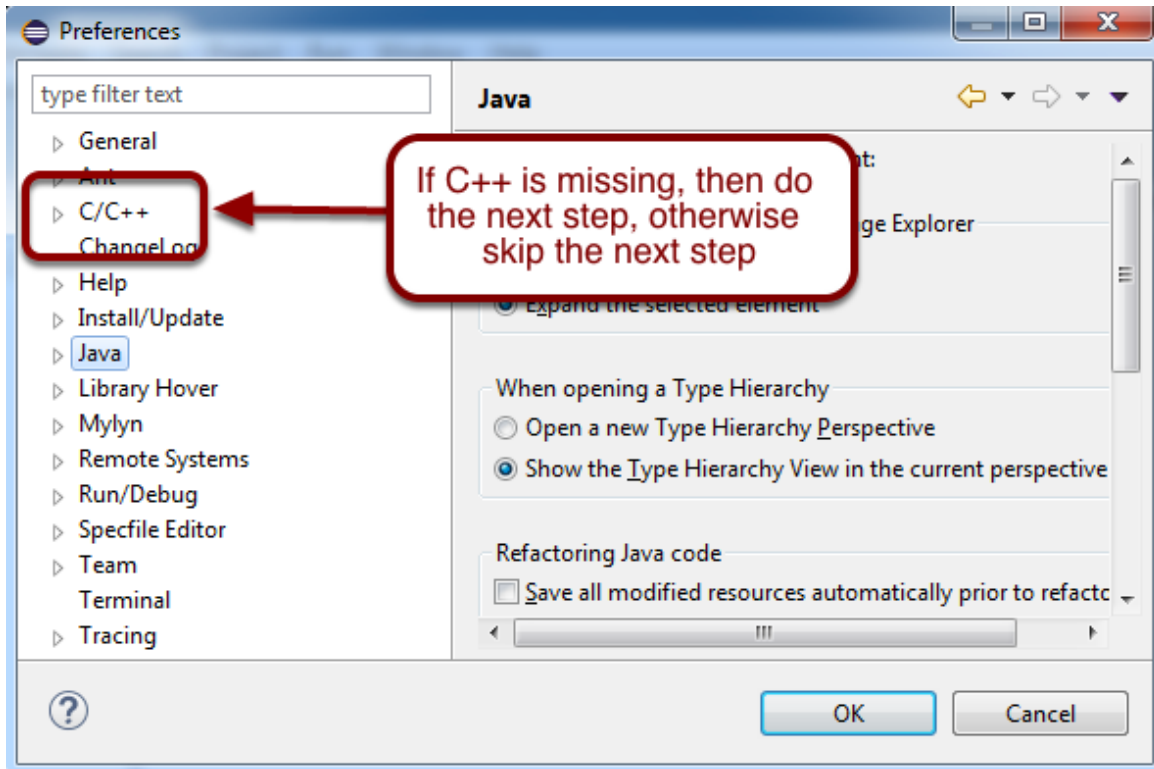
Starting Eclipse for the first time



Start Eclipse (it will be on your start menu if you chose to pin it from the previous step.) The first time Eclipse starts it will ask you for the location of your workspace. A workspace is the location on disk where projects and files are stored by default. You can have more than one workspace, but it's suggested to take the default location until you have more experience with Eclipse.

Getting Started with the Control System

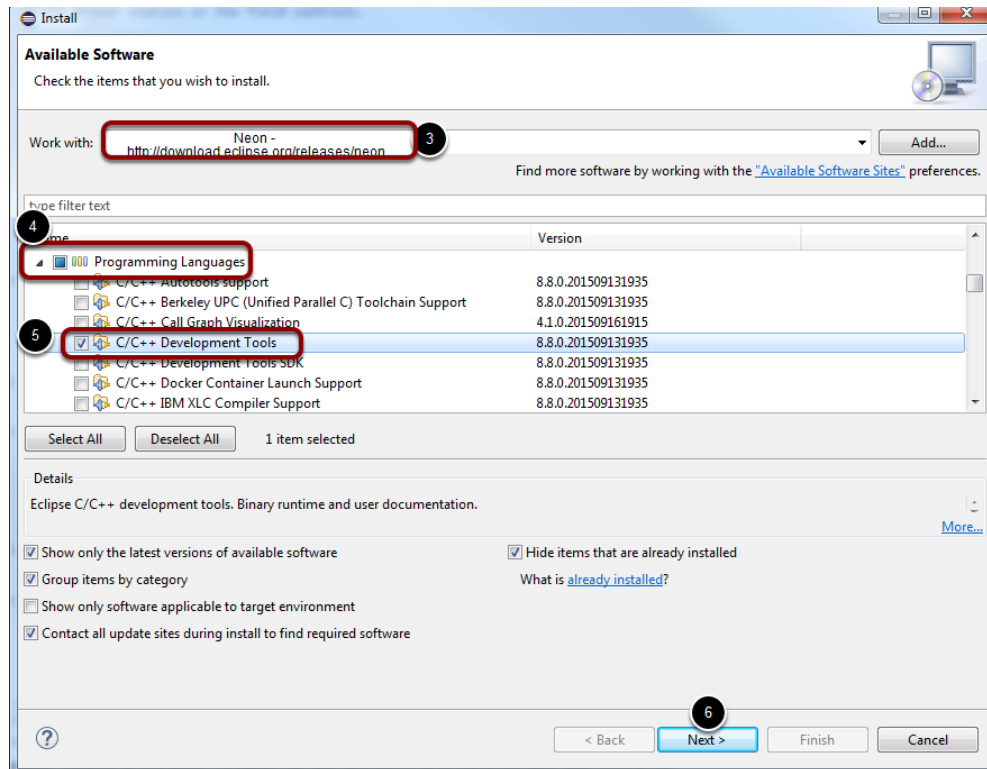
Adding C++ to Java Eclipse



To program C++ with the Java version of Eclipse, you will have to have the C++ Development Tools (CDT) installed. To determine if they're already installed, select Window, then Preferences from the menu bar. Then look for C/C++ on the left side of the Preferences window. If it is missing then you must install it. The installation procedure is in the next step.

Getting Started with the Control System

Install Eclipse C++ Development Tools



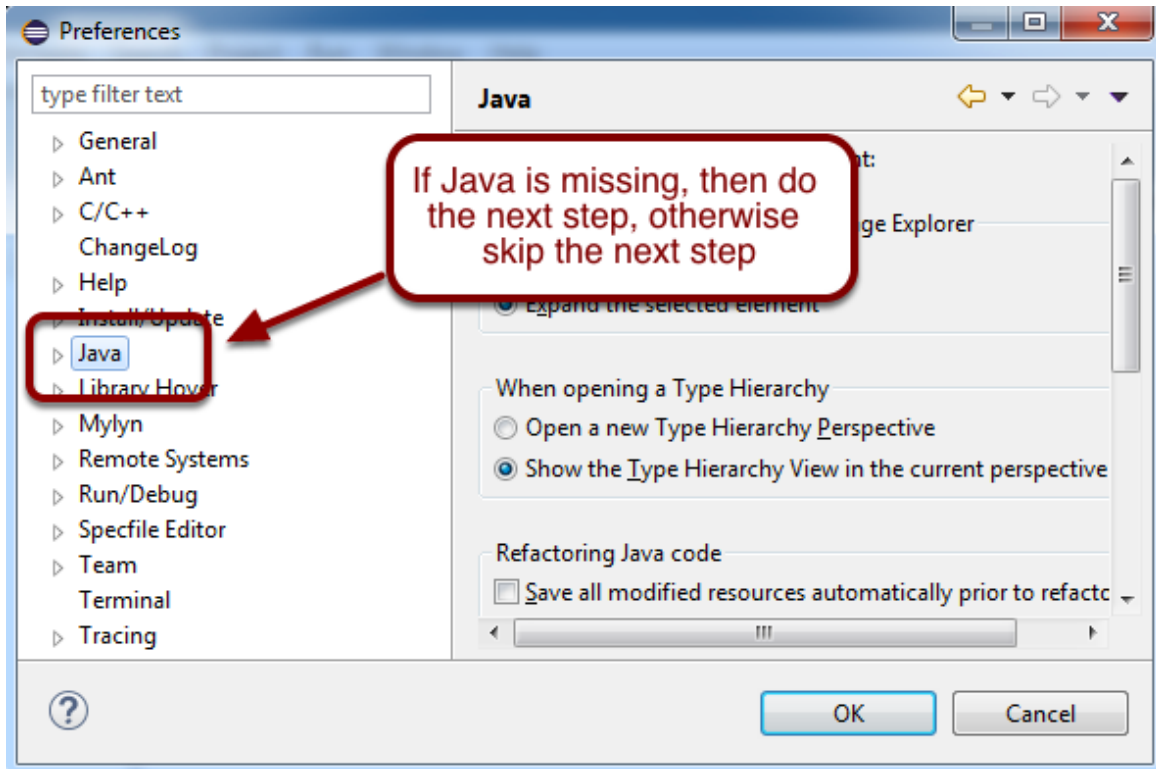
If C/C++ is missing from the preferences window (see previous step), then it must be installed

1. Close the Preferences window if it's open.
2. Select Help, then Install New Software... from the menu bar.
3. Click the dropdown and select the "Neon" site as shown.
4. Scroll down to the Programming Languages section and click the arrow to expand.
5. Click the checkbox to choose Eclipse C++ Development Tools
6. Click Next.
7. Take the defaults for the other options and let Eclipse restart.

When these steps are finished, and Eclipse has restarted, C++ should be an available option on the Preferences window, and all the C++ perspectives will be available.

Getting Started with the Control System

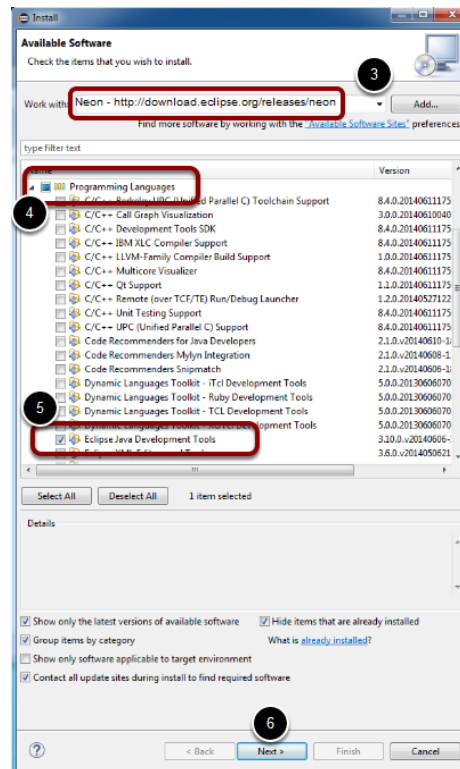
Adding Java to C++ Eclipse



To program Java with the C++ version of Eclipse, you will need to have the Java Development Tools (JDT) installed. To determine if they're already installed, select Window, then Preferences from the menu bar. Then look for Java on the left side of the Preferences window. If it is missing then you must install it. The installation procedure is in the next step. If you do have the Java development tools installed (Java is shown), then skip the next step and continue configuring to Setting up the JDK in Eclipse.

Getting Started with the Control System

Install Eclipse Java Development Tools



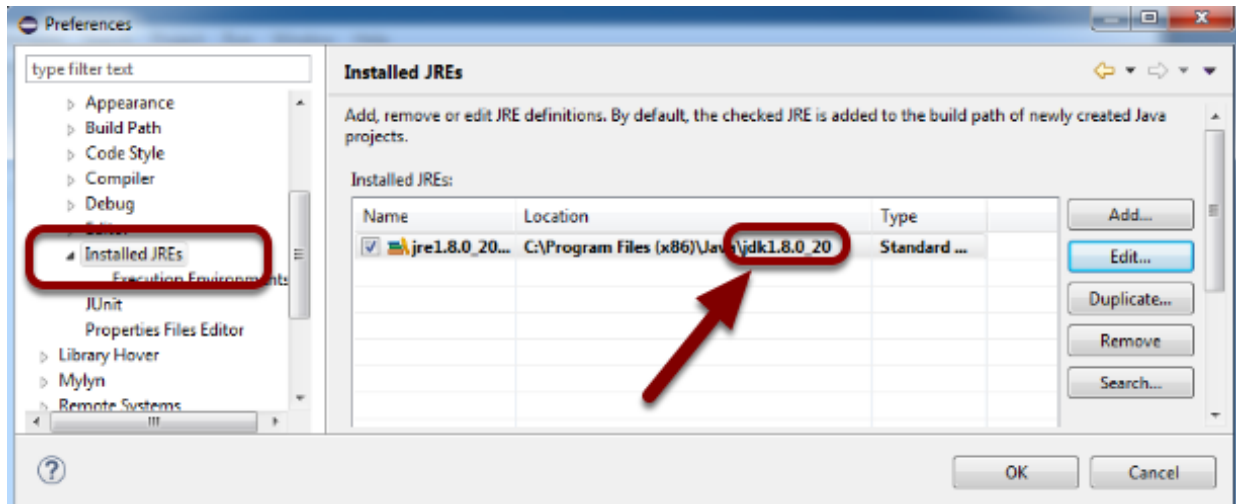
If Java is missing from the preferences window (see previous step), then it must be installed

1. Close the Preferences window if it's open.
2. Select Help, then Install New Software... from the menu bar.
3. Click the dropdown and select the "Neon" site as shown.
4. Scroll down to the Programming Languages section and click the arrow to expand.
5. Choose Eclipse Java Development Tools
6. Click Next.
7. Take the defaults for the other options and let Eclipse restart.

When these steps are finished, and Eclipse has restarted, Java should be an available option on the Preferences window, and all the Java perspectives will be available.

Getting Started with the Control System

Setting up the JDK in eclipse (java teams only)

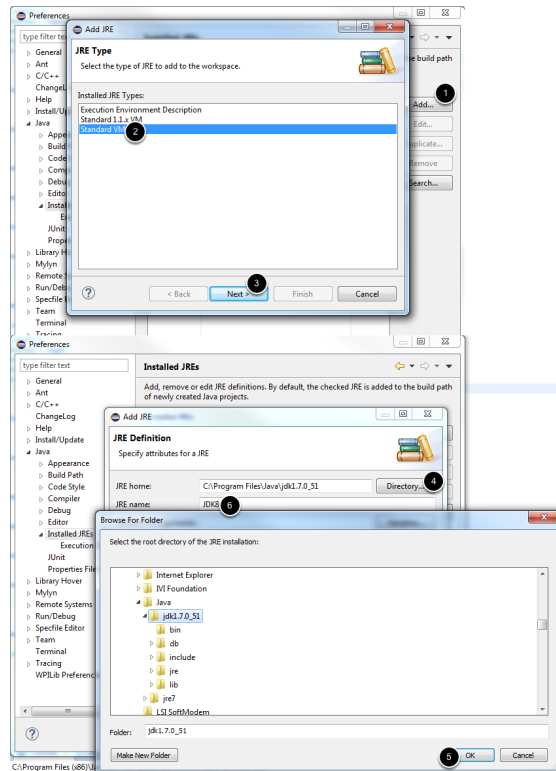


1. Select Windows from the menu bar, then Preferences.
2. Choose Java preferences in the list on the left of the Preferences window, then Installed JREs.
3. Be sure that the installed JDK is selected as shown (make sure the "Location" field includes jdk 8 or 1.8, the name field may be the same in either location). This will enable eclipse to build Java programs for the RoboRIO. Without this setting you will see error messages about the JRE path not being set correctly.

If you have no entries, or your only entry does not say jdk in the Location, proceed with the next step. Otherwise, skip the next step.

Getting Started with the Control System

Adding the JDK

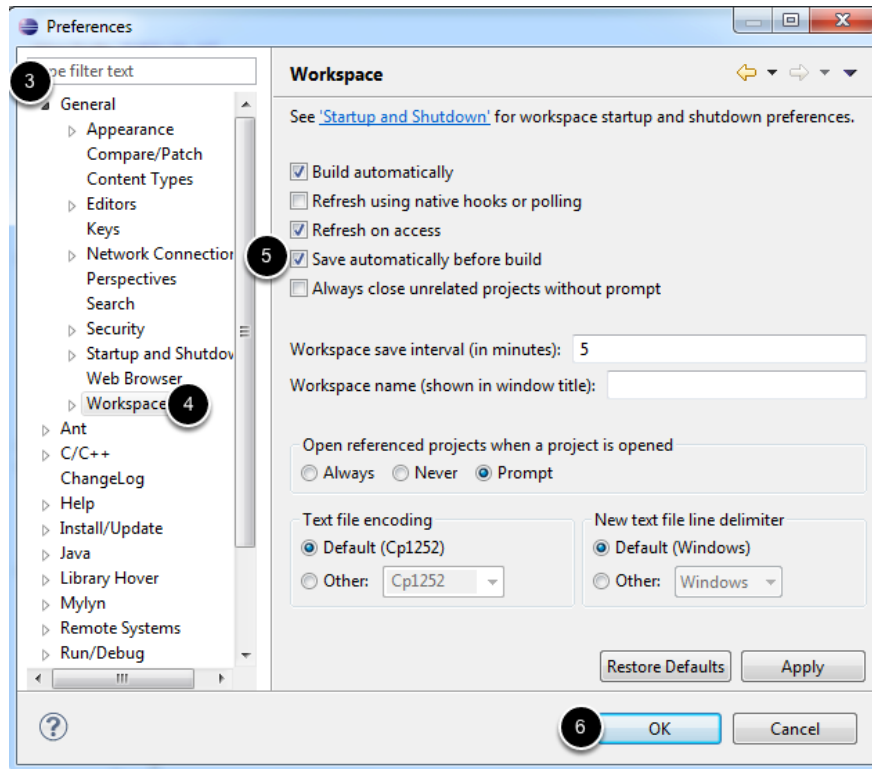


Only if the JDK is not shown in the step above:

1. Click **Add**
2. Select **Standard VM**
3. Click **Next**
4. Click **Directory** and browse to the folder for the JDK (usually C:\Program Files\Java* or C:\Program Files (x86)\Java*). The image shows jdk1.7.0_51, you will likely have a jdk1.8.* version.
5. Click **OK**. Pick a name for the JRE such as JDK8.
6. Click **Finish**
7. Make sure the box for the newly added JDK entry is checked.

Getting Started with the Control System

Configuring eclipse

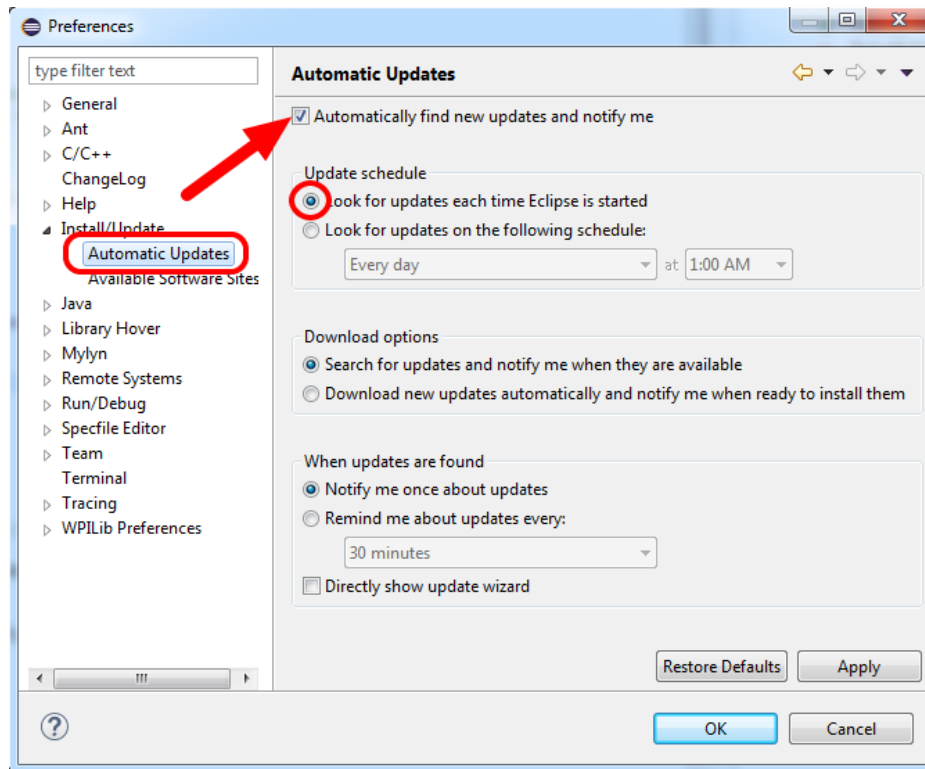


There are a huge number of configuration options for eclipse to set up the environment for your preferences. One suggested setting to note is: "Save automatically before build." This setting will cause all of your workspace changes to be saved when you build the project. If you don't set this, remember to save changes before building, otherwise the rebuilt program won't reflect your newest updates.

To set this, go to **Window -> Preferences -> General -> Workspace -> Check Save automatically before build -> OK**

Getting Started with the Control System

Automatic Updates

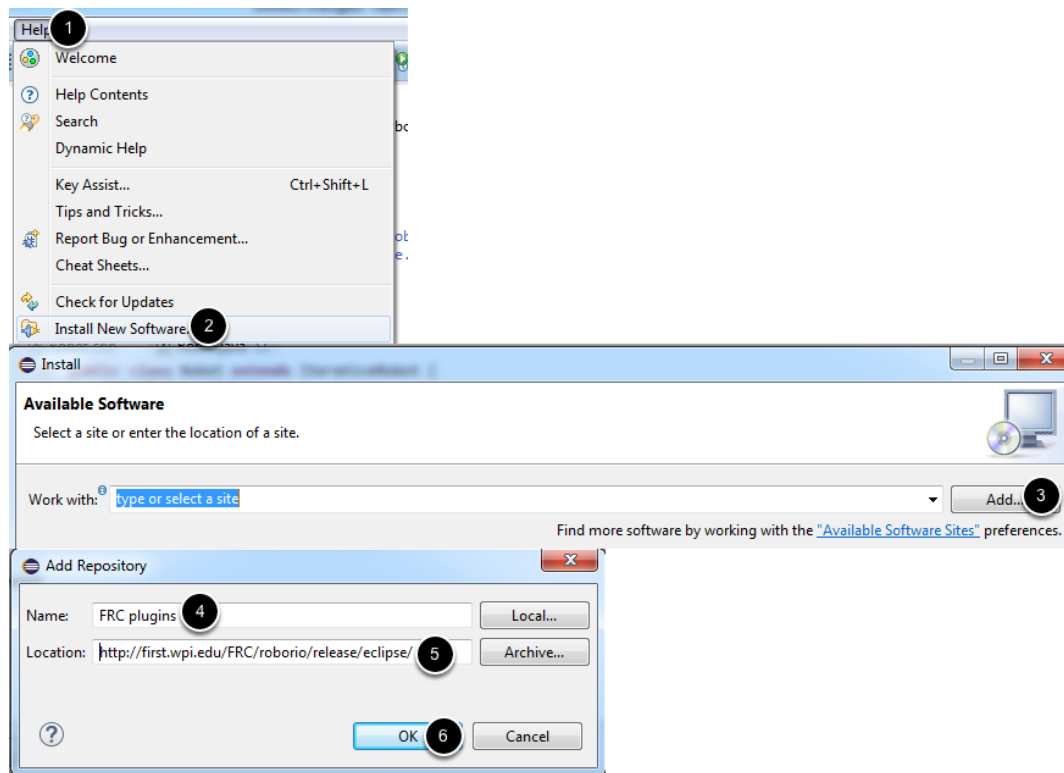


Another recommended setting is enabling Automatic Updates. With Automatic Updates enabled, Eclipse will check for updated versions of the plugins each time it starts and inform you if an update is available. This will help insure you are notified of new versions of the WPILib plugins.

To enable Automatic Updates, select **Install/Update** then **Automatic Updates**. Check the box at the top to **Automatically find new updates and notify me**. Select the radio button to **Look for updates each time Eclipse is started**. Then click **OK**.

Getting Started with the Control System

Installing the development plugins - Option 1: Online Install



It is recommended to install the plugins using this method, which requires an active internet connection and fetches the plugins directly from the WPILib site. This will allow you to check for updates to the plugins using Eclipse.

