Troubleshooting

Last Updated: 03-04-2017

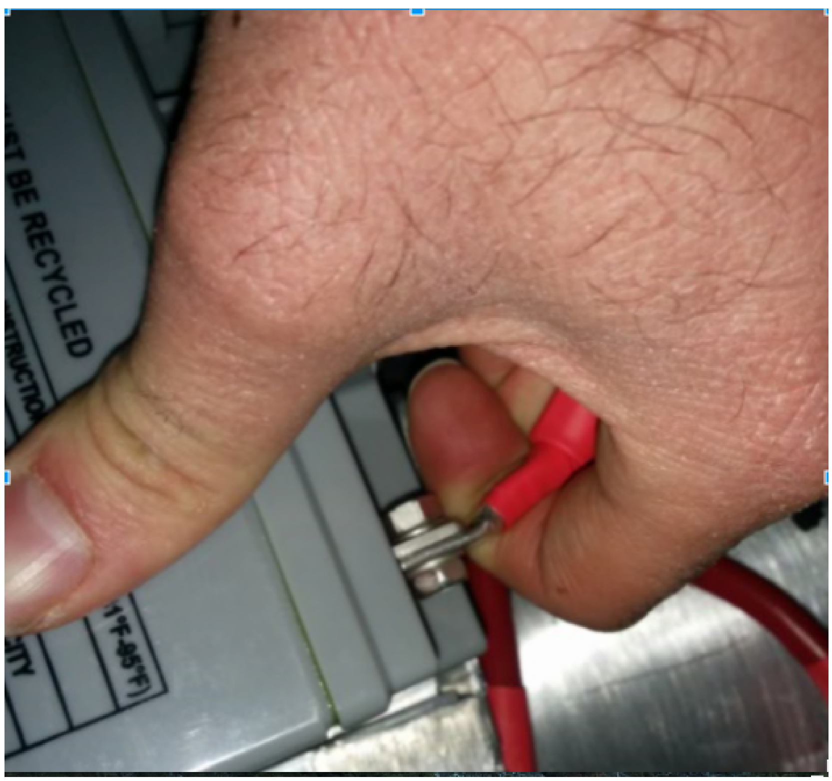
Table of Contents

General Troubleshooting

Robot Preemptive Troubleshooting

In FIRST Robotics Competion, robots take a lot of stress while driving around the field. It is important to make sure that connections are tight, parts are bolted securely in place and that everthing is mounted so that a robot bouncing around the field does not break.

Check battery connections

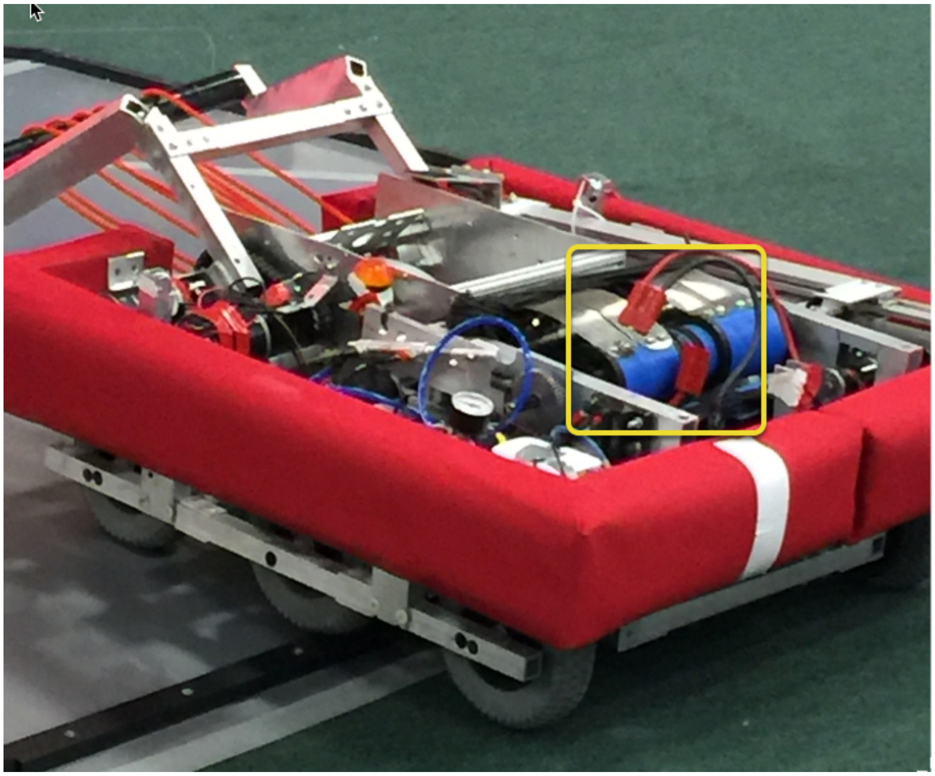


The tape the should be covering the battery connection in these examples has been removed to illustrate what is going on. On your robots, the connections should be covered.

Wiggle battery harness connector. Often these are loose because the screws loosen, or sometimes the crimp is not completely closed. You will only catch the really bad ones though because often the electrical tape stiffens the connection to a point where it feels stiff. Using a voltmeter or Battery Beak will help with this.

Apply considerable force onto the battery cable at 90 degrees to try to move the direction of the cable leaving the battery, if successful the connection was not tight enough to begin with and it should be redone.