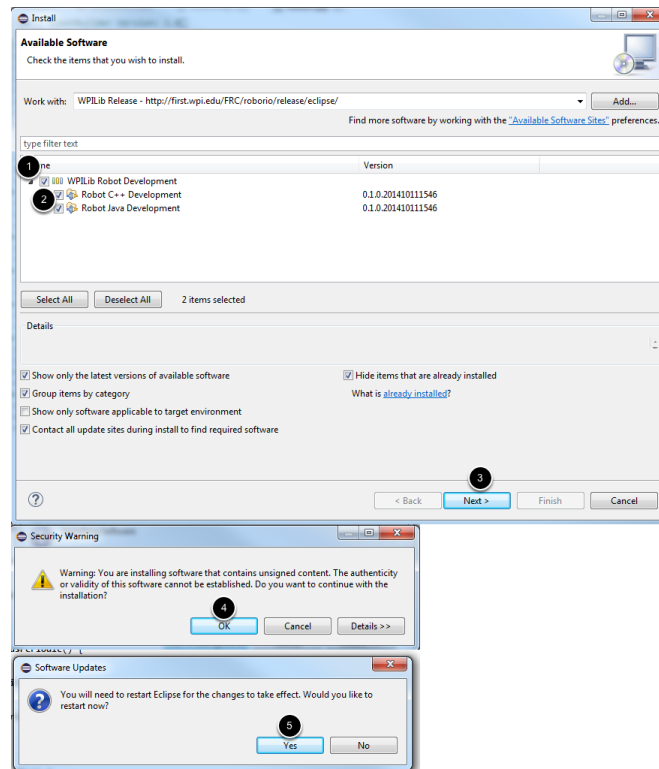
Eclipse extensions are based on user-installed plugins. To get the WPILib development tools you will need to install the correct plugin for your language.

When Eclipse starts:

1. Select "Help"
2. Click "Install new software".
3. From here you need to add a software update site, the location where the plugins will be downloaded. Push the "Add..." button then fill in the "Add Repository" dialog with:
4. Name: FRC Plugins
5. Location: <http://first.wpi.edu/FRC/roborio/release/eclipse/>
6. Click "OK".

Getting Started with the Control System

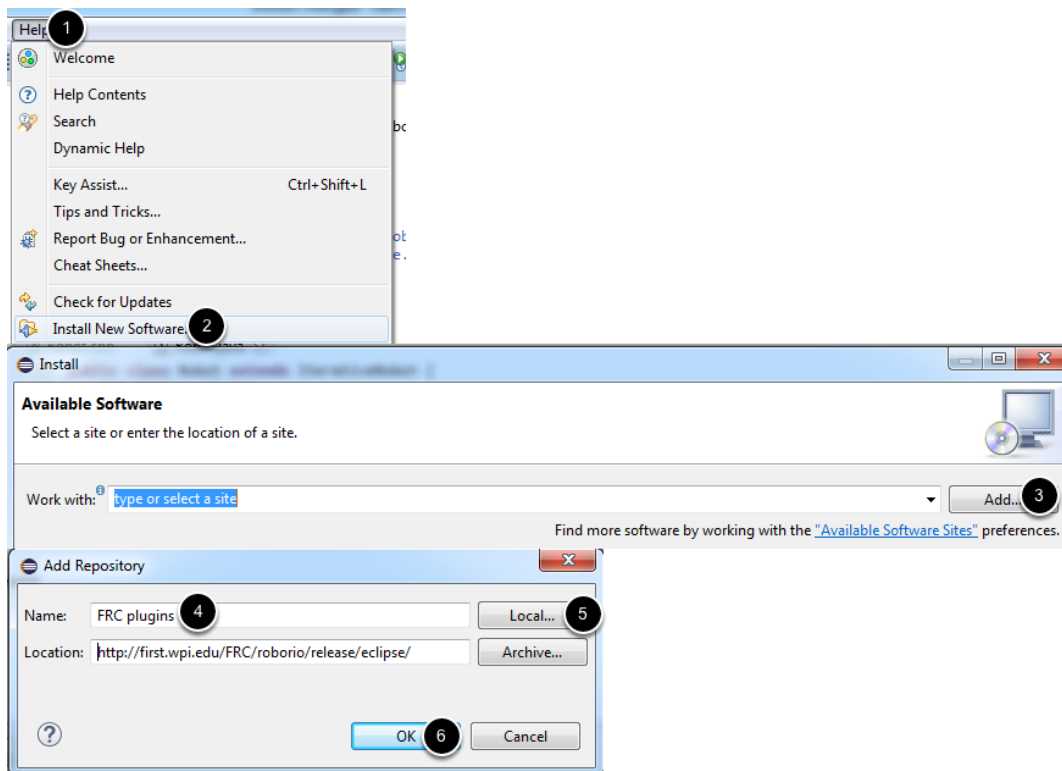
Selecting the correct plugins



1. Click the arrows if necessary to expand the WPILib Robot Development menu.
2. Select the WPILib Robot Development plugin for your desired language (you can install both if you wish to try programming in both languages)
3. Click **Next**, **Next** on the next page, then click the radio button to accept the license agreement and click **Finish**
4. If you receive a Security Warning prompt, click OK to continue.
5. When prompted, restart Eclipse. After Eclipse restarts and you select your Workspace (if prompted) you will see a dialog that says Installing Java. This details the installation progress of the plugins, wait for the install to complete before proceeding. This dialog should only appear when the plugins are first installed or updated.

Getting Started with the Control System

Installing the development plugins - Option 2: Download and install offline



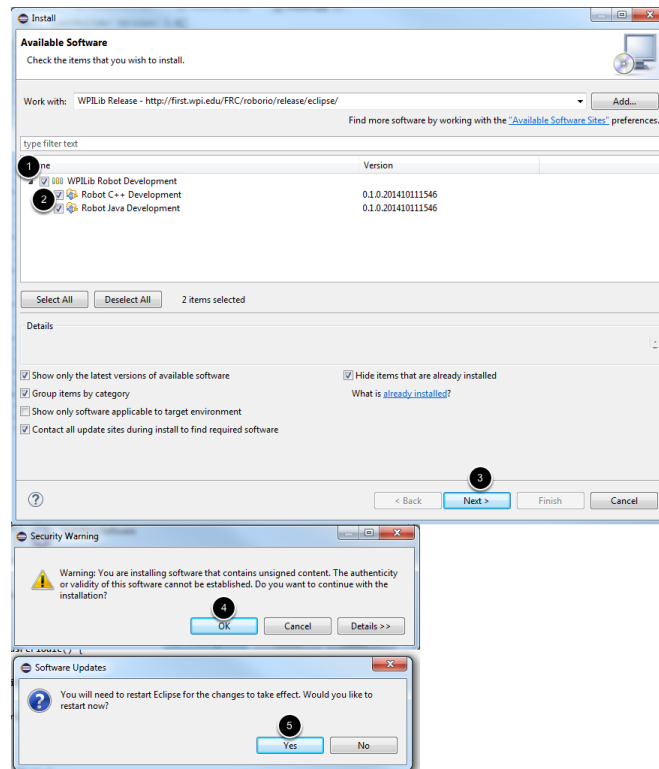
If you need to download the plugins and install them on a different machine offline, you will be unable to check for updates using Eclipse. Download the zipfile containing the plugins from <http://first.wpi.edu/FRC/roborio/release/> Right click on the zip and select Extract All to extract the files.

When Eclipse starts:

1. Select "Help"
2. Click "Install new software".
3. From here you need to add the downloaded plugin location. Push the "Add..." button then fill in the "Add Repository" dialog with:
4. Name: FRC Plugins Offline
5. Click Local
6. Browse to the "site" directory inside the directory you extracted the zip file to.
7. Click "OK".

Getting Started with the Control System

Selecting the correct plugins

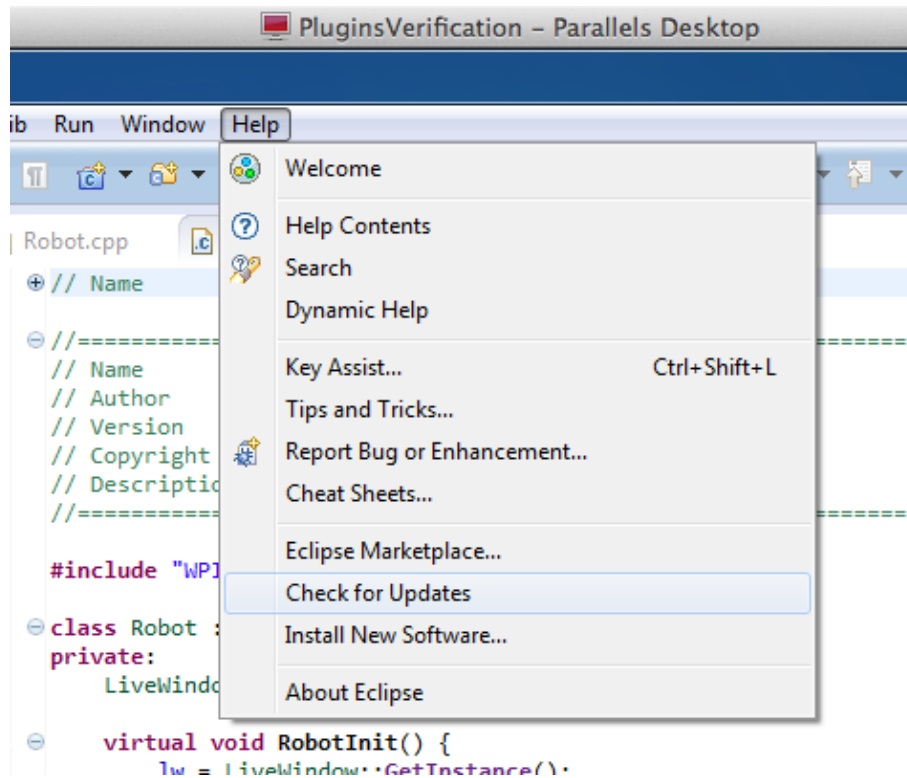


1. Click the arrows if necessary to expand the WPILib Robot Development menu.
2. Select the WPILib Robot Development plugin for your desired language (you can install both if you wish to try programming in both languages)
3. Click **Next**, **Next** on the next page, then click the radio button to accept the license agreement and click **Finish**
4. If you receive a Security Warning prompt, click OK to continue.
5. When prompted, restart Eclipse. After Eclipse restarts and you select your Workspace (if prompted) you will see a dialog that says Installing Java. This details the installation progress of the plugins, wait for the install to complete before proceeding. This dialog should only appear when the plugins are first installed or updated.

If updated plugins are released, you can either repeat this process (you will get one additional Eclipse window telling you that the components are already installed and an upgrade will be performed instead of an install), or if online installation is an option, you can complete the online installation steps above, then get future updates using the Eclipse Automatic Updates (or the manual update check described below)

Getting Started with the Control System

Updating the plugins manually



Note: This only works if the plugins were installed using Option 1 - Online Install from above. For updating plugins when the offline install was used, see the note at the end of the step above.

If you choose not to enable Automatic Updates as described above, you will need to manually have Eclipse check for updates to install new versions of the plugins.

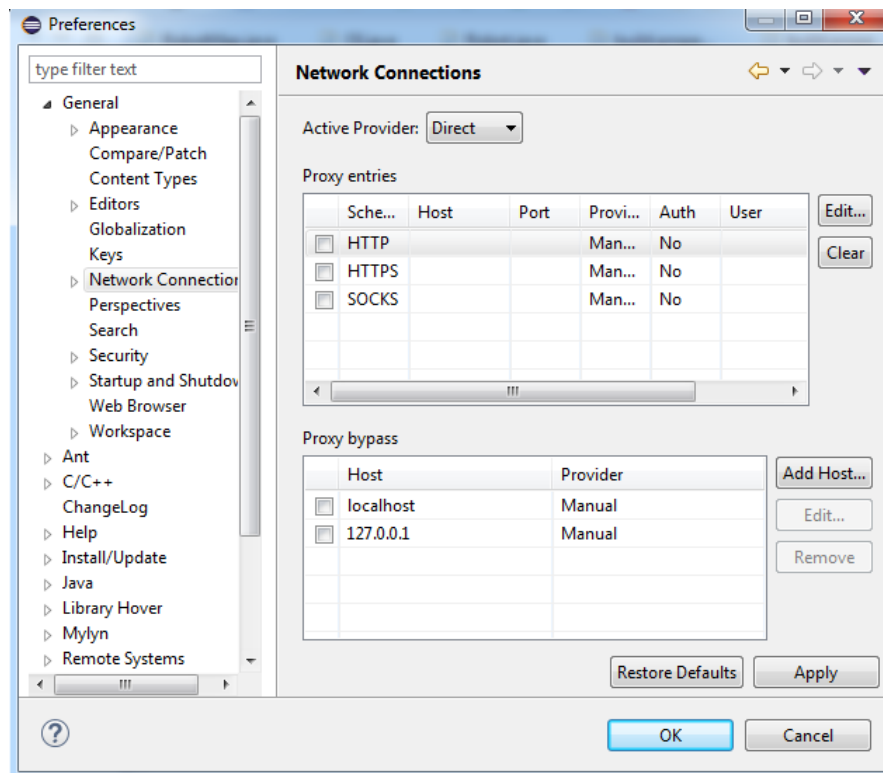
Select **Help** from the menu bar, followed by **Check for updates**. Eclipse will check if there is a newer version available of any installed plugin and inform you if an update is found. Updating the plugins will ensure that your development system is up to date with the latest version of the development tools.

Troubleshooting

Below are some troubleshooting steps for commonly encountered issues

Getting Started with the Control System

Unable to read repository at <http://first.wpi.edu/FRC/roborio/release/eclipse/content.xml>



This error occurs if Eclipse cannot contact the server to download the plugins. There are a couple possible causes of this issue:

1. Your computer is not connected to the Internet. Verify your network connection and try again.
2. Your firewall is blocking Eclipse. Try adding an exception for Eclipse or disabling your Firewall.
3. Your proxy settings were read improperly by Eclipse. In Eclipse Select **Window->Preferences->General->Network Connections**. If you don't use a proxy or don't know, set the Active Provider to **Direct**. If you use a proxy set the Active Provider to **Manual** and configure the proxy information by selecting the protocol and clicking Edit.

Getting Started with the Control System

Need Java 1.7 or newer

If you get an error message when attempting to run Eclipse that says you "need Java 1.7 or newer", you have mismatched versions of Java and Eclipse installed. The easiest fix is to go back and download the other version of Eclipse (32 bit if you had 64, 64 if you had 32).

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 DVD are required for running either the Driver Station or Utilities.

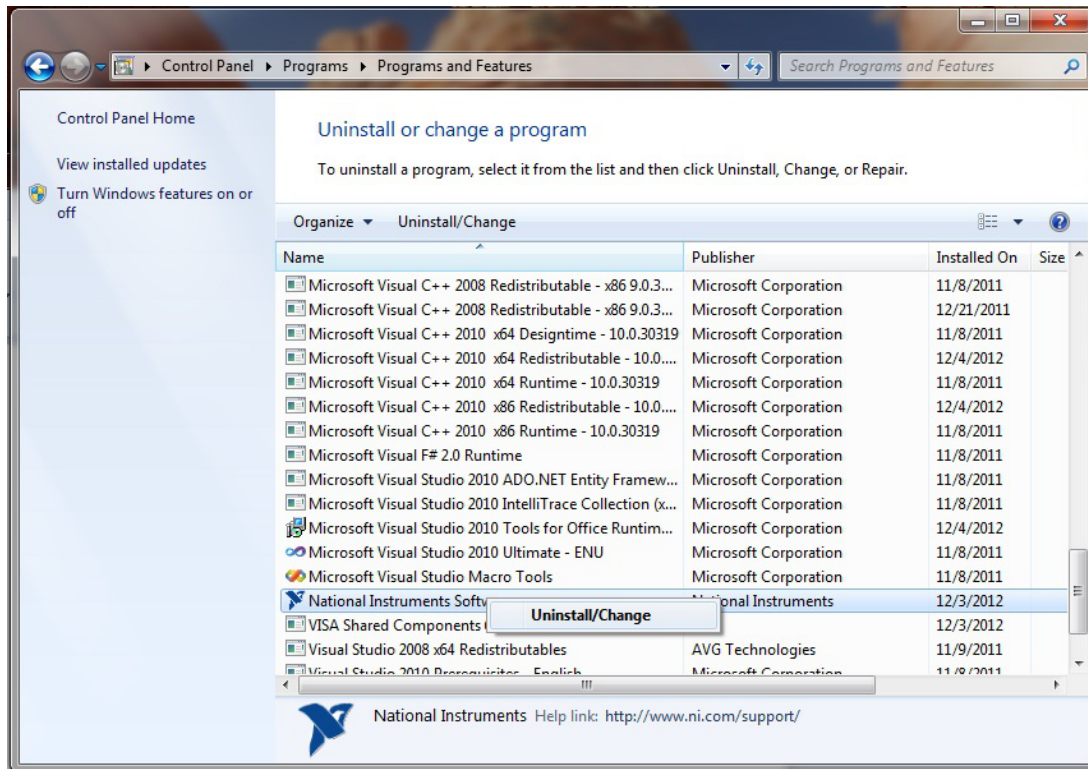
C++ and Java teams wishing to use NI Vision Assistant should run the full DVD installer as described in the article - [Installing LabVIEW for FRC \(LabVIEW only\)](#)

⚠ 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.

⚠ LabVIEW Teams: 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

Getting Started with the Control System

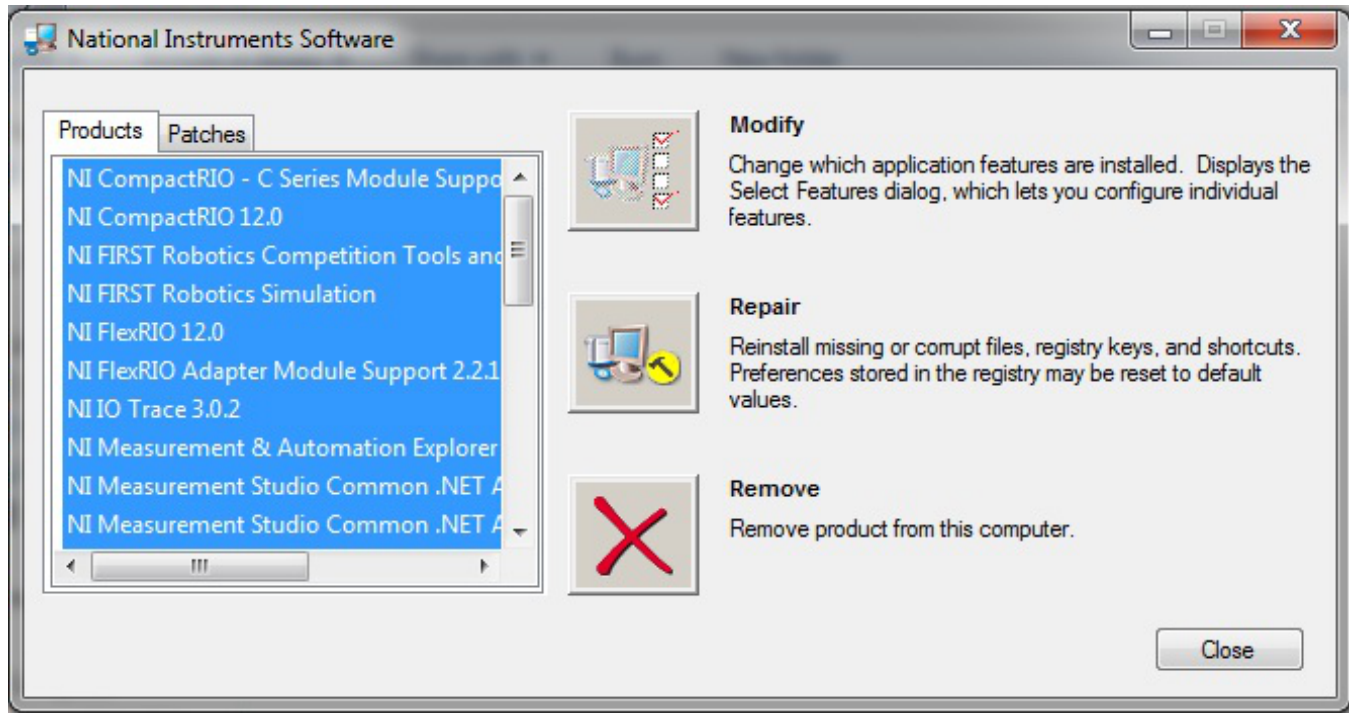
Uninstall Old Versions (Recommended)



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 2018 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.

Downloading the Update

Download the update from <http://www.ni.com/download/first-robotics-software-2017/7183/en/>



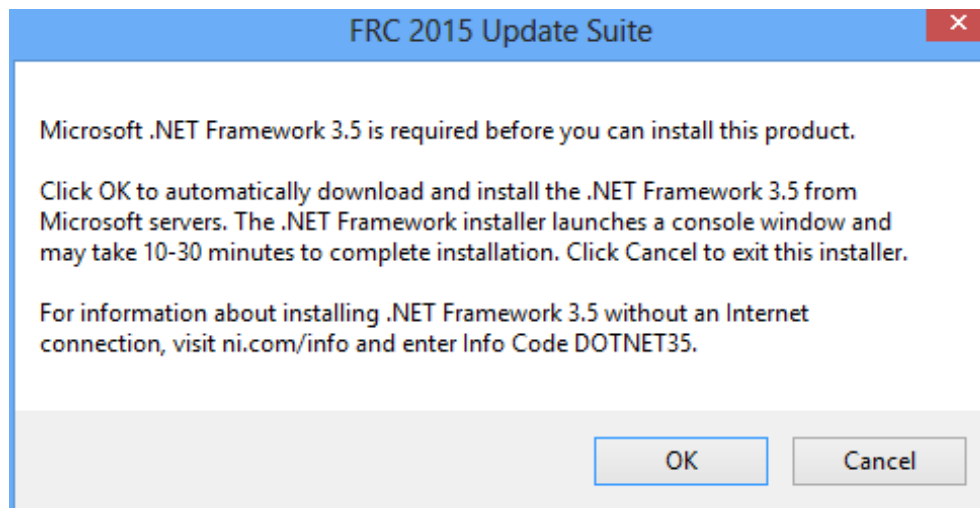
Note: This download will require the decryption key from the Kickoff broadcast

Getting Started with the Control System

.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.

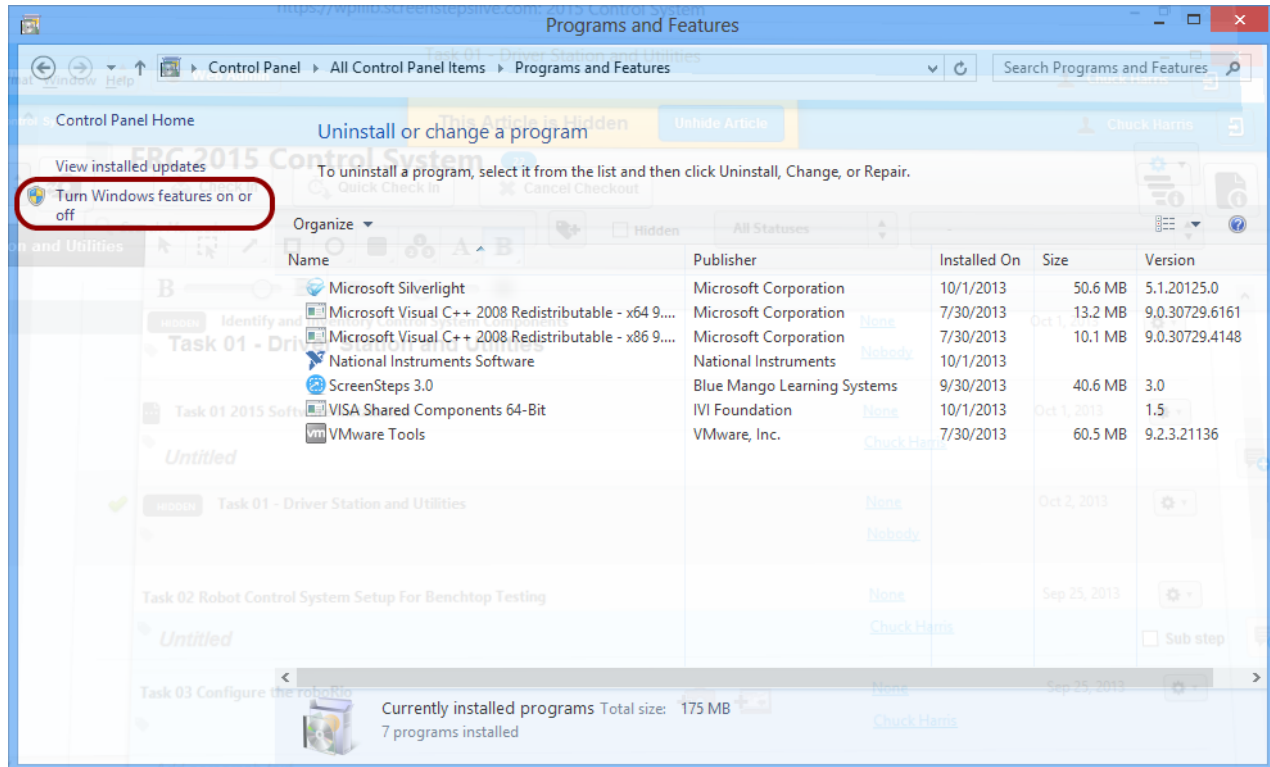
.NET Framework 3.5



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

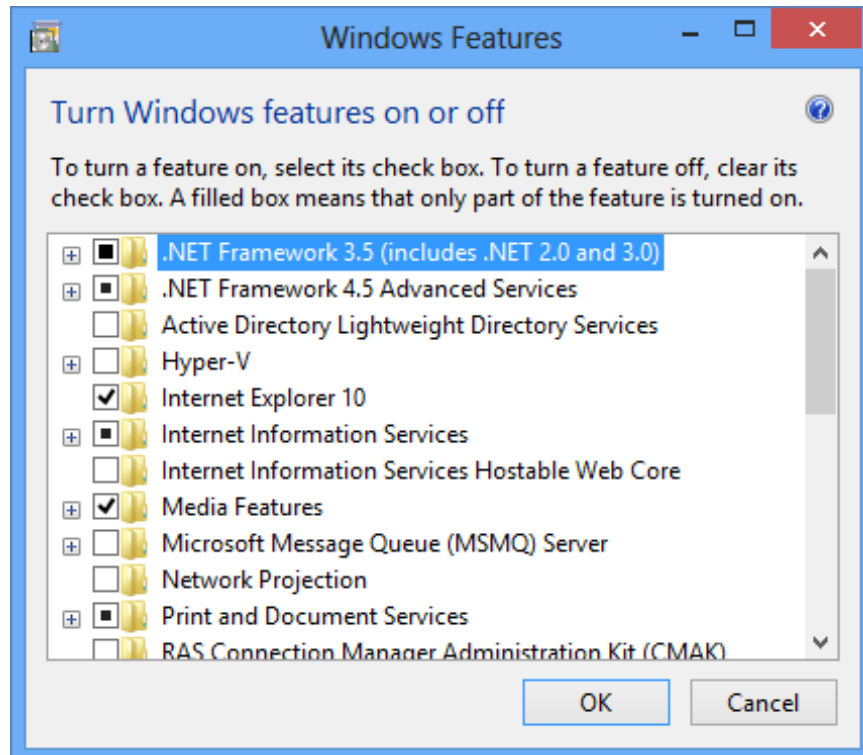
Programs and Features



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

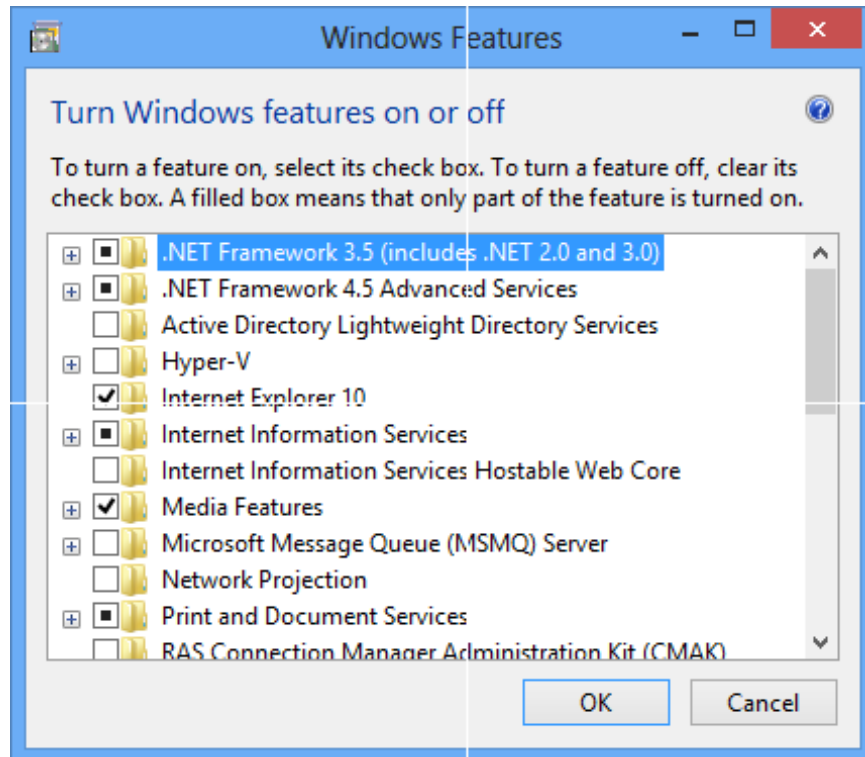
Windows Features (.NET Framework 3.5 not on)



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 2016 Update Suite](#).

Getting Started with the Control System

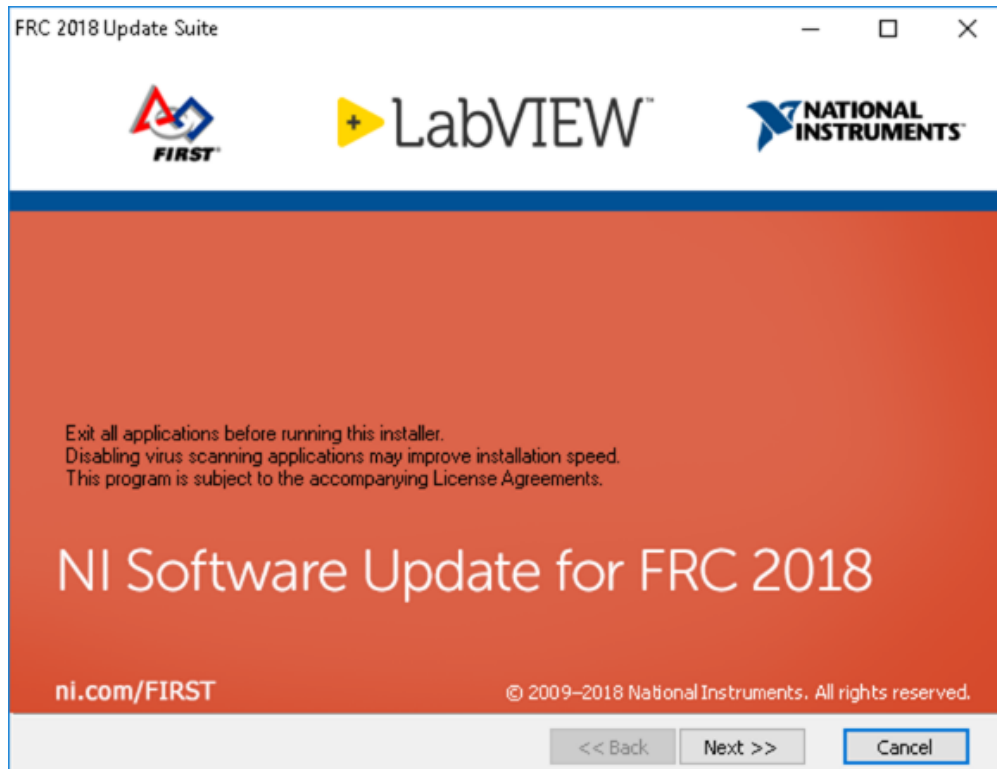
Windows Features (.NET Framework 3.5 already on)



If a black dot is shown next to ".NET Framework 3.5" the feature is already on. Click "Cancel" and [restart installation of FRC 2017 Update Suite](#).

Getting Started with the Control System

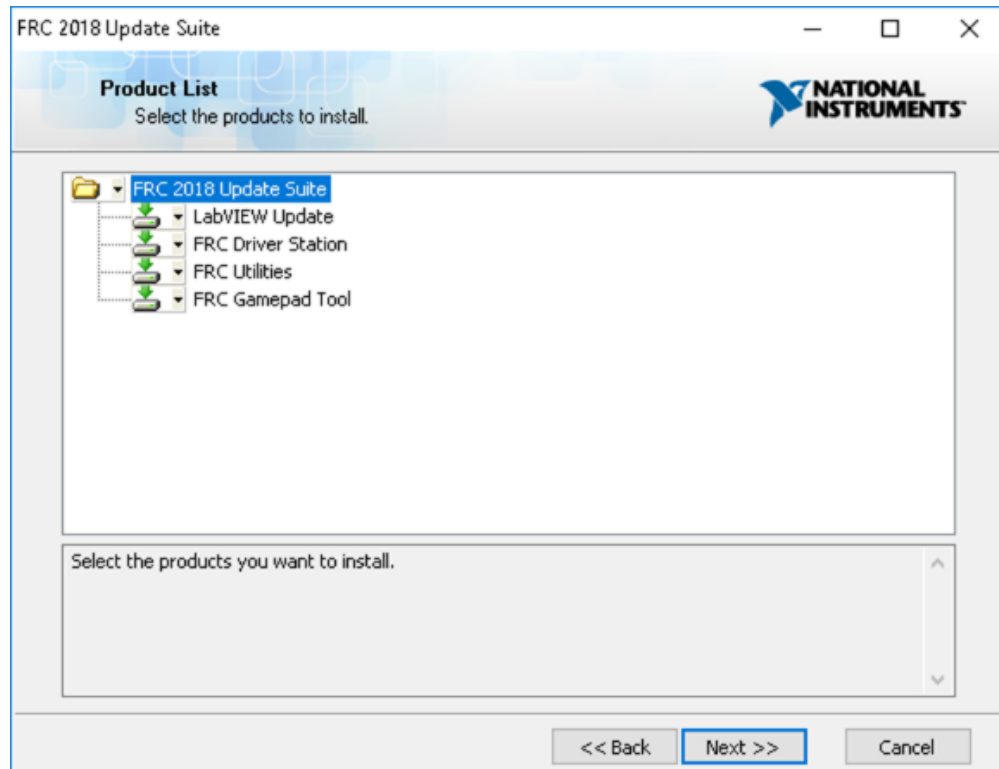
Welcome



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

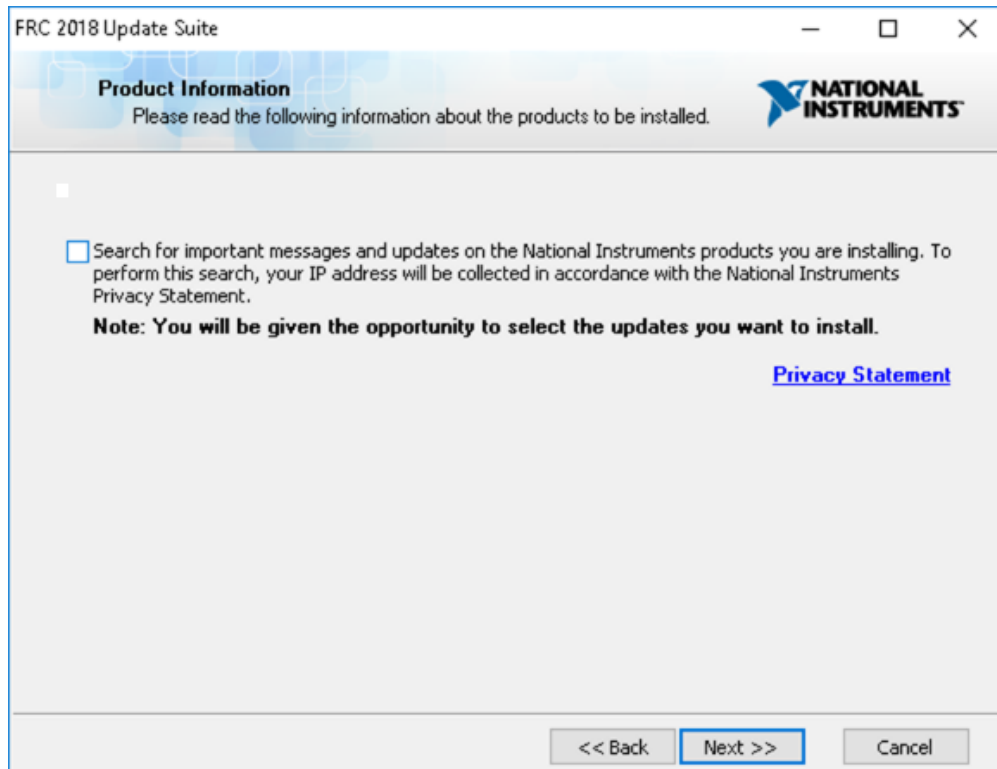
Product List



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

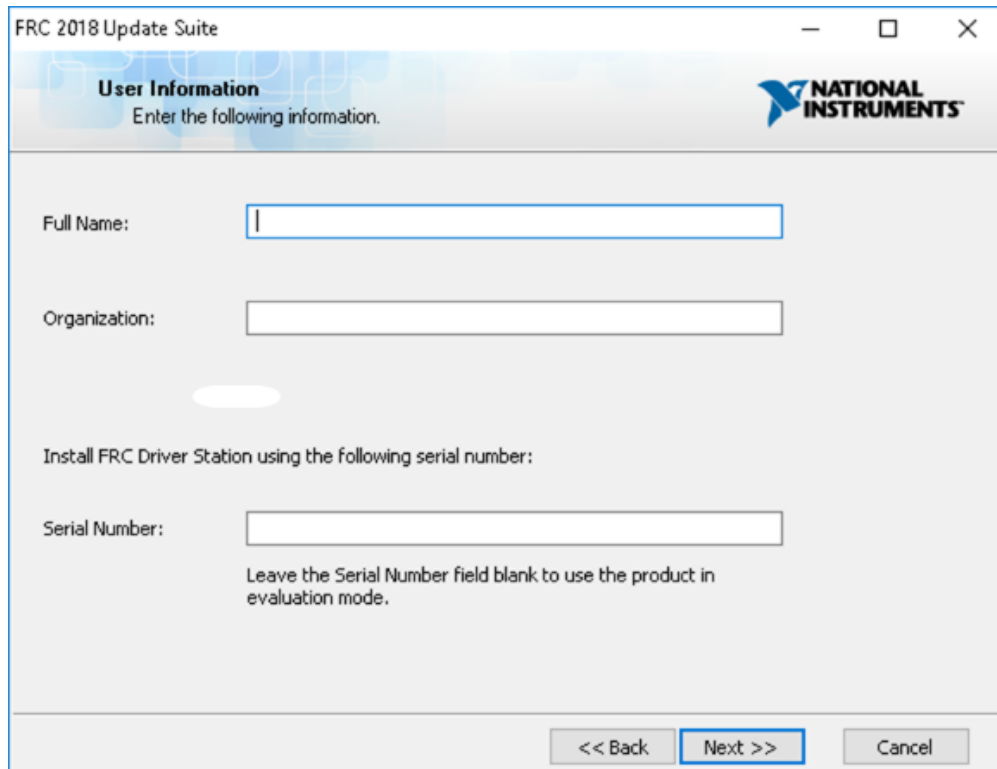


Un-check the box, then Click "Next".

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

Getting Started with the Control System

User Information

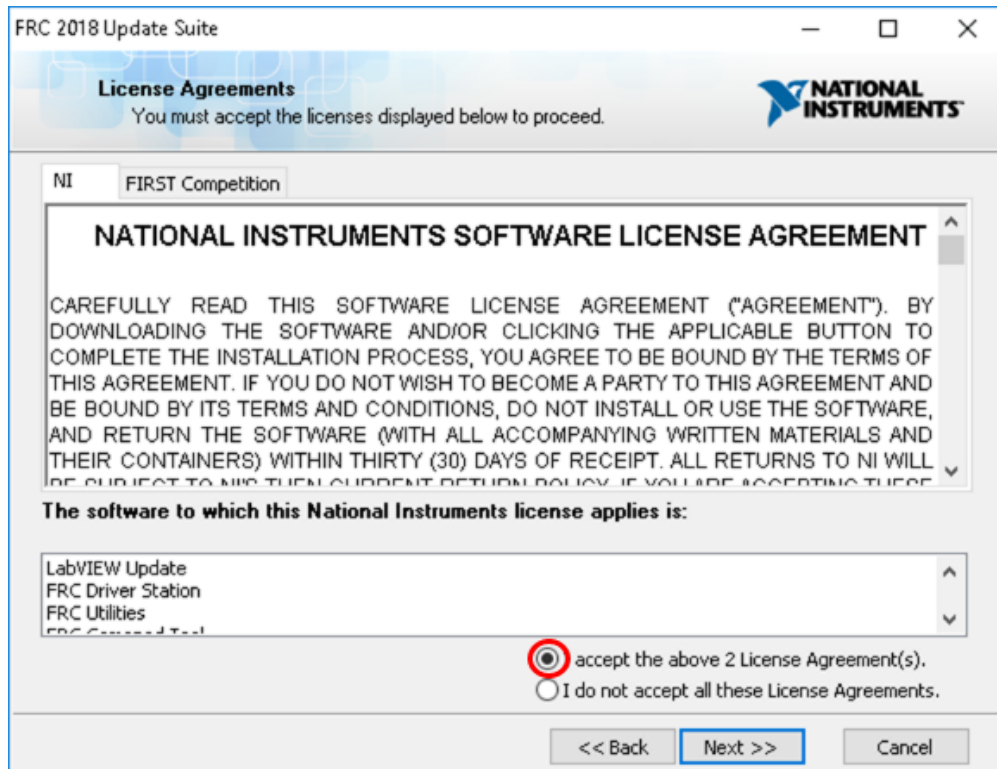


The screenshot shows a Windows-style dialog box titled "FRC 2018 Update Suite". The dialog has a header bar with the title and standard window controls (minimize, maximize, close). Below the header, the text "User Information" is displayed in bold, followed by "Enter the following information." and the National Instruments logo. The main area contains three input fields: "Full Name:" with a text box, "Organization:" with a text box, and "Serial Number:" with a text box. Below the "Serial Number" field, there is a note: "Leave the Serial Number field blank to use the product in evaluation mode." At the bottom of the dialog, there are three buttons: "<< Back", "Next >>" (which is highlighted with a blue border), and "Cancel".

Enter full name and organization and the serial number from your kit of parts then click **Next**

Getting Started with the Control System

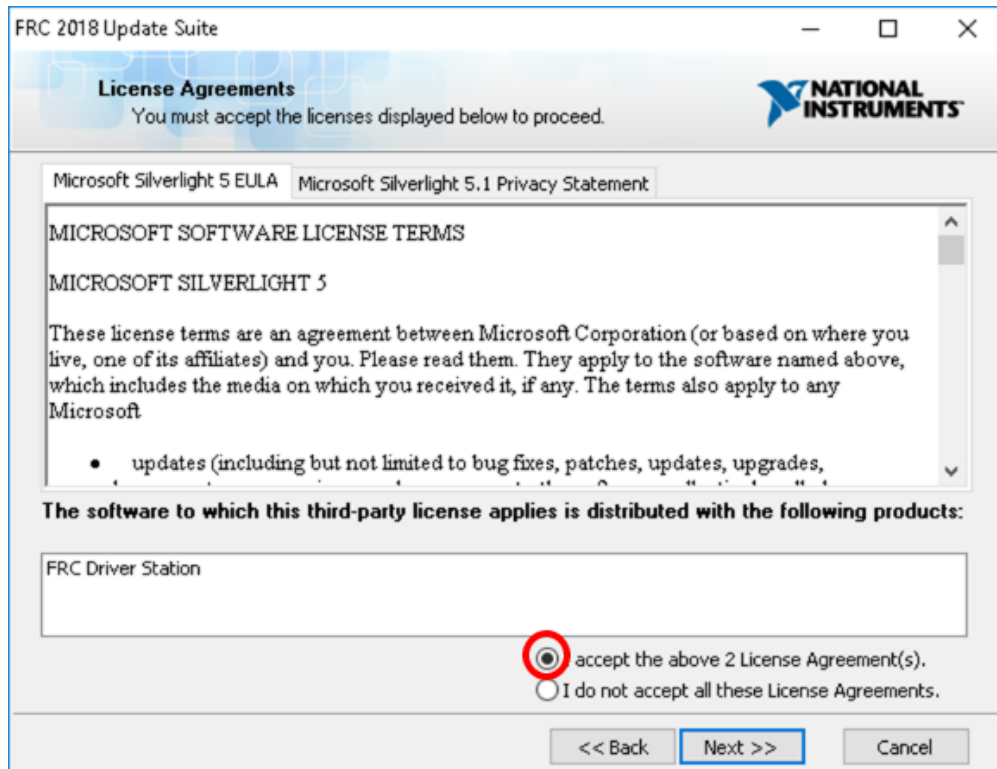
License Agreements



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

Getting Started with the Control System

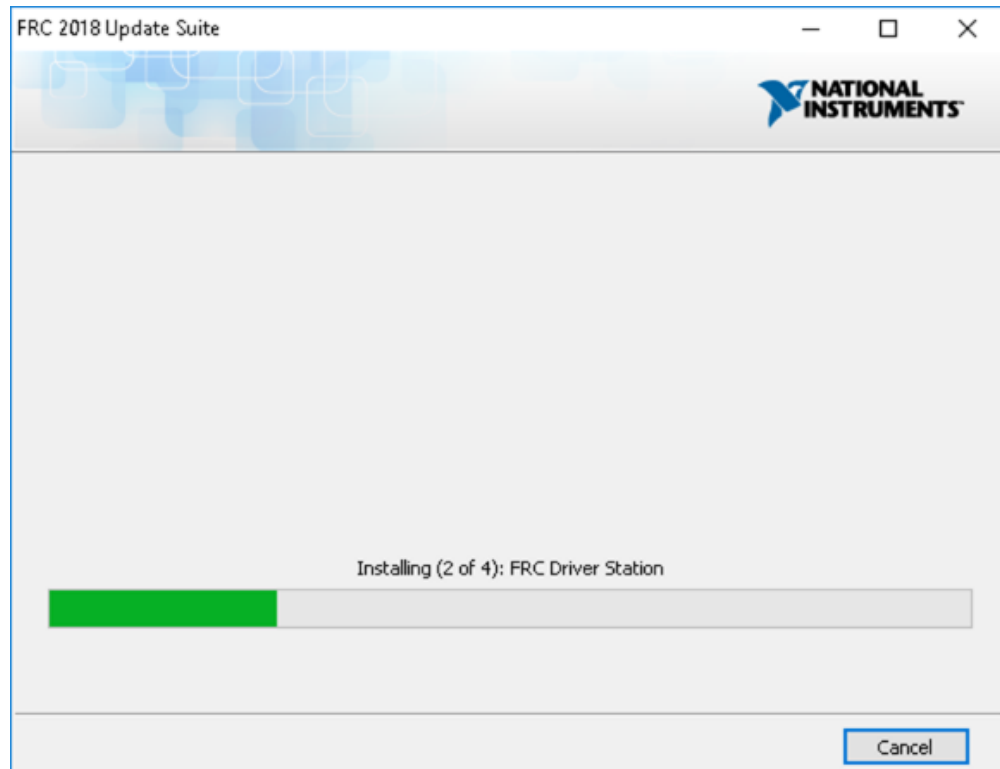
License Agreements Page 2



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

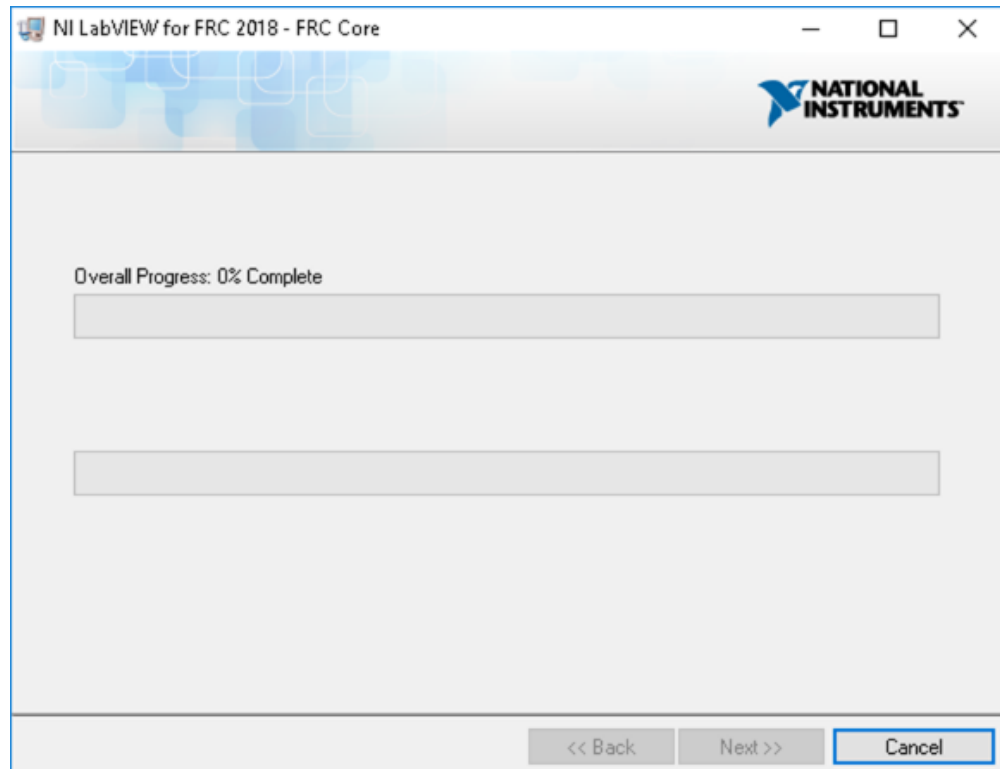
Getting Started with the Control System

Summary Progress



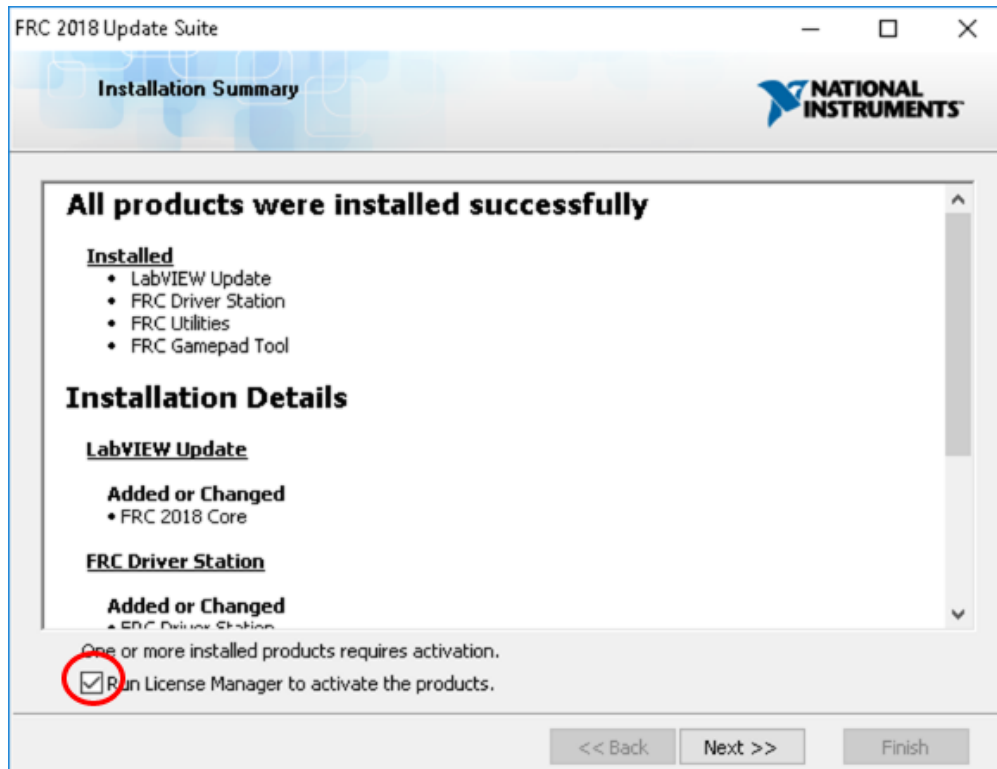
Getting Started with the Control System

Detail Progress



Getting Started with the Control System

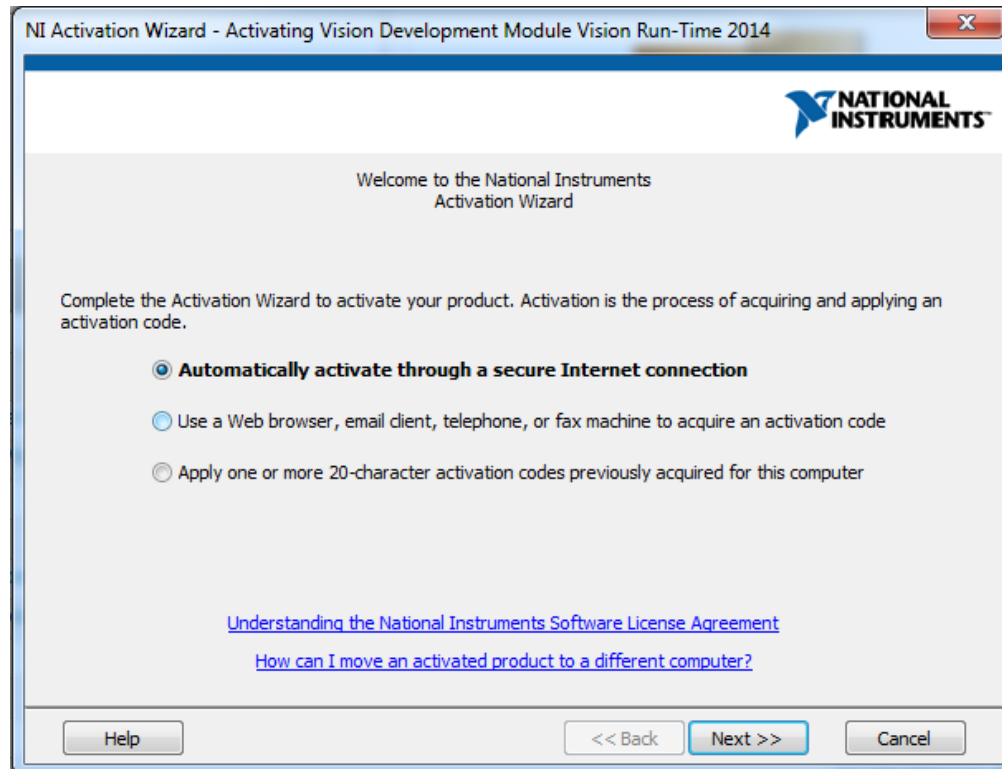
Installation Summary



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

Getting Started with the Control System

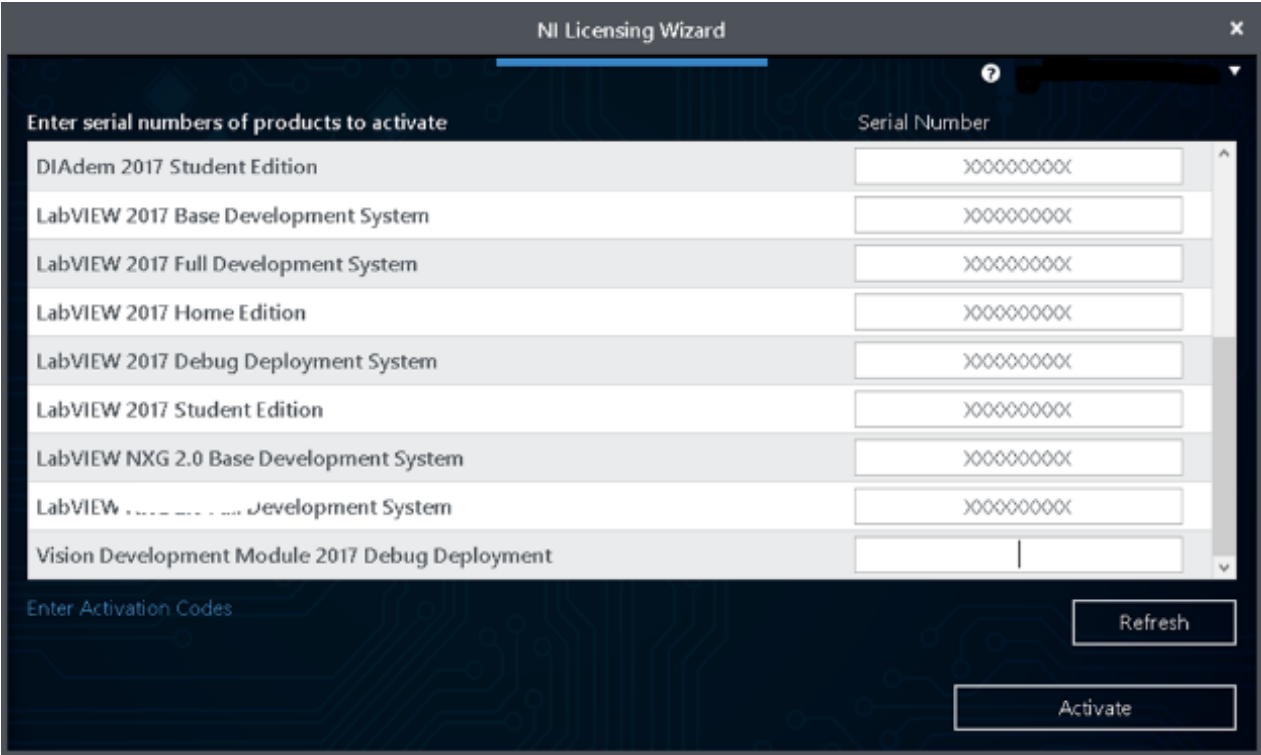
Activate



Select your desired activation method (Internet activation recommended), then click Next

Getting Started with the Control System

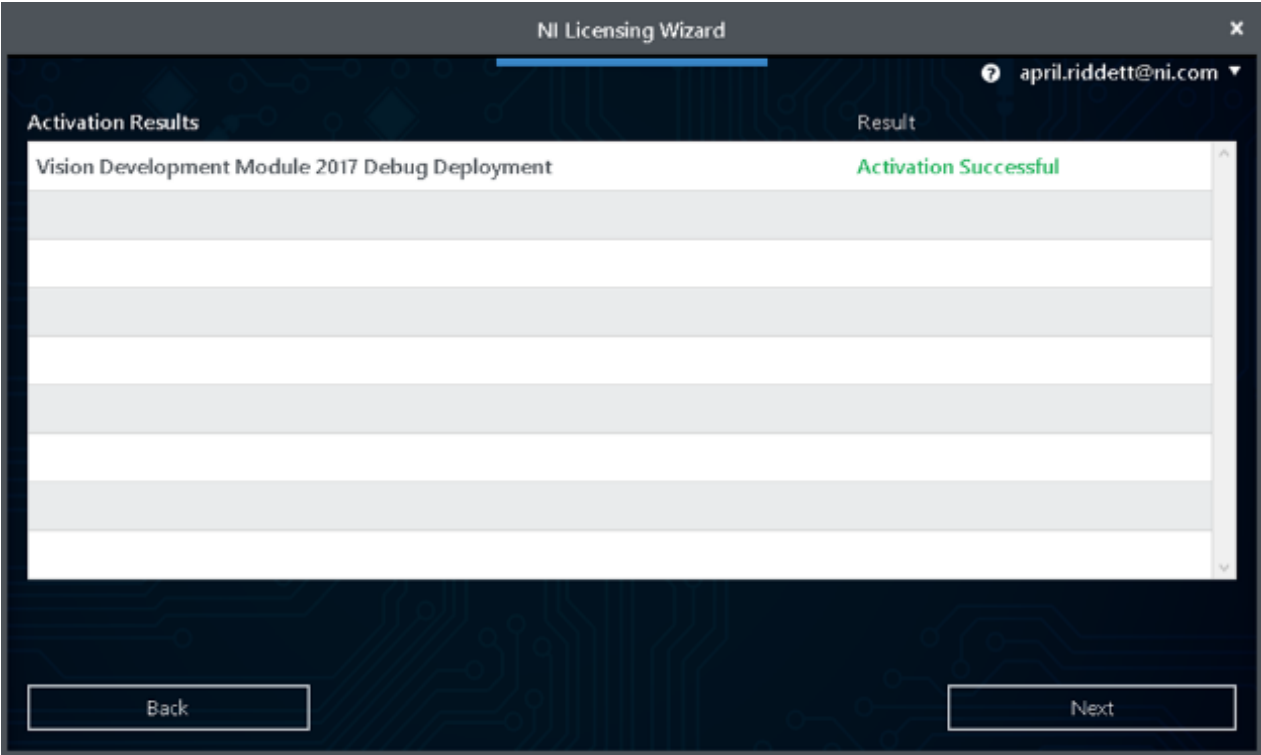
Serial Number Confirmation



Log into the NI Licensing Wizard, then enter the serial number from your LabVIEW packet from the Kit of Parts and press Activate.

Getting Started with the Control System

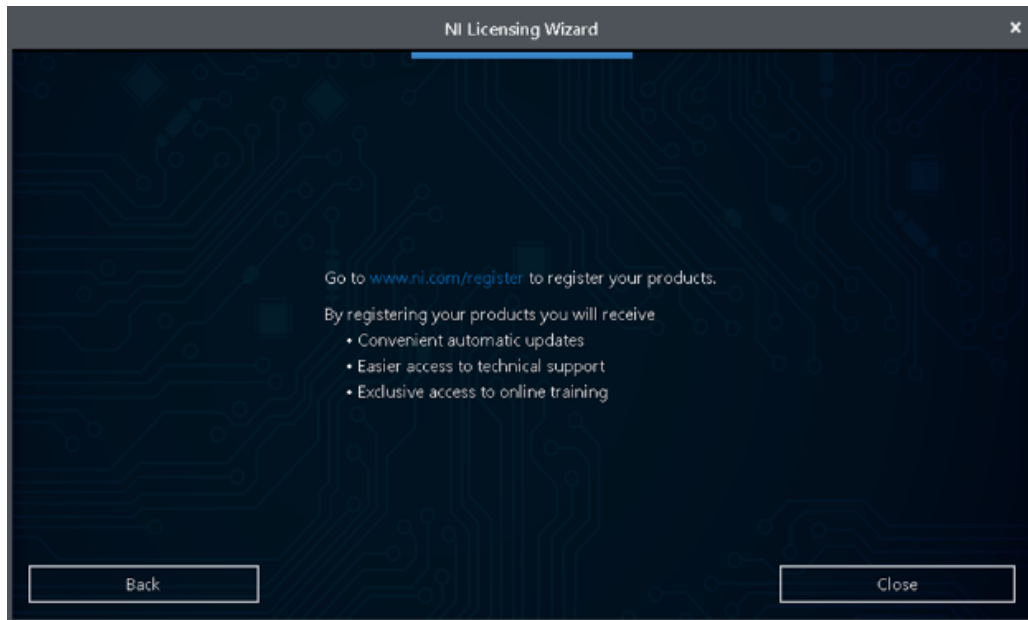
Finish Installation



After the product has activated successfully, click **Next**.

Getting Started with the Control System

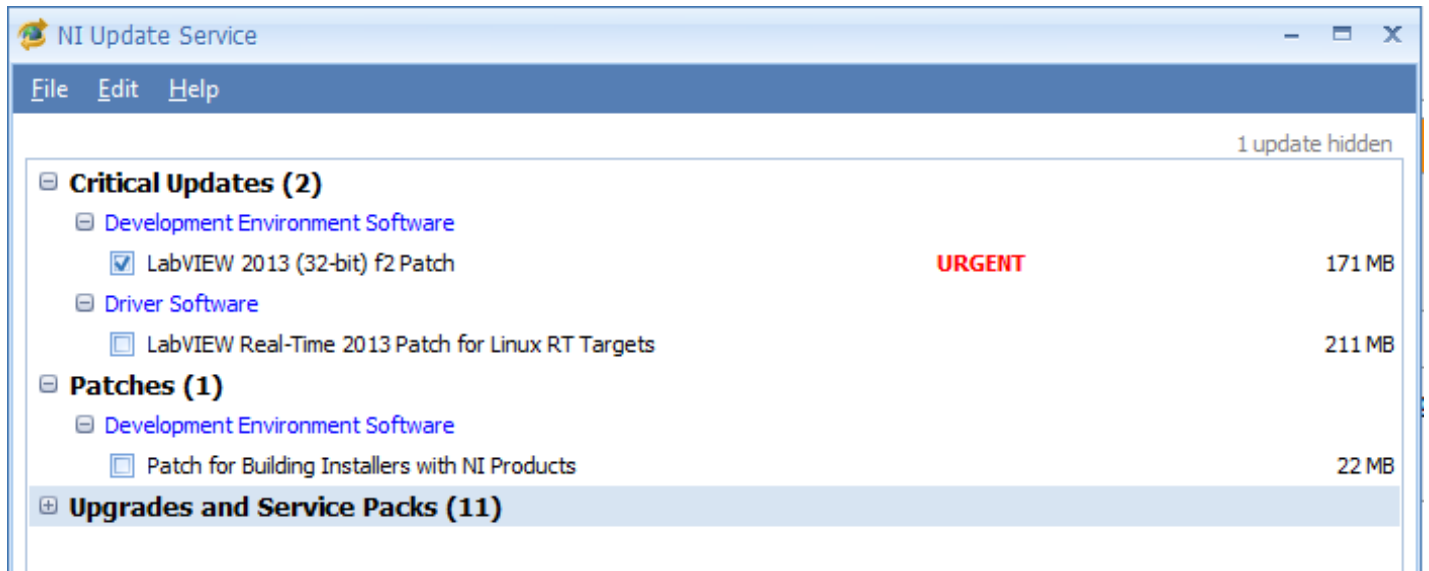
Finish Installation



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).

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

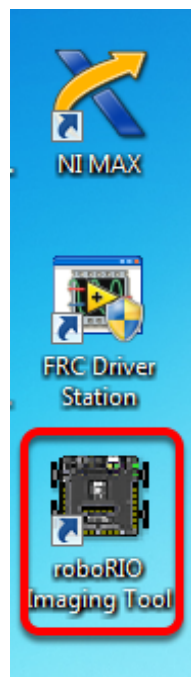
Connect a USB cable from the roboRIO USB Device port to the PC. This requires a USB Type A male (standard PC end) to Type B male cable (square with 2 cut corners), most commonly found as a printer USB cable.

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.

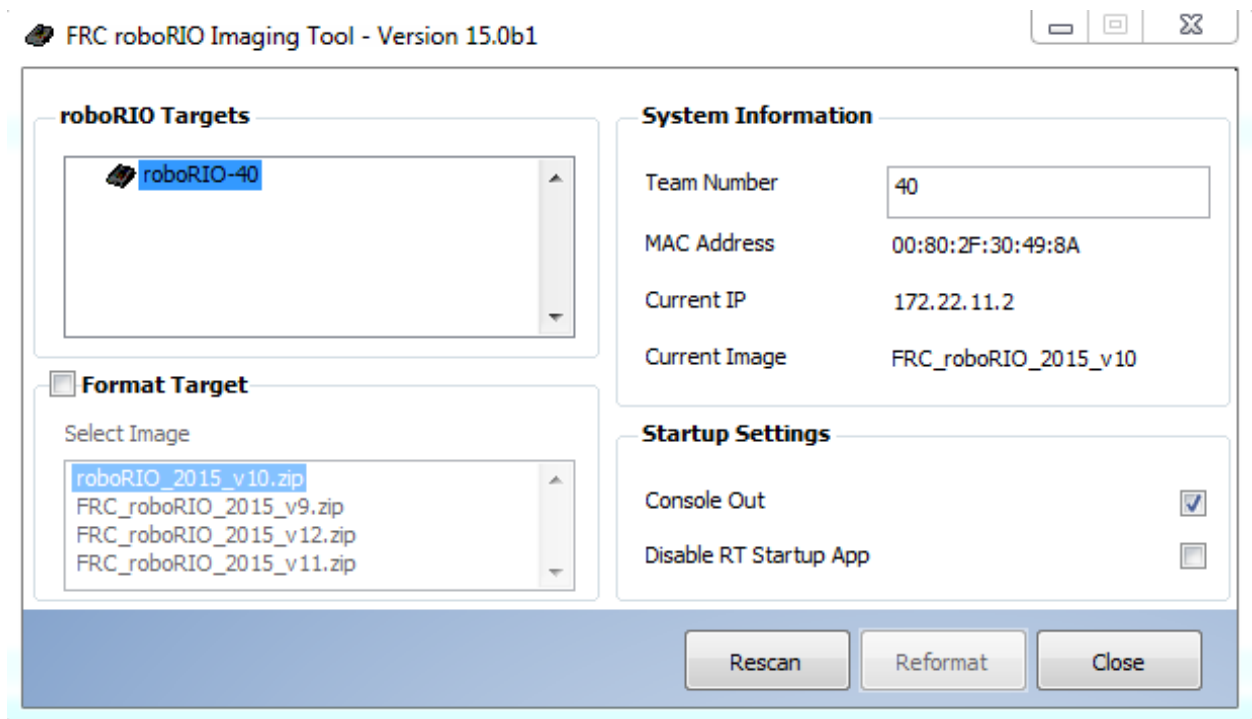
Launching the Imaging Tool



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

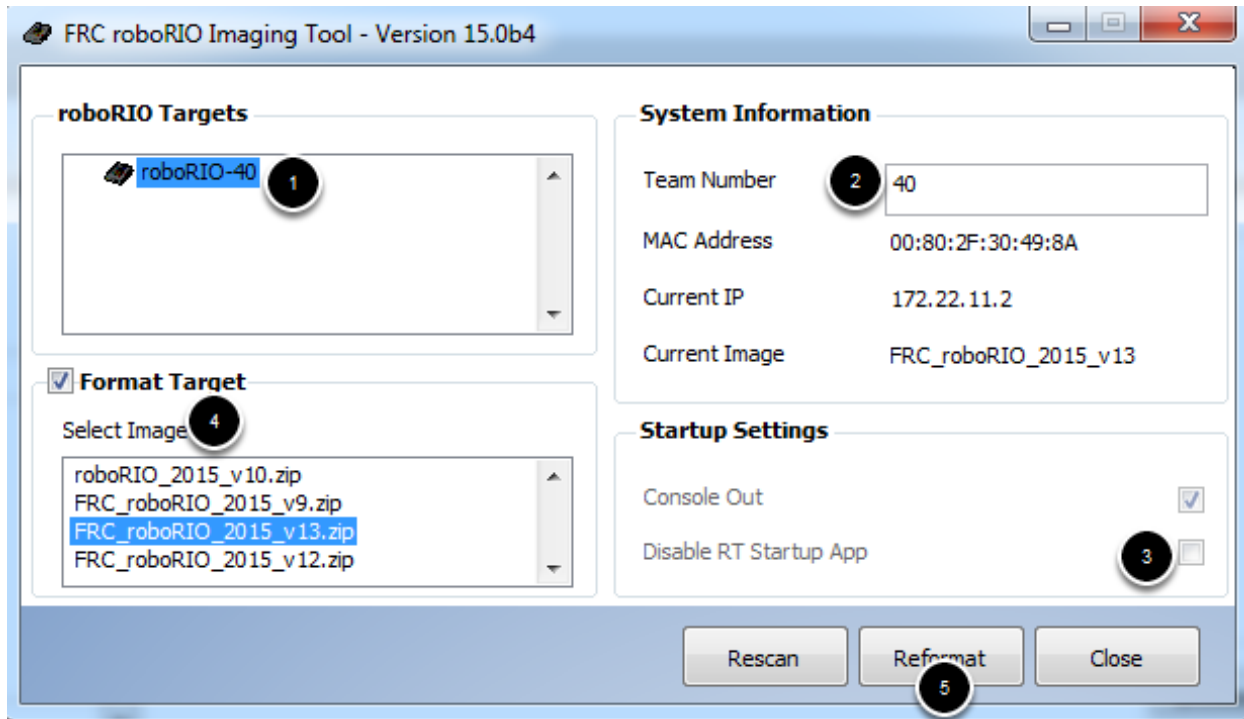
roboRIO Imaging Tool



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 the available image versions that may be loaded onto the roboRIO. The right hand column contains information and settings for the roboRIO selected in the top left pane.

Getting Started with the Control System

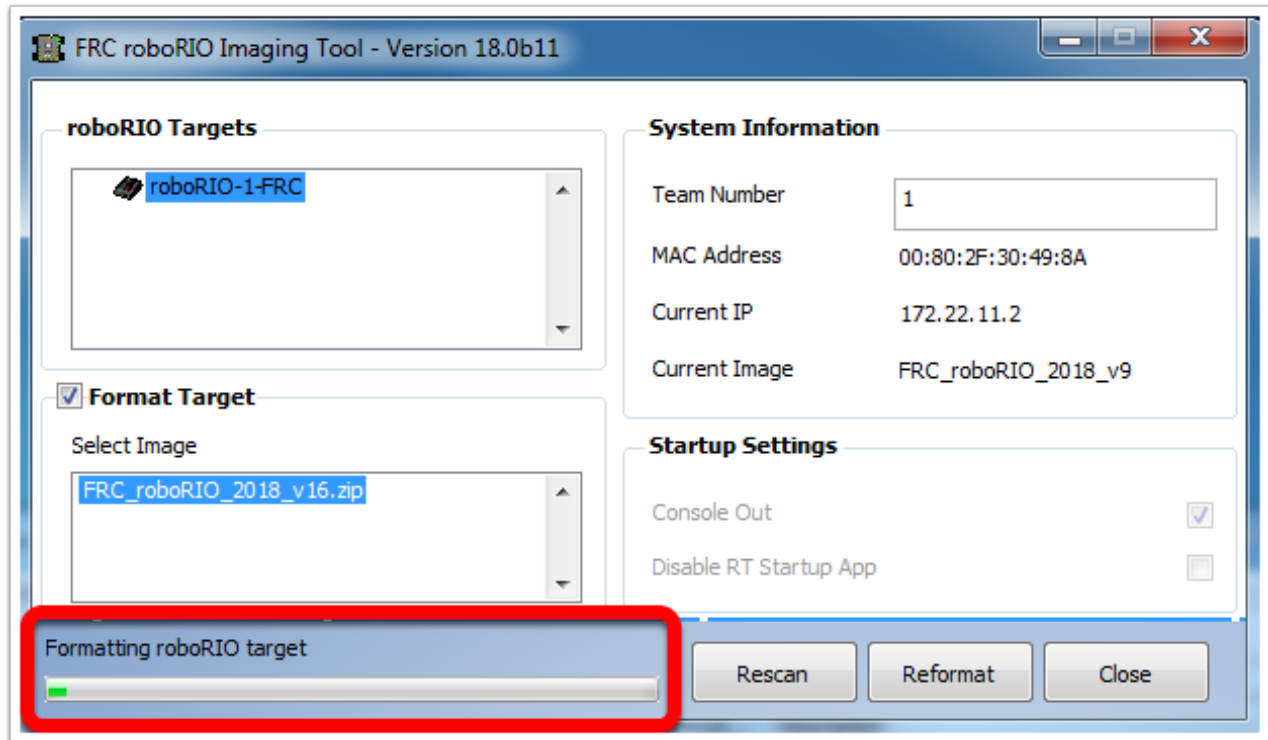
Imaging the roboRIO



1. Make sure the roboRIO is selected in the top left pane
2. Enter your team number in the box in the top right
3. Make sure the Disable RT Startup App box is unchecked in the bottom right
4. Check the box to Format Target and select the latest image version in the box.
5. Click Reformat to begin the imaging process.

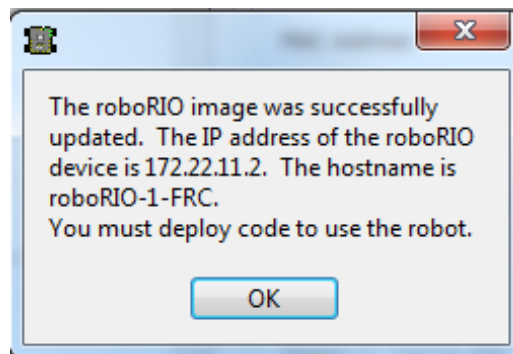
Getting Started with the Control System

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 [Phoenix Lifeboat tool from CTRE](#) to install the webdash plugin after imaging.

 Note: There is no longer a separate Java Installation process required after imaging for Java teams. The WPILib Eclipse plugin will deploy an OpenJDK version of the JRE to the roboRIO if necessary on each code deploy.

Troubleshooting

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

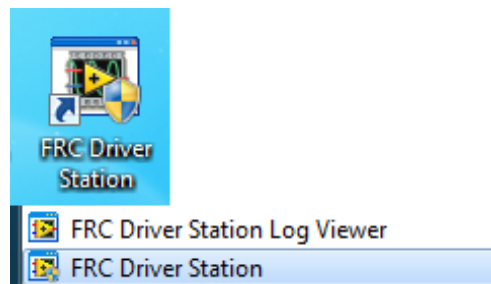
1. Try running the roboRIO Imaging Tool as Administrator by right-clicking on the Desktop icon to launch it.
2. Try accessing the roboRIO webpage with a web-browser at <http://172.22.11.2/> and/or verify that the NI network adapter appears in your list of Network Adapters in the Control Panel. If not, 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. Make sure your roboRIO firmware is up to date using the instructions here: [Updating your roboRIO firmware](#)
5. Try a different PC
6. 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

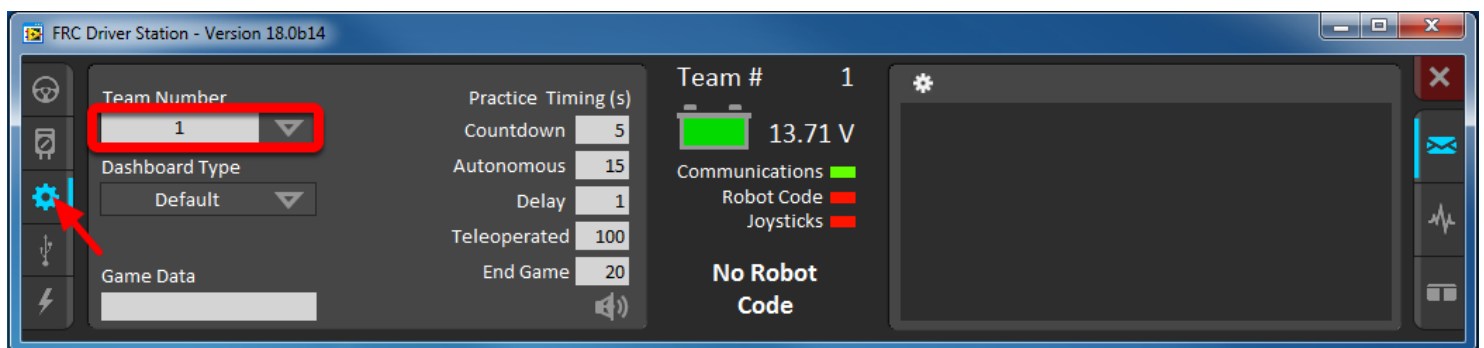
This article describes the use and features of the FRC Driver Station Powered by NI LabVIEW.
For information on installing the Driver Station software see [this document](#).

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

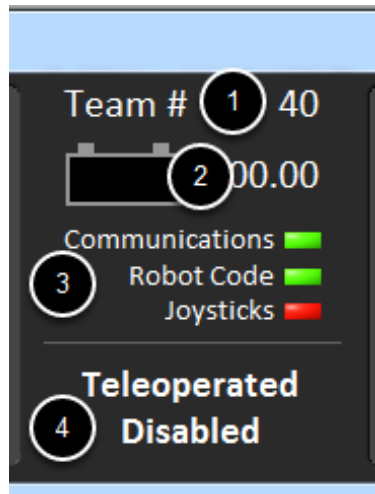


The DS must be set to your team number in order to connect to your robot. In order to do this click the **Setup** tab then enter your team number in the team number box. Press return or click outside the box for the setting to take effect.

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

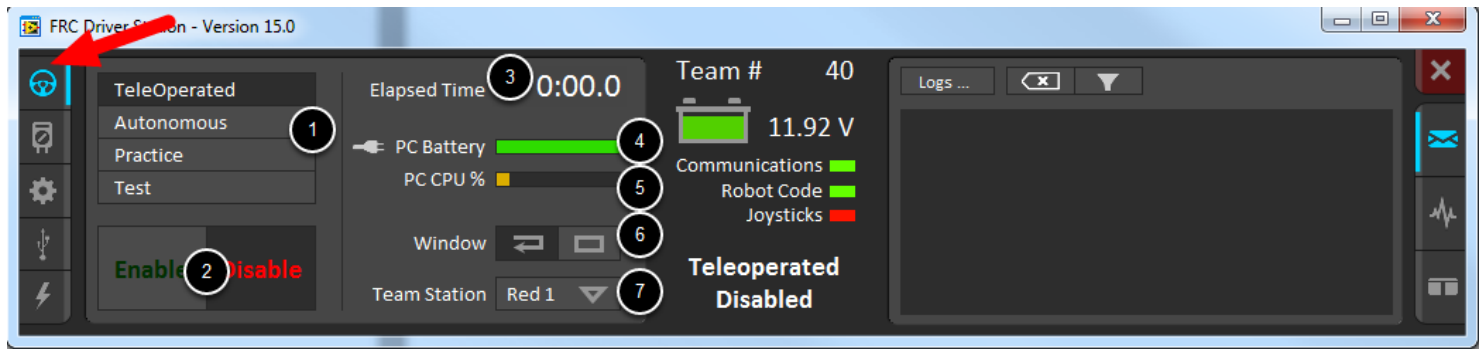


The Status Pane of the Driver Station is located in the center of the display and is always visible regardless of the tab selected. It displays a selection of critical information about the state of the DS and robot:

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 UDO 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



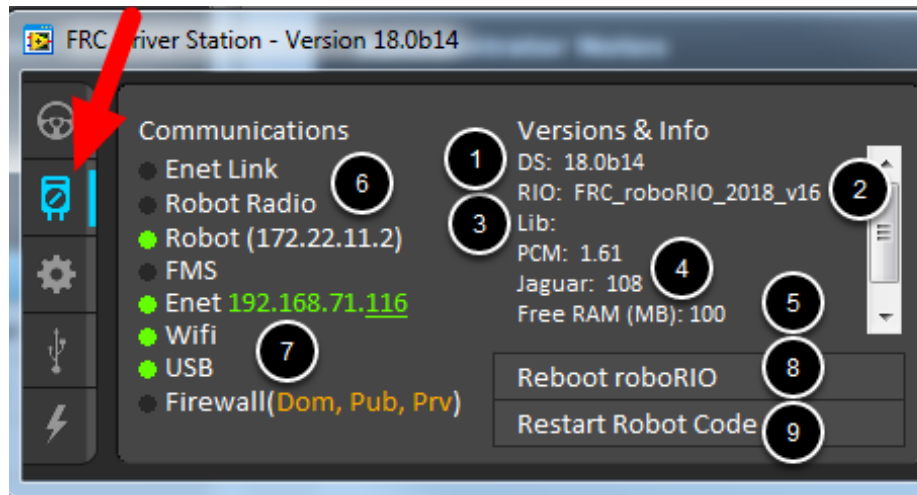
The Operations Tab is used to control the mode of the robot and provide additional key status indicators while the robot is running.

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 `[] \` (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



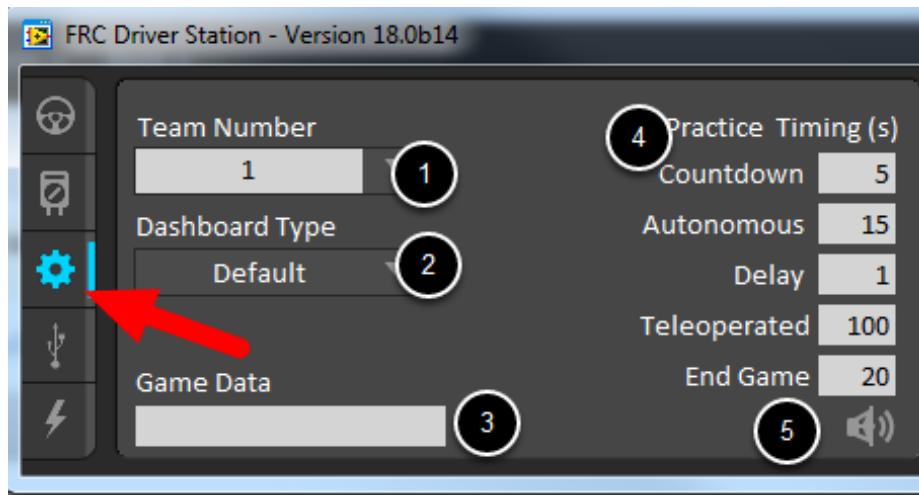
The Diagnostics Tab contains additional status indicators that teams can use to diagnose issues with their robot:

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



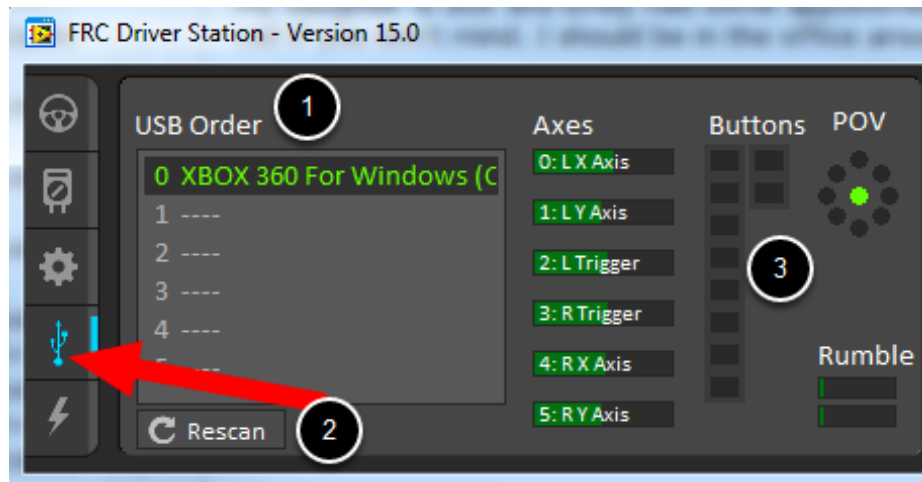
The Setup Tab contains a number of buttons teams can use to control the operation of the Driver Station:

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

- 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.
- Audio Control - This button controls whether audio tones are sounded when the Practice Mode is used.

USB Devices Tab

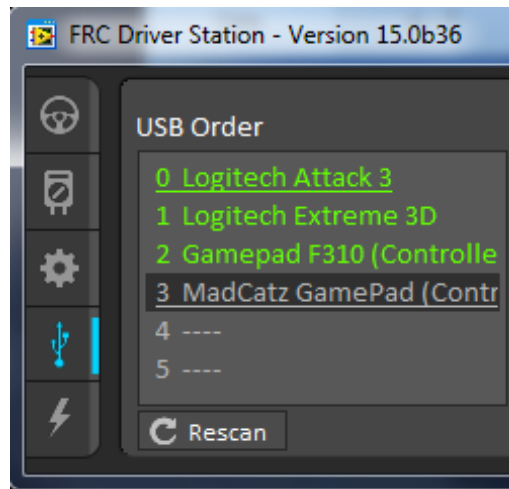


The USB Devices tab includes the information about the USB Devices connected to the DS

- 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
- 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.
- Device indicators - These indicators show the current status of the Axes, buttons and POV of the joystick.
- 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

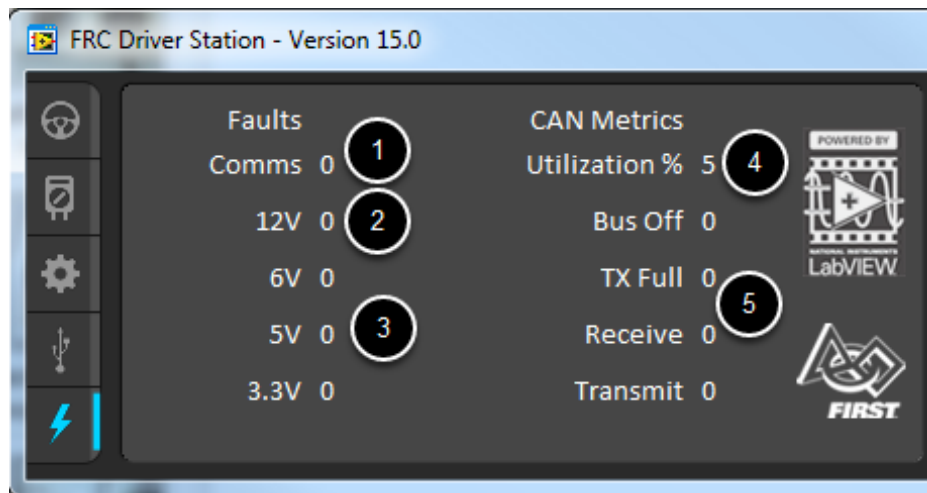
Getting Started with the Control System

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



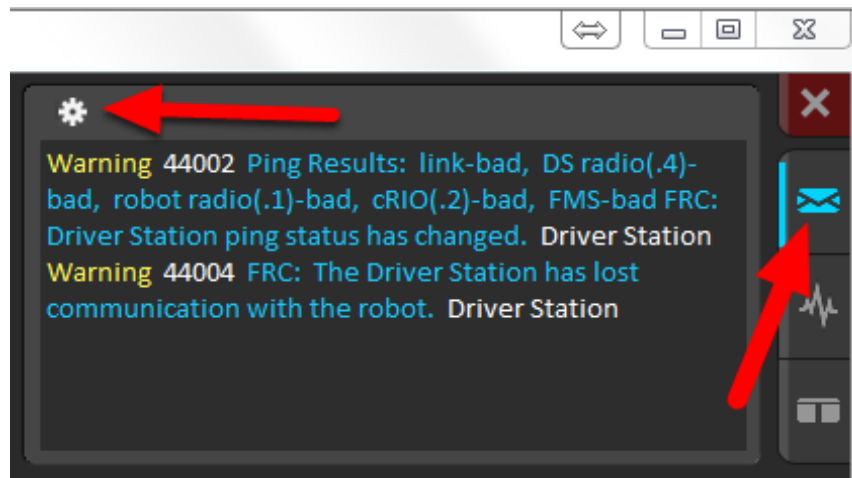
The last tab on the left side of the DS is the CAN\Robot Power Tab. This tab contains information about the power status of the roboRIO and the status of the CAN bus:

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

Getting Started with the Control System

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

1. Confirm that battery voltage is displayed

Getting Started with the Control System

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, you must 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.

 The 2018 OM5P-AC firmware is believed to resolve the issue with the 2nd Ethernet port.

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

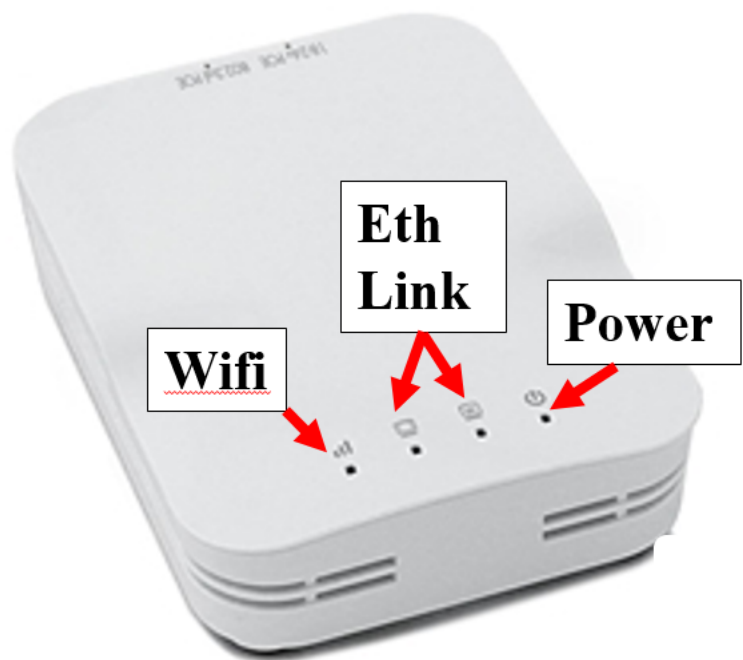
By default, the Radio Configuration Utility will program the radio to enforce the 7Mbps bandwidth limit on traffic exiting the radio over the wireless interface. In the home configuration (AP mode) this is a total, not a per client limit. This means that streaming video to multiple clients is not recommended.

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	
	Bridge Mode, Unlinked or non-FRC firmware
Off	
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

- 7Mb/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 18.1.0](#)

[FRC Radio Configuration 18.1.0 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.

Install the software

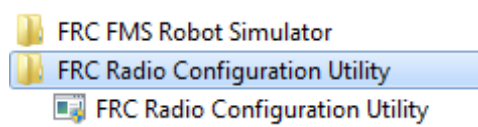
	FRC_Radio_Configuration_10_8_15.exe	10/8/2015 1:50 PM	Application
	FRCicon_RGB_Border.bmp	2/20/2015 1:27 PM	Bitmap Image

Double click on FRC_Radio_Configuration_VERSION.exe to launch the installer. Follow the prompts to complete the installation.

Getting Started with the Control System

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.

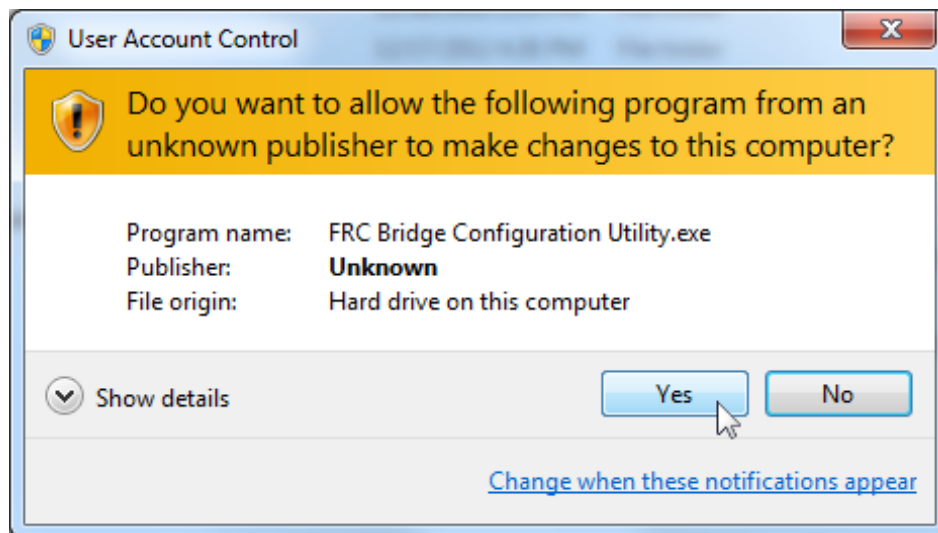
Launch the software



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\

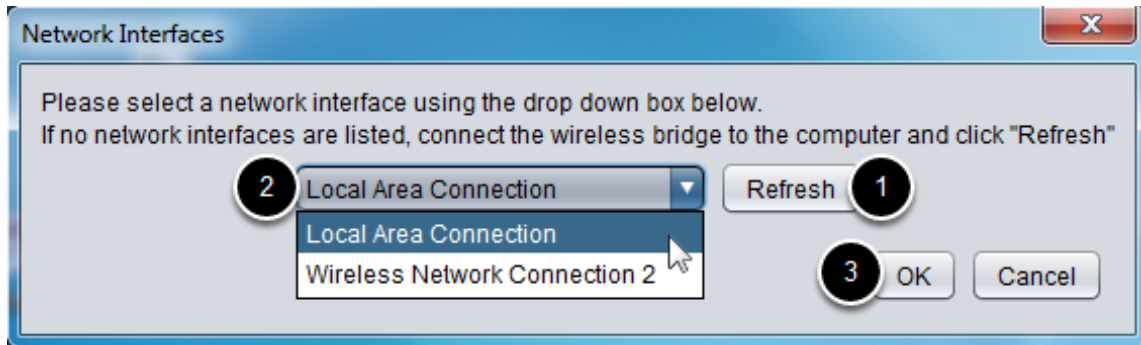
Allow the program to make changes, if prompted



If the 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.

Getting Started with the Control System

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

Open Mesh Firmware Note

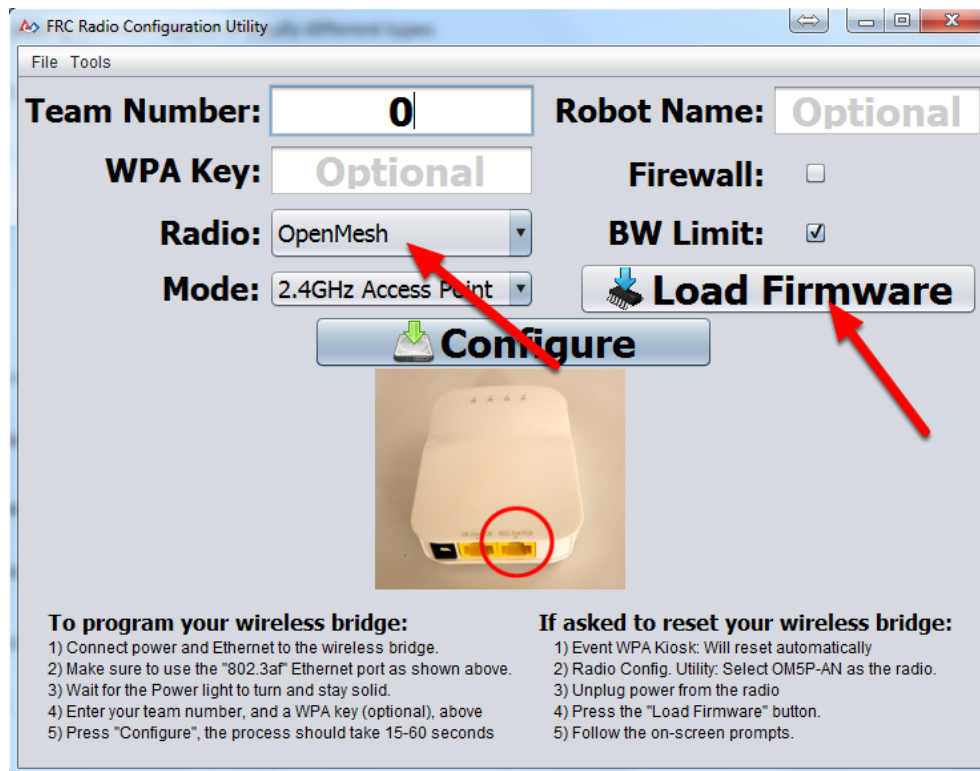
For the FRC Radio Configuration Utility to program the OM5P-AN and OM5P-AC radio, the radio must be running an FRC specific build of the OpenWRT firmware. OM5P-AC radios with the received firmware must be updated, the Radio Configuration Utility should throw an "old firmware" notification if it is not. OM5P-AN radios must be updated to the new firmware in order for the utility to program them, they will not be detected by the utility if you attempt to Configure before updating.

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

 **Note:** Radios used in 2017 will need to be updated to 2018 firmware before configuring.

Getting Started with the Control System

Loading FRC Firmware to OpenMesh radio



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
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.

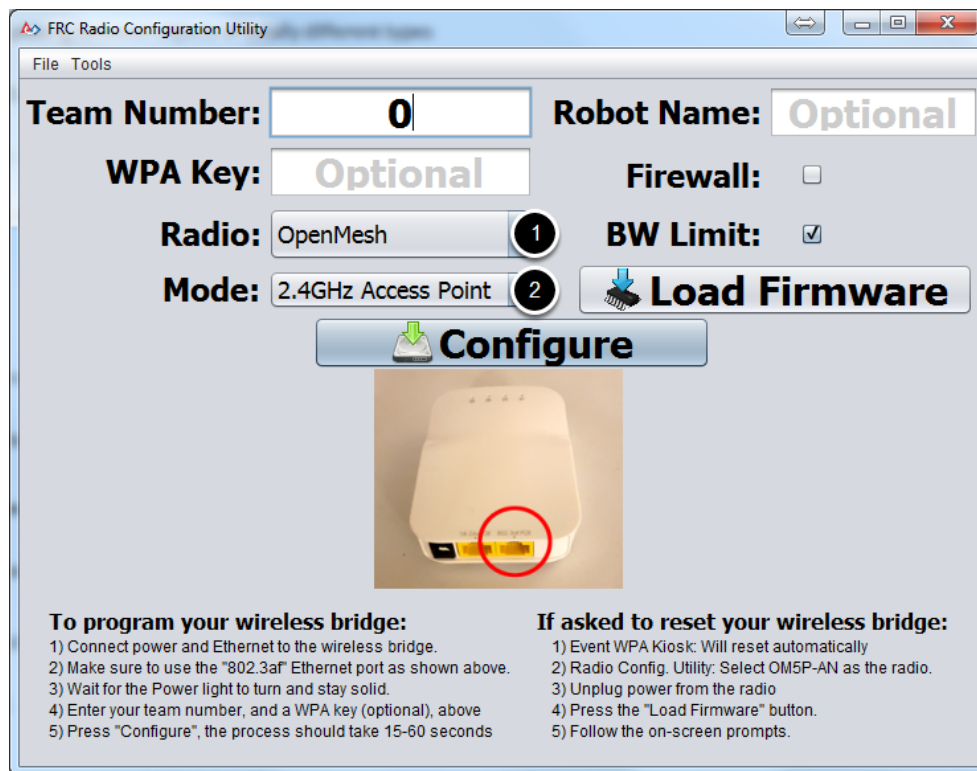
 If you see an error about NPF name, try disabling all adapters other than the one being

Getting Started with the Control System

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](#).

Select a bridge model and operating mode



FRC Radio Configuration Utility

File Tools

Team Number: **Robot Name:**

WPA Key: **Firewall:** ☐

Radio: **BW Limit:** ☒

Mode: **Load Firmware**

Configure

To program your wireless bridge:

- 1) Connect power and Ethernet to the wireless bridge.
- 2) Make sure to use the "802.3af" Ethernet port as shown above.
- 3) Wait for the Power light to turn and stay solid.
- 4) Enter your team number, and a WPA key (optional), above
- 5) Press "Configure", the process should take 15-60 seconds

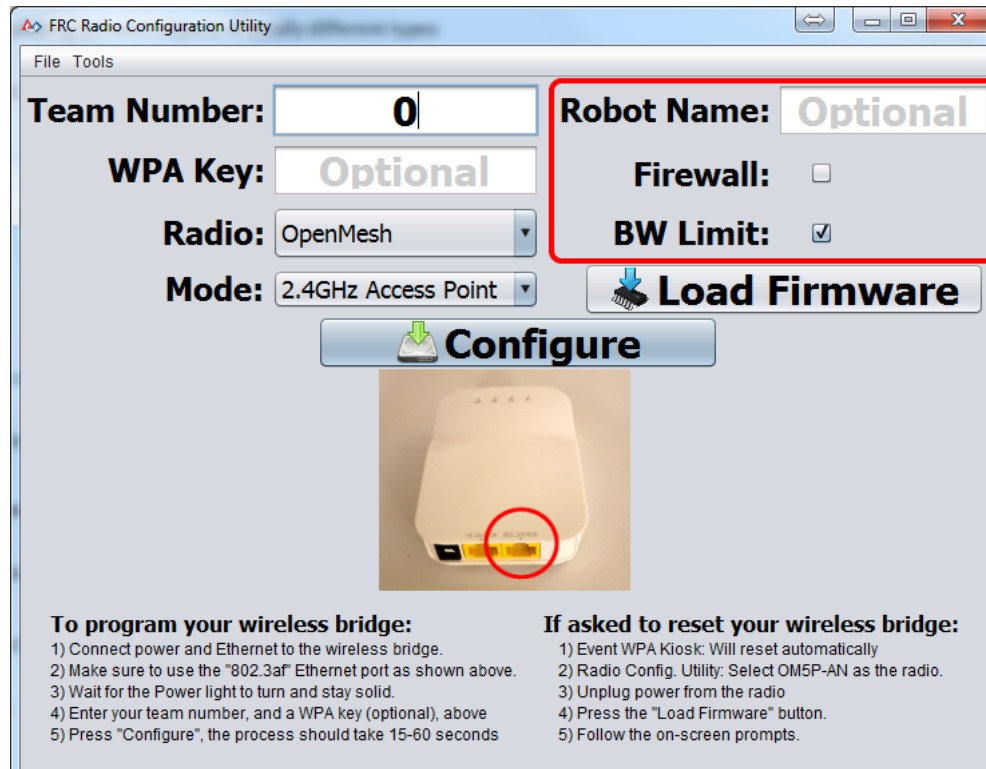
If asked to reset your wireless bridge:

- 1) Event WPA Kiosk: Will reset automatically
- 2) Radio Config. Utility: Select OM5P-AN as the radio.
- 3) Unplug power from the radio
- 4) Press the "Load Firmware" button.
- 5) Follow the on-screen prompts.

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:

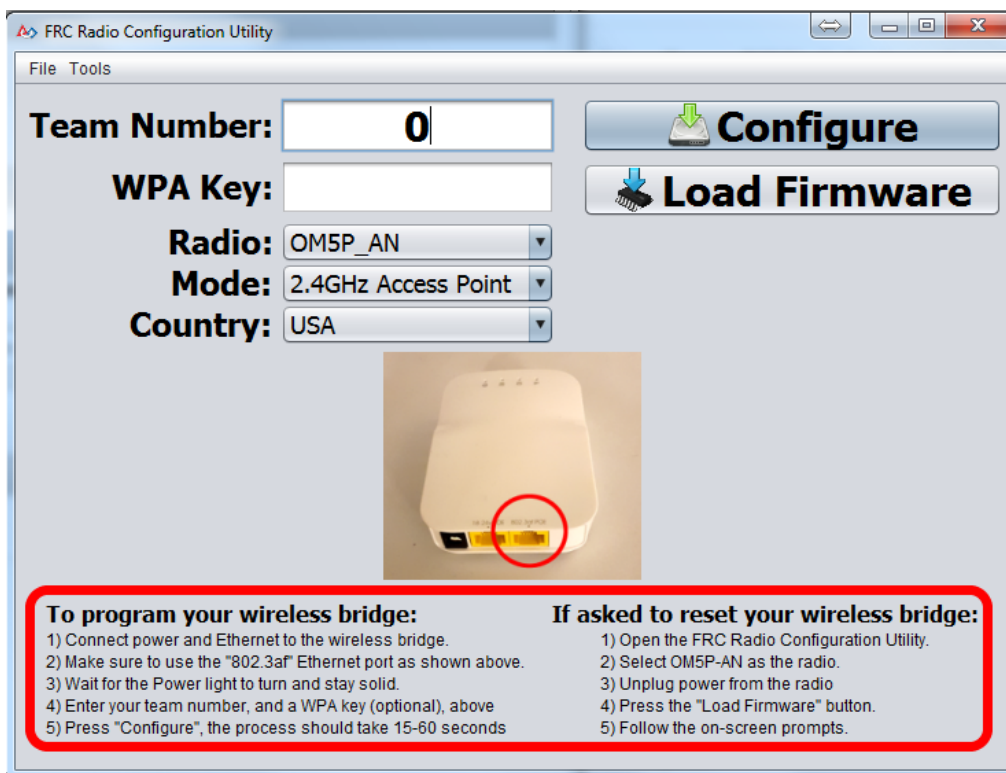
1. **Robot Name:** This is a string that gets appended to the SSID used by the radio. This allows you to have multiple networks with the same team number and still be able to distinguish them.
2. **Firewall:** If this box is checked, the radio firewall will be configured to attempt to mimic the port 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 7MB/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



Team Number:

WPA Key:

Radio: OM5P_AN

Mode: 2.4GHz Access Point

Country: USA

Configure

Load Firmware

To program your wireless bridge:

- 1) Connect power and Ethernet to the wireless bridge.
- 2) Make sure to use the "802.3af" Ethernet port as shown above.
- 3) Wait for the Power light to turn and stay solid.
- 4) Enter your team number, and a WPA key (optional), above
- 5) Press "Configure", the process should take 15-60 seconds

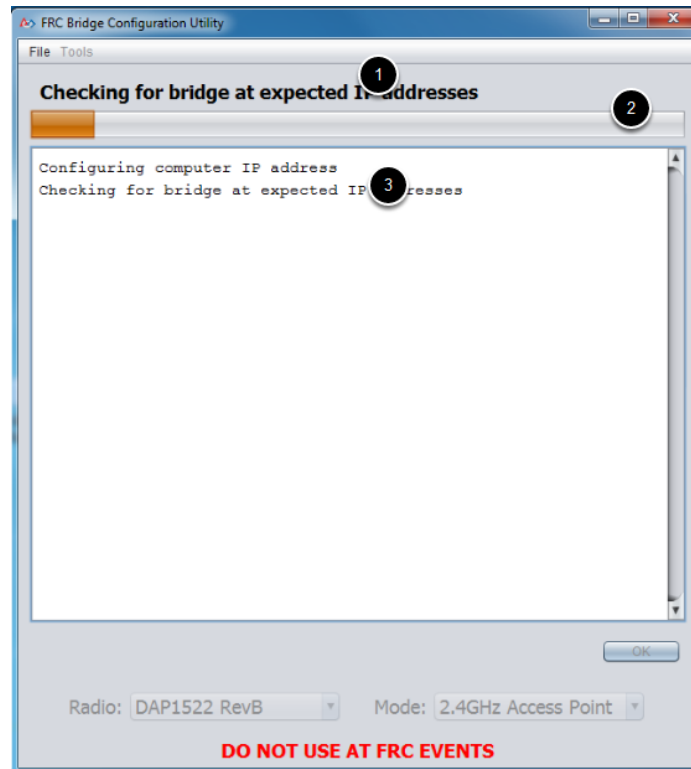
If asked to reset your wireless bridge:

- 1) Open the FRC Radio Configuration Utility.
- 2) Select OM5P-AN as the radio.
- 3) Unplug power from the radio
- 4) Press the "Load Firmware" button.
- 5) Follow the on-screen prompts.

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

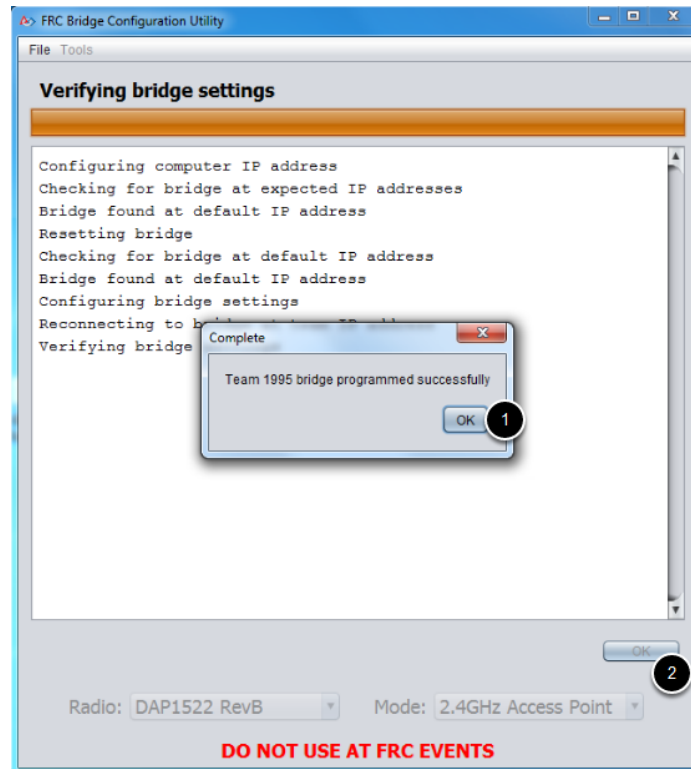


Throughout the configuration process, the window will indicate:

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

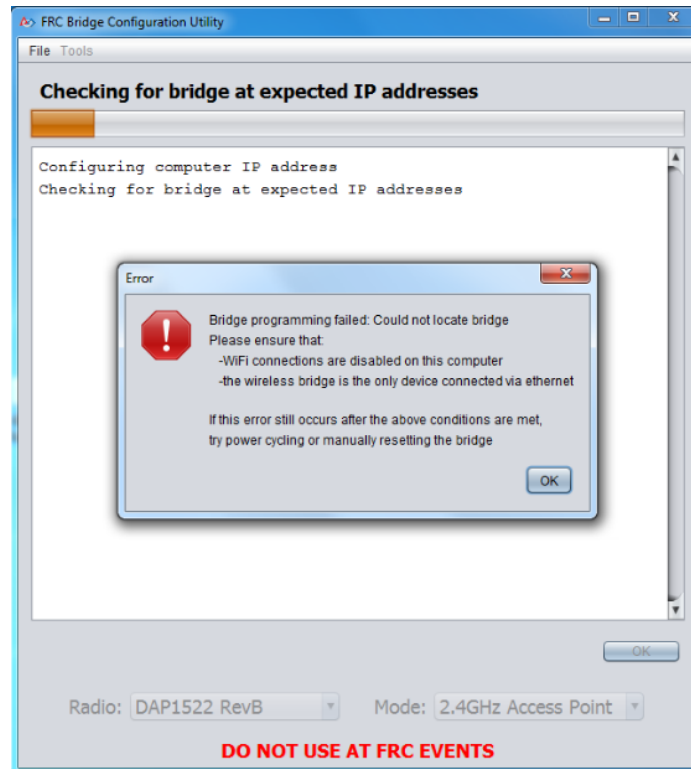


Once the configuration is complete:

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

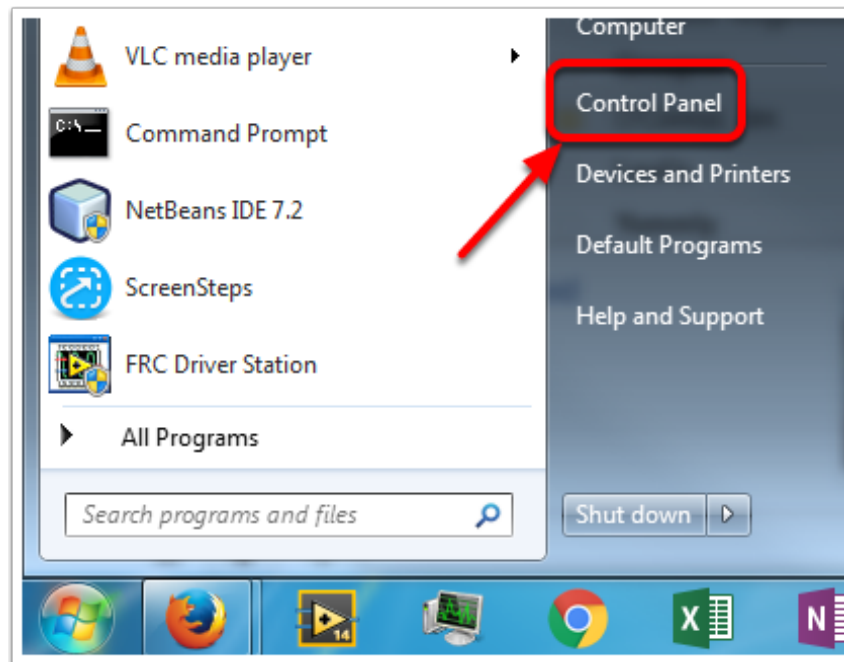


If an error occurs during the configuration process, follow the instructions in the error message to correct the problem.

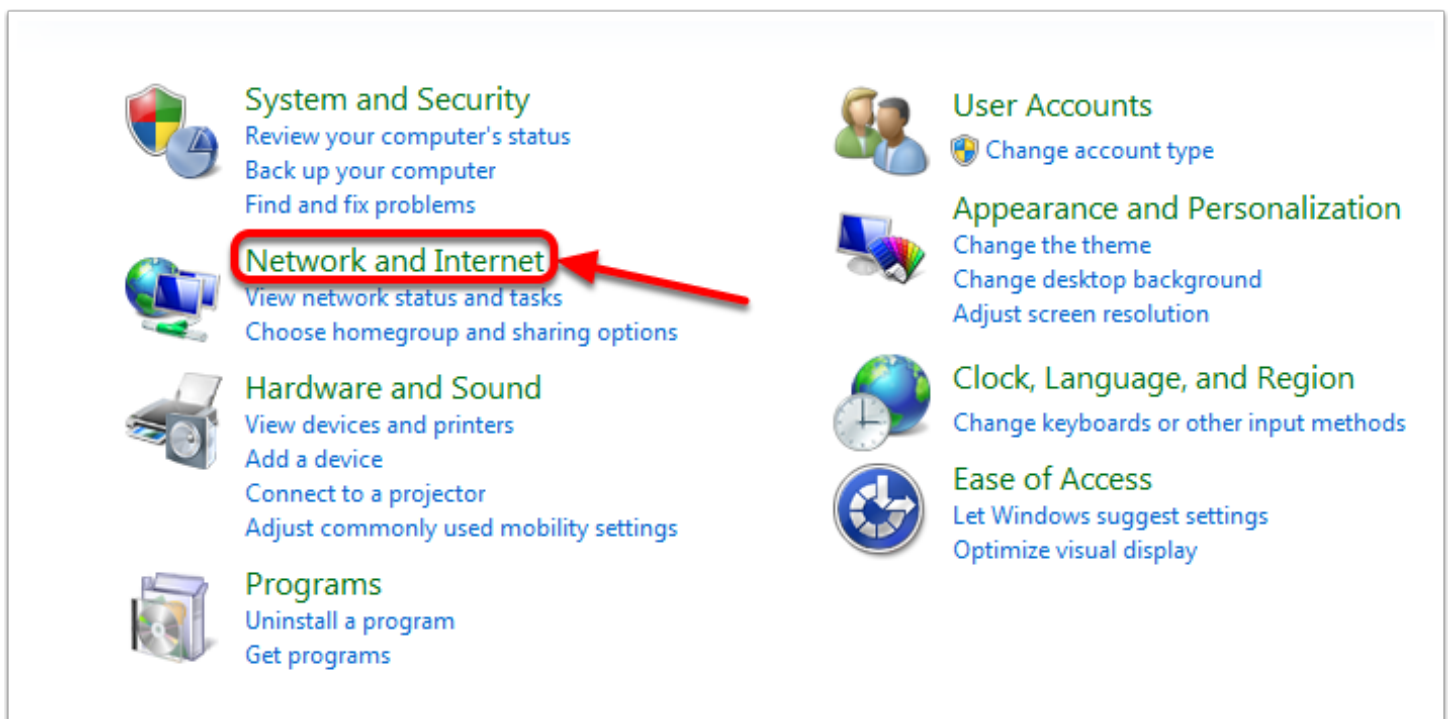
Troubleshooting: Disabling Network Adapters

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

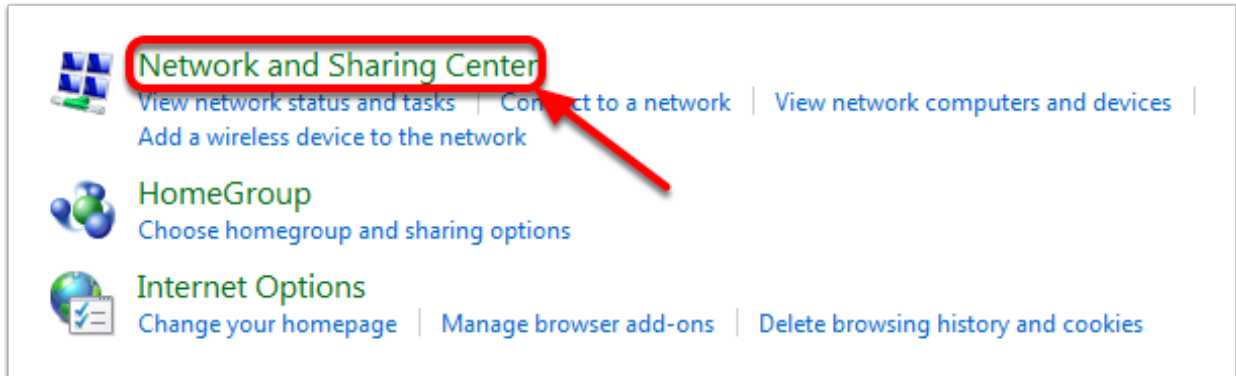


Open the Control Panel by going to Start->Control Panel

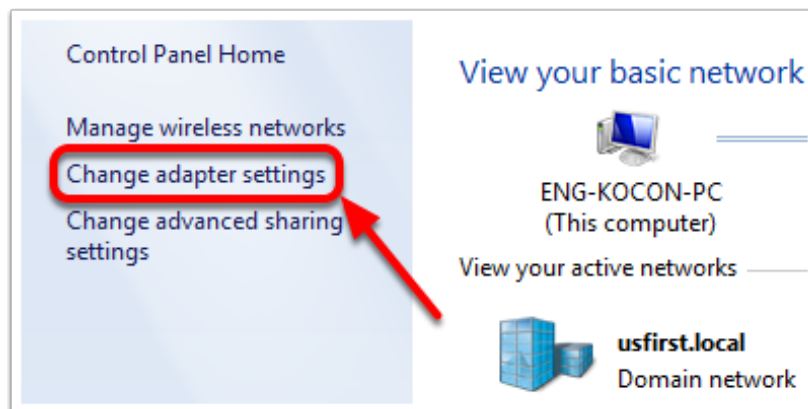


Getting Started with the Control System

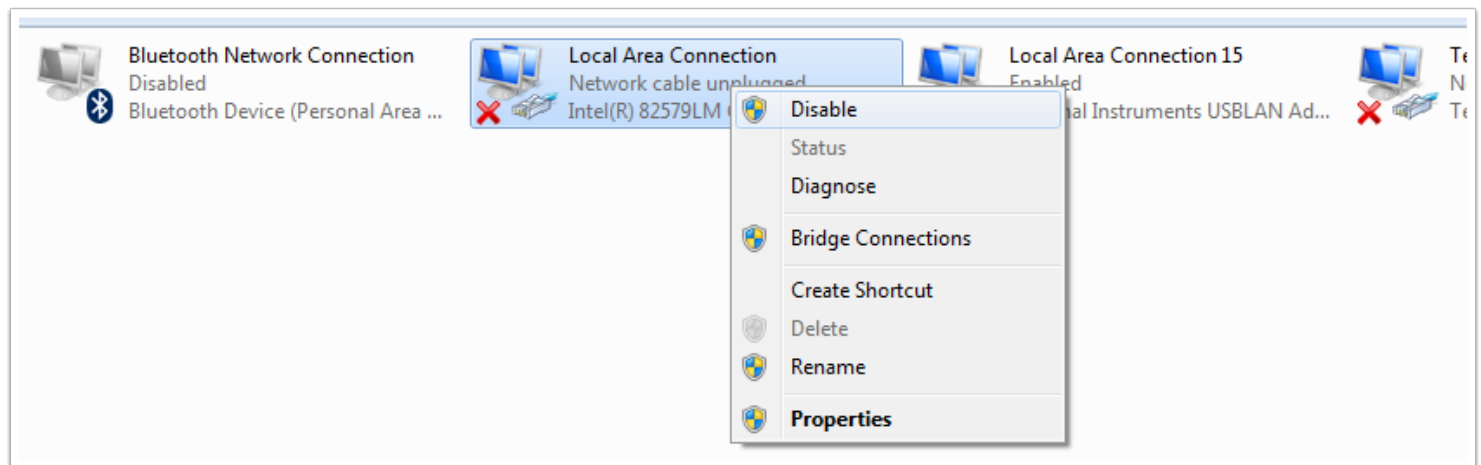
Choose the Network and Internet category.



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.

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

1. Confirm that battery voltage is displayed


Getting Started with the Control System

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

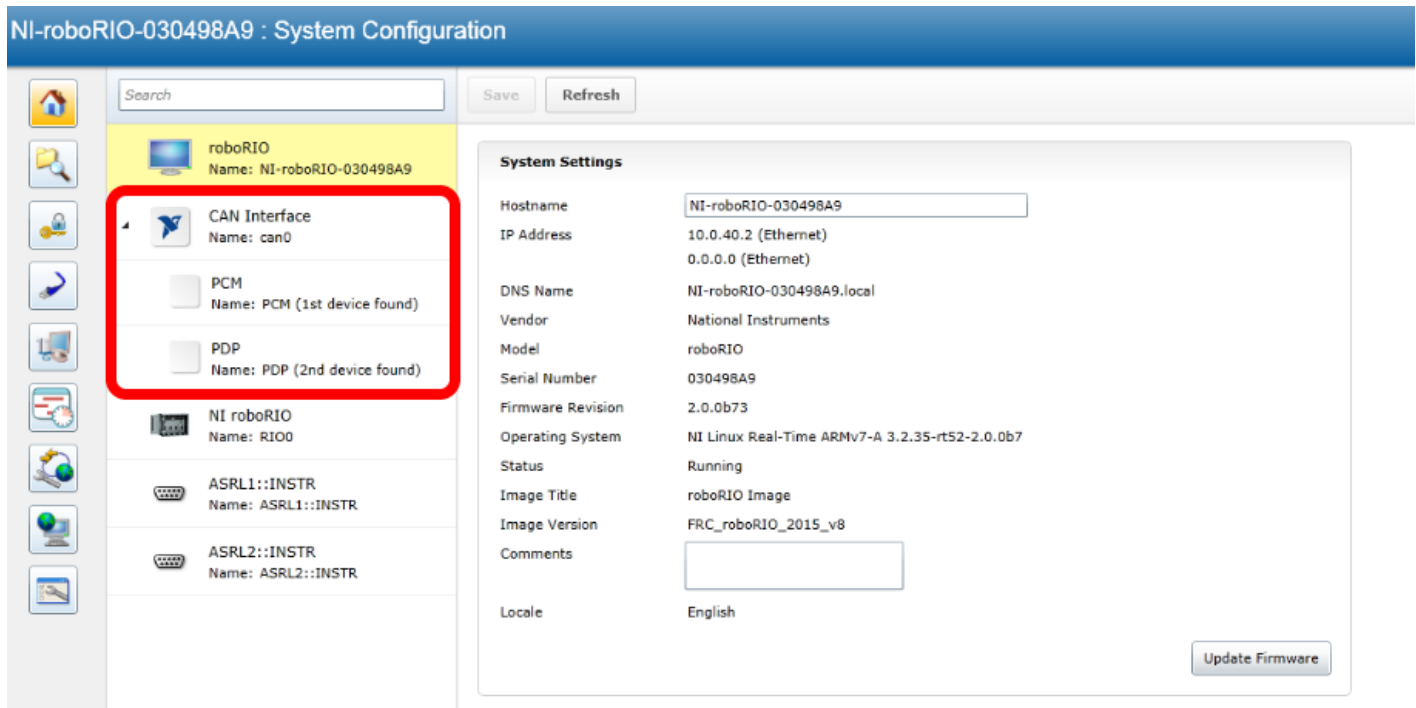
This document describes the process of updating the firmware on the Cross the Road Electronics CAN devices.

Note: Google Chrome is removing support for the Silverlight plugin. You will need to use a different browser such as Internet Explorer to access the roboRIO webdashboard.

 The CAN Webdashboard plugin is no longer included in the base roboRIO image. To enable CAN device functionality in the roboRIO WebDashboard (and proceed with this article) you must now install the plugin using CTRE's Phoenix Lifeboat. Instructions can be found on the [CTRE Phoenix Wiki](#).

Getting Started with the Control System

Accessing CAN Node Settings



Open the WebDash by using a browser to navigate to the roboRIO's address (172.22.11.2 for USB, or "roboRIO-####-FRC.local" where #### is your team number, with no leading zeroes, for either interface). You should see a page that looks like the image above, with the CAN devices listed out below the CAN Interface.

Note: The discovery order (e.g. "1st device found") is needed to separate devices of the same type but has no actual significance. You may see the PDP or a Jaguar or Talon SRX discovered first on your CAN network, even if the PCM is the first node in your CAN chain.

Troubleshooting

If you do not see any nodes below the CAN Interface entry try the following:

- Check the CAN cabling. If the LEDs on the PCM and PDP are red then they are not seeing CAN. Note that just because the LEDs on the devices are green does not mean the CAN cabling to the

Getting Started with the Control System

roboRIO are correct, they will turn green if the two other devices can see each other on the CAN network.

- Try refreshing the page. The device polling is done once every five seconds and the webpage itself doesn't always react to the Refresh button so if in doubt force a refresh by using the browser's refresh button or closing and re-opening the page.
- Make sure the CAN Interface is expanded. Double clicking the CAN Interface entry (or clicking the triangle to the left of the entry if present) will collapse the tree, repeating will expand it.
- Try restarting the browser. Occasionally the Silverlight plugin may crash or lock up resulting in the CAN devices silently not refreshing.

Settings

The screenshot displays the Control System interface. On the left is a sidebar with a tree view of devices. The 'CAN Interface' is expanded, showing a list of nodes. The 'PCM (1st device found)' node is selected and highlighted in yellow. Below it is the 'PDP (2nd device found)' node. Further down are 'NI roboRIO RIO0', 'ASRL1::INSTR', and 'ASRL2::INSTR'. The main area on the right is titled 'Settings' and displays the configuration for the selected PCM node. It includes fields for Name, Device ID, Software Status, Hardware Revision, Manufacture Date, Bootloader Revision, Vendor, Model, Firmware Revision, and Status. An 'Update Firmware' button is located at the bottom right of the settings panel.

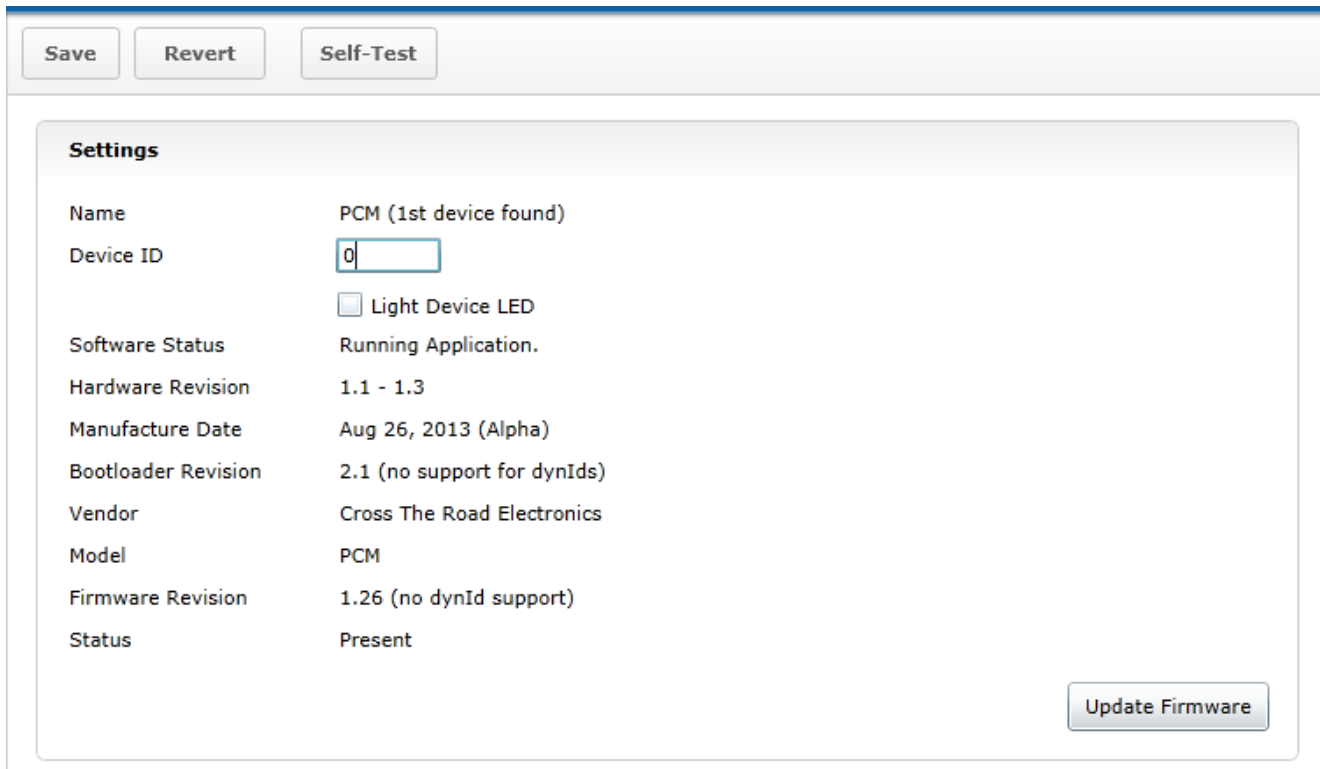
Settings	
Name	PCM (1st device found)
Device ID	<input type="text" value="0"/>
	<input type="checkbox"/> Light Device LED
Software Status	Running Application.
Hardware Revision	1.1 - 1.3
Manufacture Date	Aug 26, 2013 (Alpha)
Bootloader Revision	2.1 (no support for dynIds)
Vendor	Cross The Road Electronics
Model	PCM
Firmware Revision	1.26 (no dynId support)
Status	Present

[Update Firmware](#)

To access the Settings page of one of the CAN nodes, select the node by clicking on it's entry in the list. The settings for that node will then be displayed in the right pane.

Getting Started with the Control System

Setting CAN IDs



The screenshot shows a web-based control system interface. At the top, there are three buttons: "Save", "Revert", and "Self-Test". Below these is a "Settings" panel. Inside the panel, the "Name" is "PCM (1st device found)". The "Device ID" is set to "0" in a text input field. There is a checkbox for "Light Device LED" which is currently unchecked. Other settings include "Software Status" (Running Application), "Hardware Revision" (1.1 - 1.3), "Manufacture Date" (Aug 26, 2013 (Alpha)), "Bootloader Revision" (2.1 (no support for dynIds)), "Vendor" (Cross The Road Electronics), "Model" (PCM), "Firmware Revision" (1.26 (no dynId support)), and "Status" (Present). An "Update Firmware" button is located at the bottom right of the settings panel.

Settings	
Name	PCM (1st device found)
Device ID	0
	<input type="checkbox"/> Light Device LED
Software Status	Running Application.
Hardware Revision	1.1 - 1.3
Manufacture Date	Aug 26, 2013 (Alpha)
Bootloader Revision	2.1 (no support for dynIds)
Vendor	Cross The Road Electronics
Model	PCM
Firmware Revision	1.26 (no dynId support)
Status	Present

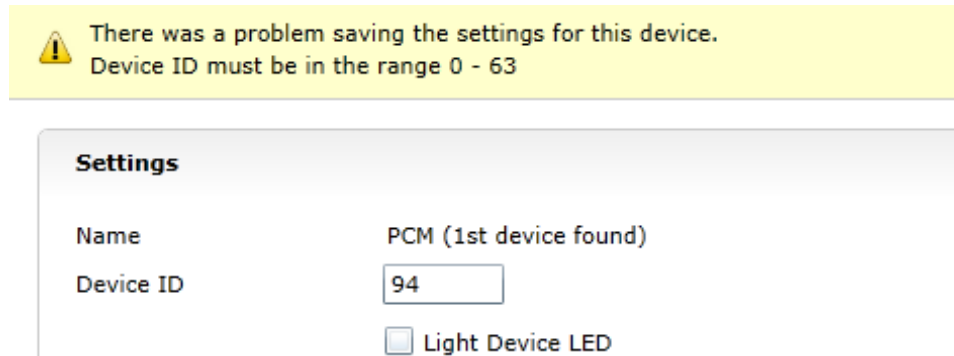
Update Firmware

Each device comes with the CAN ID set to a default value of 0. If using only a single device of that type it is recommended to leave the ID at the default value to allow for the use of default Opens/Constructors. If using multiples of a particular device type (I.E. 2 PCMs or 4 Talon SRXs) you will need to change the node ID of all but one device. To change the node ID:

- Highlight/Select the Device ID and replace it with your desired ID.
- Press "Save". The "Save" button will depress and the "Refresh" button will appear.
- The PDP, PCM and Talon SRX require no additional action to save the new ID. For CAN Jaguars, a notice will appear instructing you to push the user button within 5 seconds. After doing so, click Refresh and verify that the new Device ID has been set.

Getting Started with the Control System

ID Ranges



The screenshot shows a web interface with a yellow error banner at the top. The banner contains a warning icon and the text: "There was a problem saving the settings for this device. Device ID must be in the range 0 - 63". Below the banner is a "Settings" panel. Inside the panel, the "Name" field is set to "PCM (1st device found)". The "Device ID" field is a text input containing the number "94". Below the "Device ID" field is a checkbox labeled "Light Device LED", which is currently unchecked.

The valid ID ranges for each type of device are:

- Pneumatics Control Module (PCM) ID - 0 to 62 (inclusive)
- Power Distribution Panel (PDP) ID - 0 to 62 (inclusive)
- Jaguar ID- 1 to 63 (inclusive)
- Talon SRX ID- 0 to 62 (inclusive)

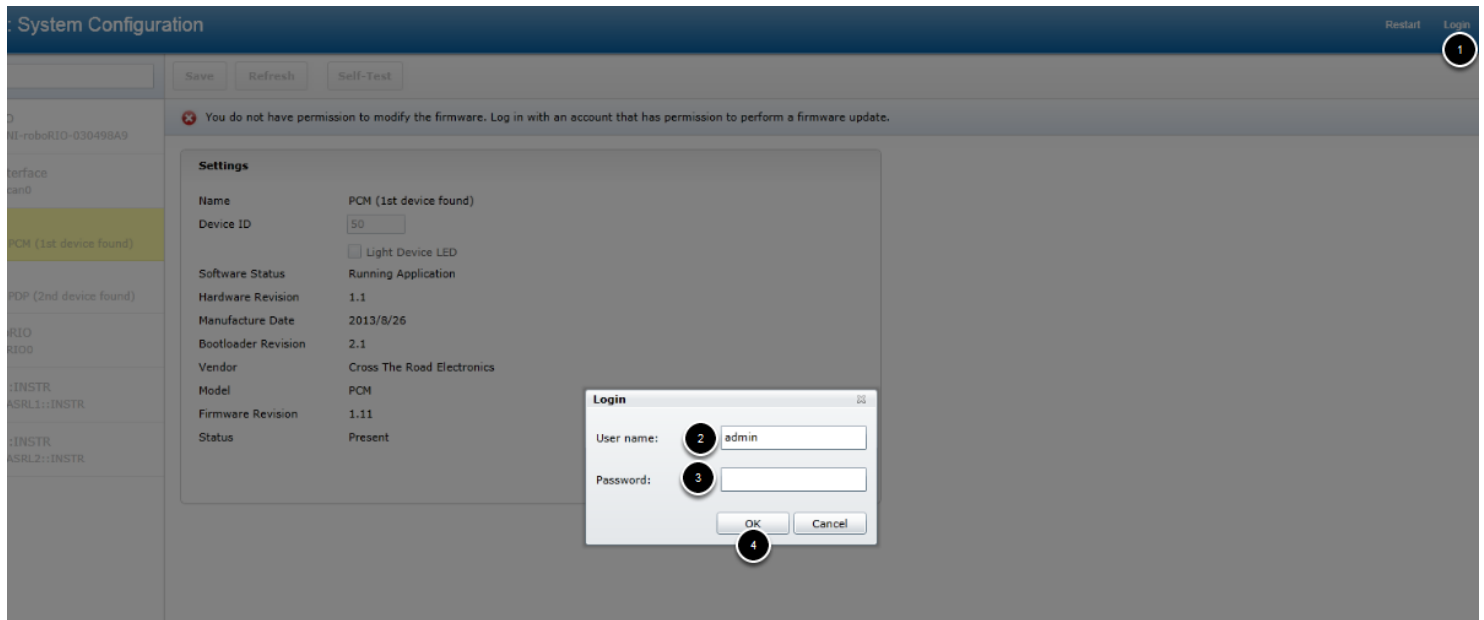
Since the ID ranges for different products don't overlap there is no issue with two or more CAN nodes of different types having the same Device ID (e.g. a PDP with ID=0, a PCM with ID=0, and a Talon SRX with ID=0 on the same bus). Using multiple devices of the same type, such as multiple PCMs or multiple Jaguars with the same node ID will result in a conflict. The web plugin supports a strategy that will allow for recovery of this condition for all devices other than Jaguars, but the devices are not properly usable from within a robot program while in this state. To recover Jaguars which have been set to the same ID you will have to remove all but one of the devices from the bus, then set the devices to non-conflicting IDs.

If you select an invalid ID you will get an immediate prompt like the one shown above.

Changing the PDP ID while using C++\Java WPILib is not recommended as there is no way to change the desired node ID in the library. PCM node IDs may be set as desired and addressed using the appropriate Open or Constructor of the Solenoid or Double Solenoid class.

Getting Started with the Control System

Updating CAN Node Firmware

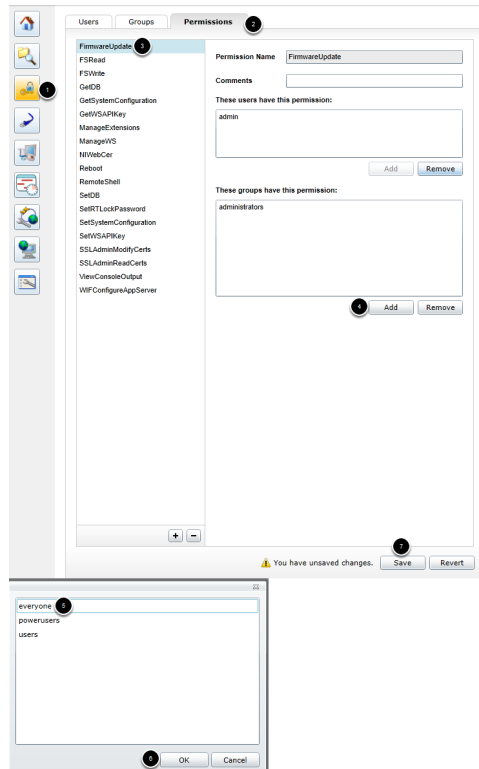


This page can also be used to update the device firmware. To load new firmware you must be logged in:

1. Click "Login" at the top right of the page.
2. Enter the User Name "admin"
3. Leave the Password field blank.
4. Click Ok.

Getting Started with the Control System

Updating Permissions



If you would like to skip the Login step in the future you can set up Permissions to allow firmware updates:


1. Click the Lock Icon in the far left pane.
2. Click the Permissions tab.
3. Select Firmware Update from the list.
4. Click Add below the second large box.
5. Select "everyone"
6. Click Ok.
7. Click Save.

Getting Started with the Control System

Update Firmware

Settings

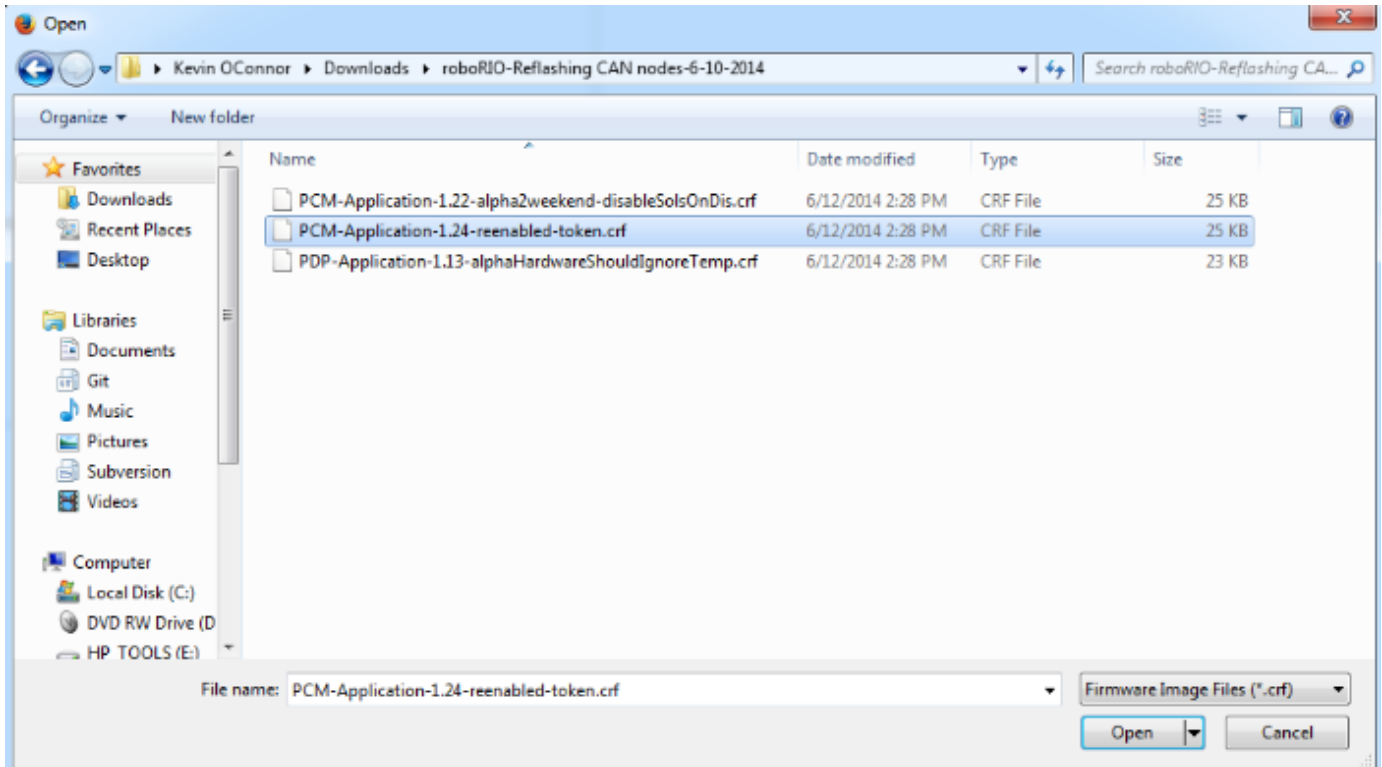
Name	PCM (1st device found)
Device ID	<input type="text" value="0"/>
	<input type="checkbox"/> Light Device LED
Software Status	Running Application.
Hardware Revision	1.1 - 1.3
Manufacture Date	Aug 26, 2013 (Alpha)
Bootloader Revision	2.1 (no support for dynIds)
Vendor	Cross The Road Electronics
Model	PCM
Firmware Revision	1.26 (no dynId support)
Status	Present



The firmware on a CAN Node is updated from the Setting's page for that node. To update the firmware of a CAN Node, press the Update Firmware button.

Getting Started with the Control System

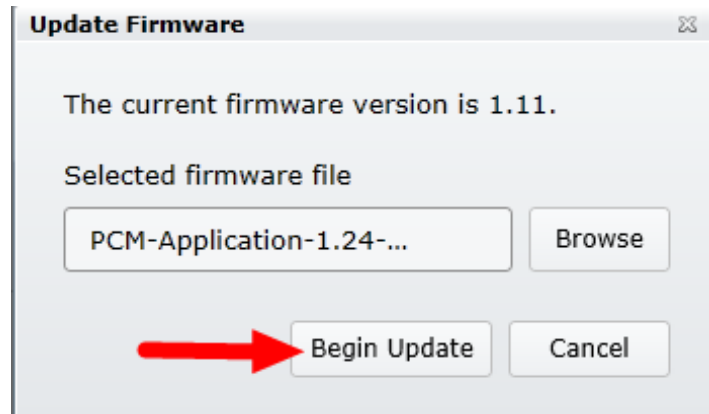
Select New Firmware



CTRE Devices use a file format call CRF (Cross The Road Firmware). Using the dialog, browse to the correct location on your computer and select the new firmware file, then click Open. Firmware for CTRE devices can be found in the C:\Users\Public\Public Documents\FRC folder.

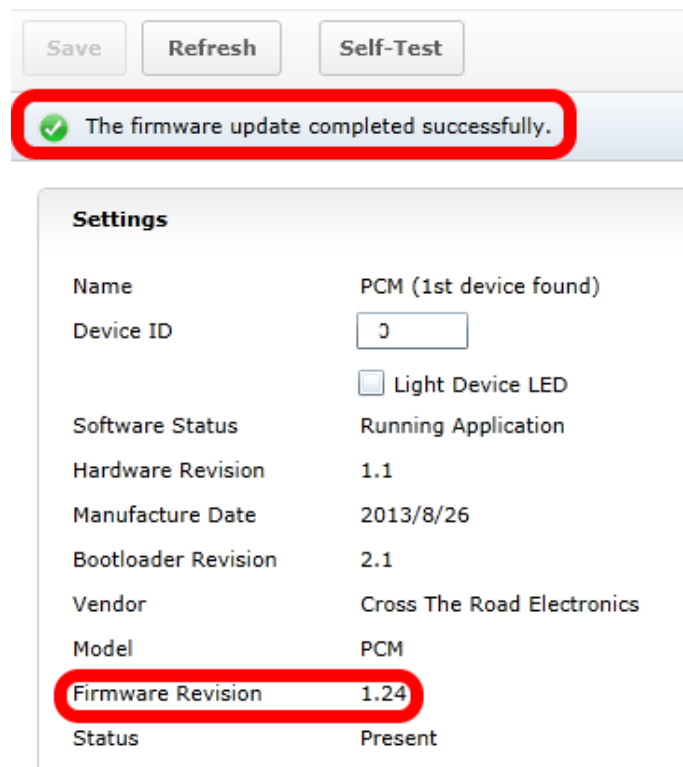
Getting Started with the Control System

Confirmation



On the dialog that appears, click Begin Update.

Update Complete



Getting Started with the Control System

If the update completes successfully, you should see a confirmation message near the top of the page and the Firmware Revision should update to match the new file.

Troubleshooting

There was a problem updating the firmware for this device.
PCM (1st device found) : CTRE_DI_CouldNotErase

Settings

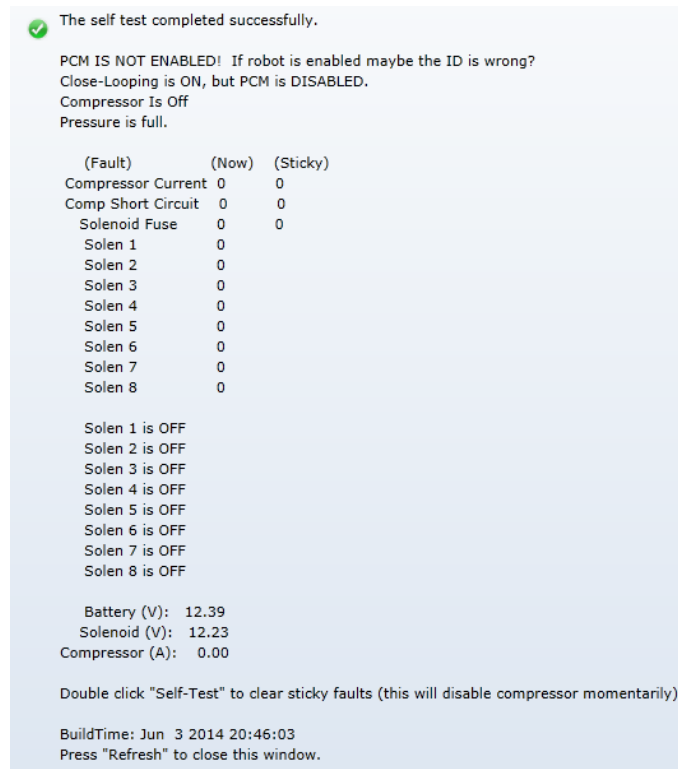
Name	PCM (1st device found)
Device ID	<input type="text" value="0"/>
	<input type="checkbox"/> Light Device LED
Software Status	Bootloader, LED is blinking green/orange.
Hardware Revision	1.1
Manufacture Date	2013/8/26
Bootloader Revision	2.1
Vendor	Cross The Road Electronics
Model	PCM
Firmware Revision	255.255
Status	Present

Update Firmware

Since ten seconds is plenty of time for power/CAN to be disconnected, an error code will be reported if a reflash is interrupted or fails. Additionally the Software Status will report “Bootloader” and Firmware Revision will be 255.255 (blank). If a CAN Device has no firmware, it’s bootloader will take over and blink green/yellow on the device’s corresponding LED. It will also keep it’s device ID, so the RIO can still be used to set device ID and reflash the application firmware (crf). This means you can reflash again using the same web interface (there is no need for a recovery button).

Getting Started with the Control System

Self-Test

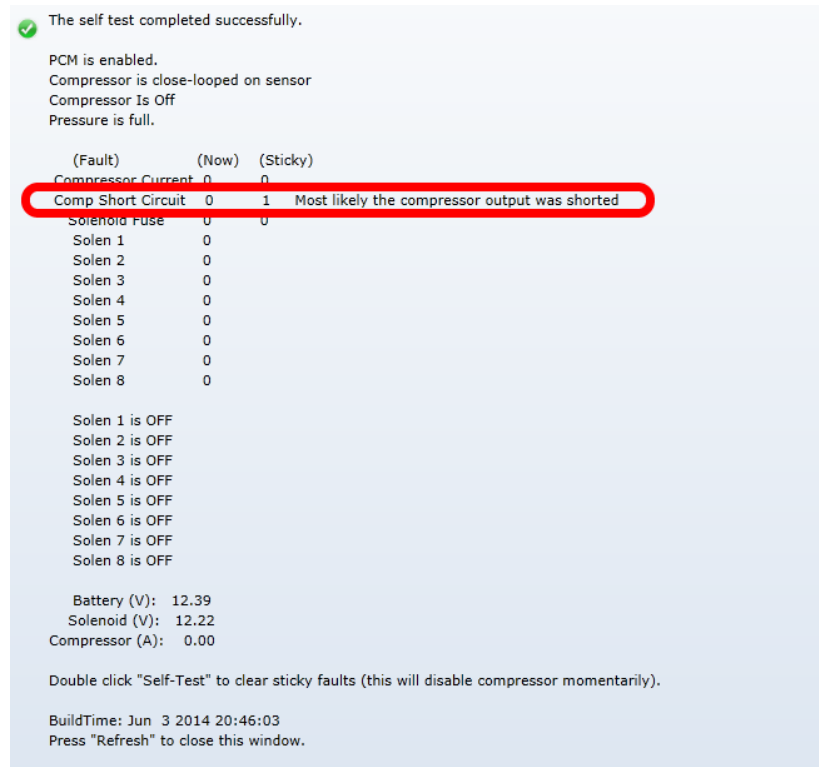


Pressing Self Test will display data captured from CAN Bus at time of press. This can include fault states, sensor inputs, output states, measured battery voltage, etc...

At the bottom of the section, the build time is displayed for checking what firmware revision is installed. The image above is an example of pressing "SelfTest" with PCM. Be sure to check if PCM is ENABLED or DISABLED. If PCM is DISABLED then either the robot is disabled or team code is talking to the wrong PCM device ID (or not talking to the PCM at all).

Getting Started with the Control System

Sticky Faults



After enabling the robot and repressing "SelfTest" we see the PCM is enabled but an intermittent short on the compressor output reveals itself in a sticky fault.

Sticky faults persist across power cycles. They also cause orange blinks on the device LED. The PCM will orange blink to signal a sticky fault only when the robot is disabled. The PDP will orange blink anytime it sees a sticky fault (since PDPs are not output devices they don't care if robot is enabled or not).

Getting Started with the Control System


Clearing Sticky Faults



To clear Sticky Faults, double click Self Test in a rapid fashion. If the faults don't clear you may need to triple click, or rapidly click until you see the "Faults cleared!" text appear.

Getting Started with the Control System

PDP Self-Test



The self test completed successfully.

Channel 1 (A): < 1.73
Channel 2 (A): < 1.73
Channel 3 (A): < 1.73
Channel 4 (A): < 1.73
Channel 5 (A): < 1.73
Channel 6 (A): < 1.73
Channel 7 (A): < 1.73
Channel 8 (A): < 1.73
Channel 9 (A): < 1.73
Channel 10 (A): < 1.73
Channel 11 (A): < 1.73
Channel 12 (A): < 1.73
Channel 13 (A): < 1.73
Channel 14 (A): < 1.73
Channel 15 (A): < 1.73
Channel 16 (A): < 1.73
Battery(V) : 13.52
Temp(C) : 98.09
FaultHardwareStatus : 0
Current FAULTS : 0000000000000000
FaultTemp : 0
FaultVbat : 0
Current FAULTS : 0000000000000000 (sticky)
StickyFaultTemp : 0
StickyFaultVbat : 0

Double click "Self-Test" to clear sticky faults.

BuildTime: Jun 3 2014 20:46:03
Press "Refresh" to close this window.

Here's an example for PDP. Notice here this PDP sees a temperature of 98.09C (don't worry this board does not have the temp sensor populated). With this firmware, no temp fault is recorded because this hardware revision does not have the temp sensor populated.

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!](#)

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.

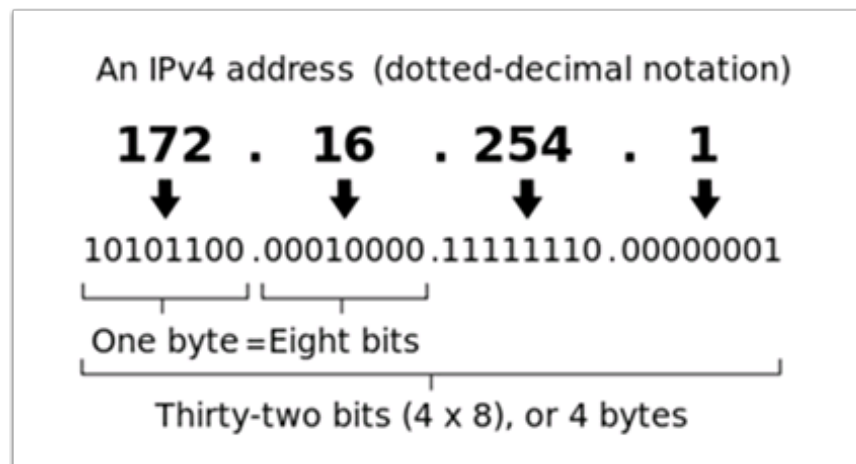
IP Addressing will be the core topic of this training, and the key for understanding how devices are communicating. By understanding how to interpret IP addresses, how they are assigned, how they are organized, and how to set them, teams should be able to configure their devices to successfully communicate.

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 4,000,000,000 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.

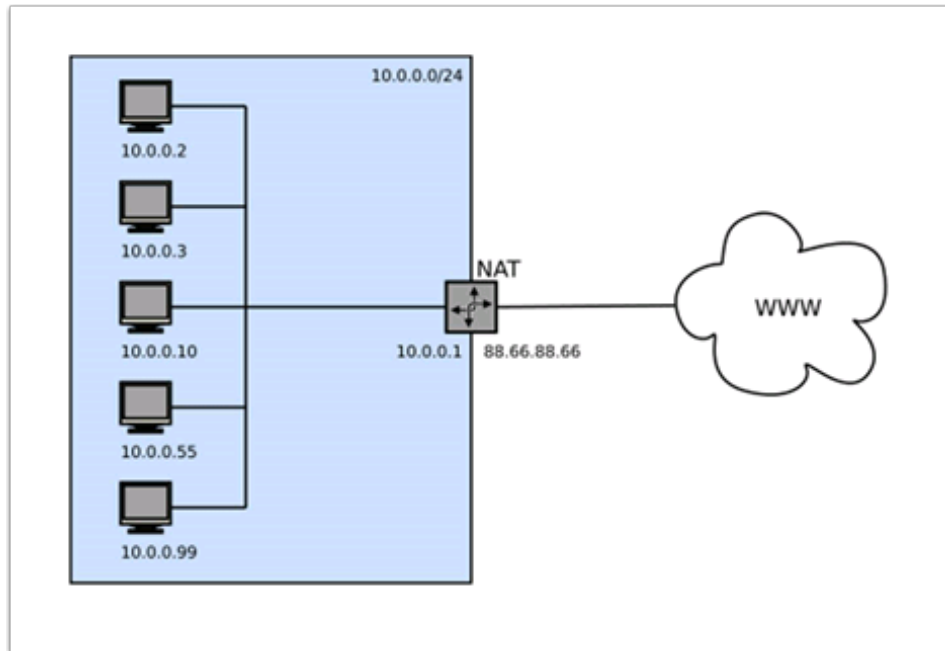
Public vs Private IP Addresses

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

Getting Started with the Control System

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.

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

Getting Started with the Control System

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 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.

Getting Started with the Control System

Addresses to avoid:

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

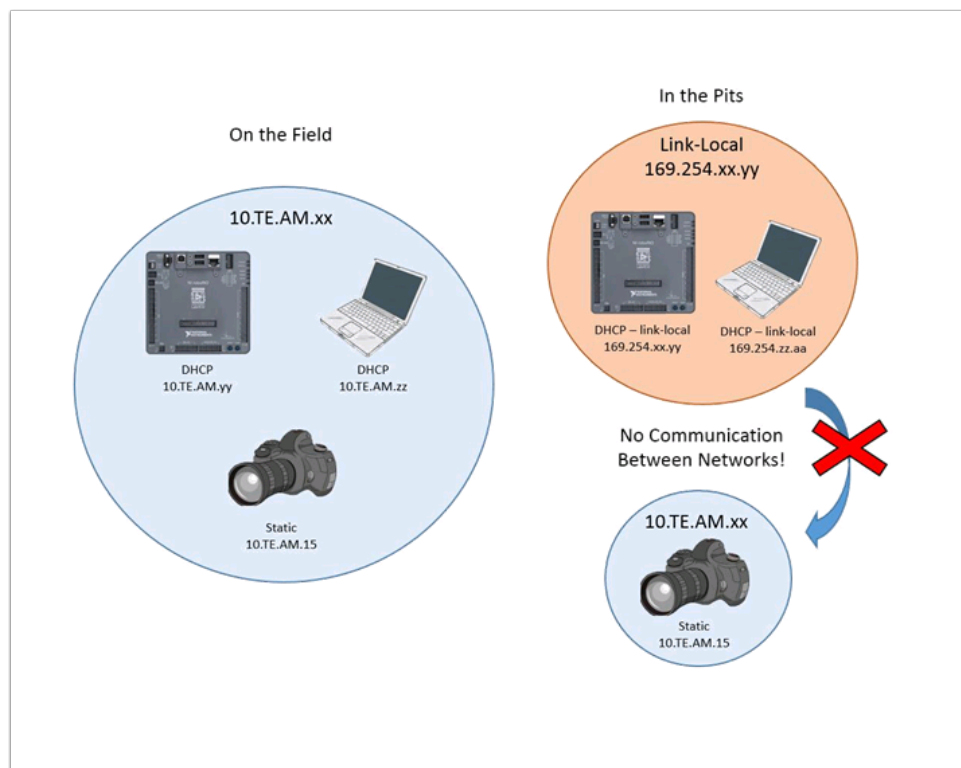
In a static addressing configuration, the configuration should be the same in the pits as it is on the field. Any programming computers will need to have an IP Address set in the 10.TE.AM.xx range with a subnet of 255.255.255.0

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. 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 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. 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.

2018 Game Data Details

In the 2018 game, *FIRST*[®] POWER UPSM, 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

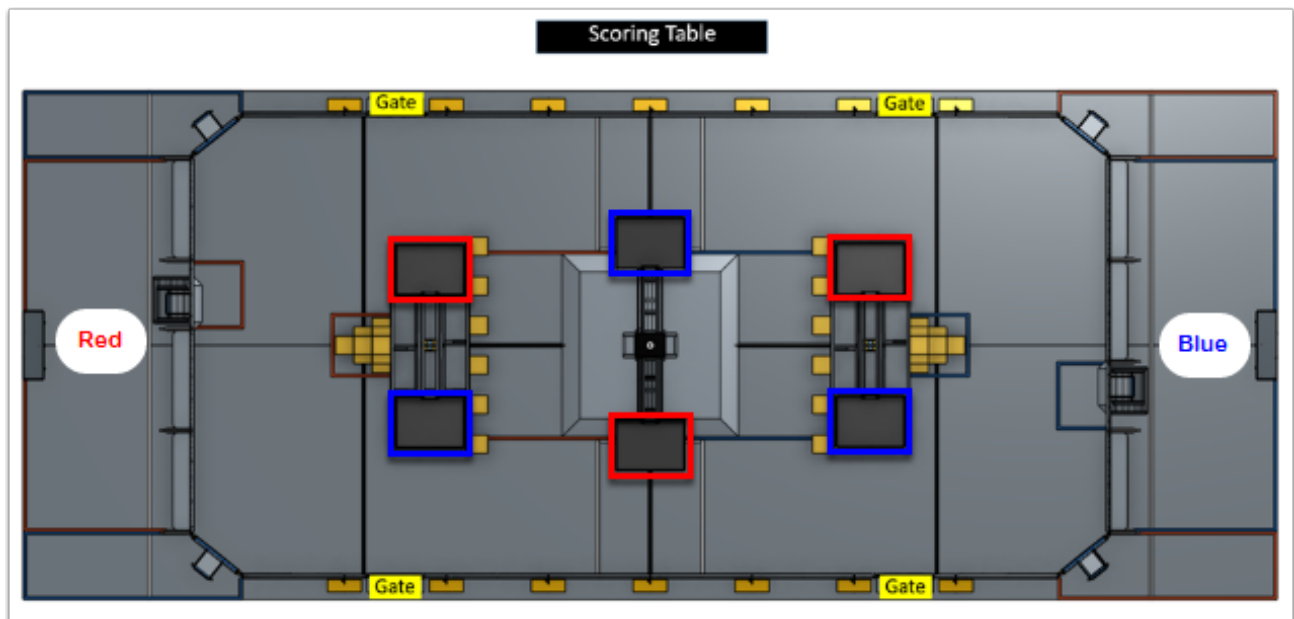
The initial state of the Game Data returned from the Robot code is non-deterministic. Before the data for the current match is sent the data returned will depend on what data is currently stored in the DS. This may be data from the last match, data from testing using the text box in the DS, or empty. 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

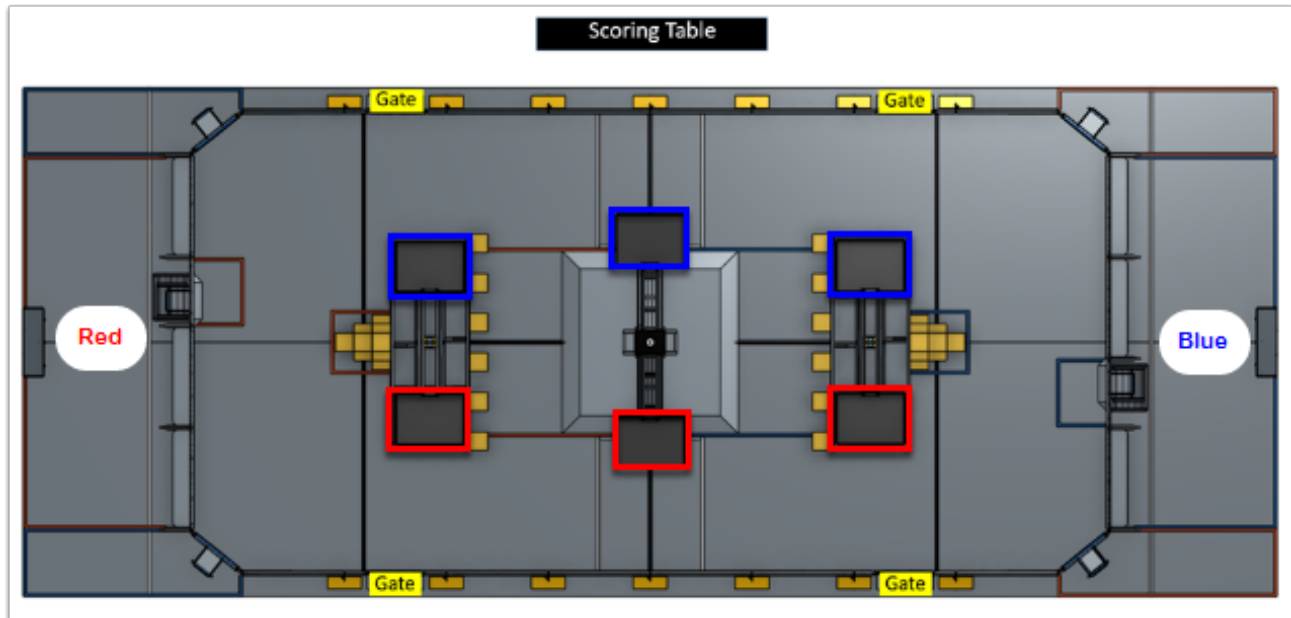


Red Alliance: "LRL"

Blue Alliance: "LRL"

Getting Started with the Control System

Example 2



Red Alliance: "RRR"

Blue Alliance: "RRR"

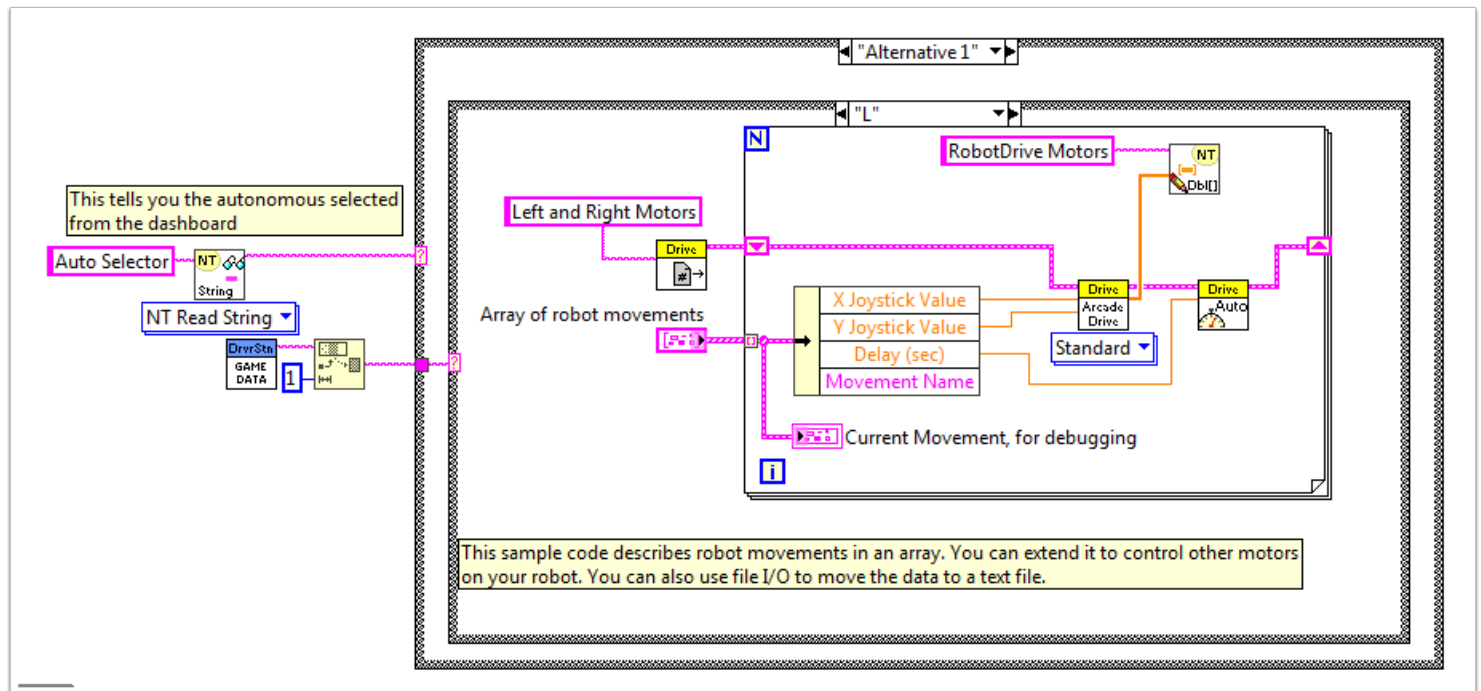
LabVIEW

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

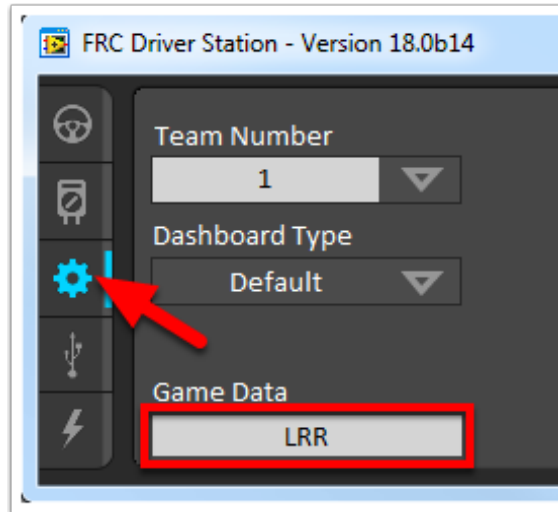
The Game Data in Java is accessed using the `getGameSpecificMessage` method in the `DriverStation` class. If you are using the `Iterative` or `Timed Robot` classes (or `Command Based`, which is based on one of those) you can query the Game Data in the `AutonomousInit()` method. If you are using `SimpleRobot`,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