Secure the battery to robot connection



In almost every event we see at least one robot where a not properly secured battery connector (the large Anderson) comes apart and disconnects power from the robot. This has happened in championship matches on the Einstein and everywhere else. Its an easy to ensure that this doesn't happen to you by securing the two connectors by wrapping a tie wrap around the connection. 10 or 12 tie wraps for the piece of mind during an event is not a high price to pay to guarantee that you will not have the problem of this robot from an actual event after a bumpy ride over a defense.

120 Amp circuit breaker

Apply a twisting force onto the cable to rotate the harness. If you are successful then the screw is not tight enough. Split washers might help here, but in the mean time, these require checking every few matches.

Because the metal is just molded into the case, every once in awhile you will break off the bolt, ask any veteran team and they'll tell you they go through a number of these every few seasons. After tightening the nut, retest by once again trying to twist the cable.

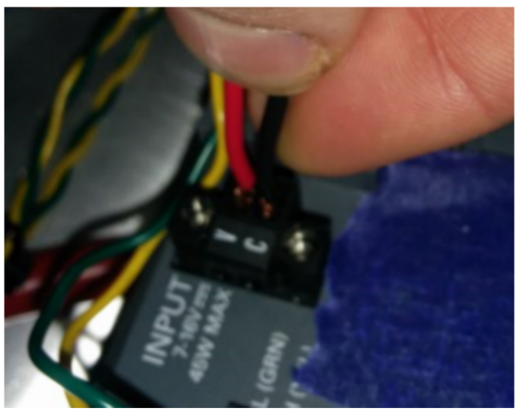
Troubleshooting

Power Distribution Panel (PDP)

Just by removing the battery cover, often you can confirm the washer.

Make sure that split washers were placed under the PDP screws, but it is not easy to visually confirm, and sometimes you can’t. You can check by removing the case.   Also if you squeeze the red and black wires together, sometimes you can catch the really lose connections.

Tug test everything



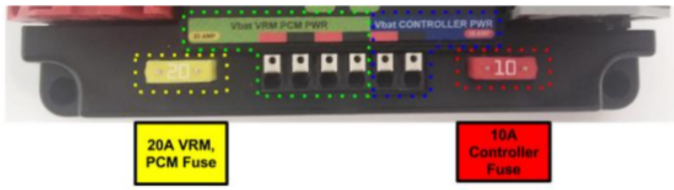
The **Weidmuller** contacts for power, compressor output, **roboRIO power connector**, and **radio power** are important to verify by tugging on the connections as shown. Make sure that none of the connections pull out.

Look for possible or impending shorts with Weidmuller connections that are close to each other, and have too-long wire-lead lengths (wires that are stripped extra long).

Spade connectors can also fail due to improper crimps, so tug-test those as well.

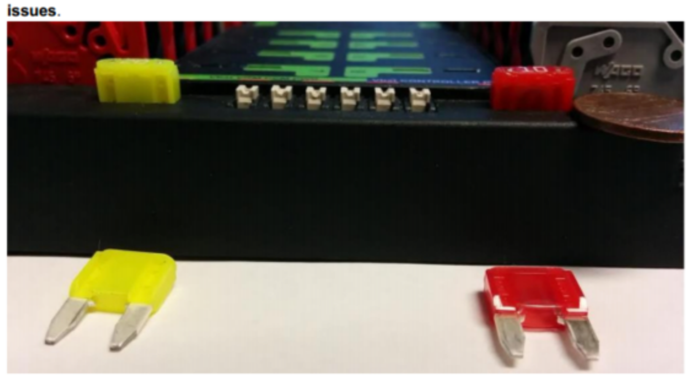
Blade fuses

2.4. Blade Fuse



Be sure to place the 20A fuse (yellow) on the left and the 10A fuse (red) on the right.

Warning: Also take care to ensure **fuses are fully seated** into the fuse holders. The fuses should descend **at least** as far as the figure below (different brand fuses have different lead lengths). It should be **nearly impossible to remove the fuse** with bare hands (without the use of pliers). If this is not properly done, the robot/radio may exhibit **intermittent connectivity issues**.



If you can remove the blade fuses by hand then they are not in completely. Make sure that they are completely seated in the PDP so that they don't pop out during robot operation.

RoboRIO swarf

Swarf is: fine chips or filings of stone, metal, or other material produced by a machining operation. Often modifications must be made to a robot while the control system parts are in place. The circuit board for the roboRIO is conformally coated, but that doesn't absolutely guarantee that metal chips won't short out traces or components inside the case. In this case, you must exercise care in making sure that none of the chips end up in the roboRIO or any of the other components. In particular, the exposed 3 pin headers are a place where chips can enter the case. A quick sweep through each of the four sides with a flashlight is usually sufficient to find the really bad areas of infiltration.

Radio barrel jack

Make sure the correct barrel jack is used, not one that is too small and falls out for no reason. This isn’t common, but ask an FTA and every once in awhile a team will use some random barrel jack that is not sized correctly, and it falls out in a match on first contact.

Ethernet cable

If the RIO to radio ethernet cable is missing the clip that locks the connector in, get another cable. This is a common problem that will happen several times in every competition. Make sure that your cables are secure. The clip often breaks off, especially when pulling it through a tight path, it snags on something then breaks.

Cable slack

Cables must be tightened down, particularly the radio power and ethernet cable. The radio power cables don’t have a lot of friction force and will fall out (even if it is the correct barrel) if the weight of the cable-slack is allowed to swing freely.

Ethernet cable is also pretty heavy, if it’s allowed to swing freely, the plastic clip may not be enough to hold the ethernet pin connectors in circuit.

Reproducing problems in the pit

Beyond the normal shaking and rattling of all cables while the robot is power and tethered, you might try picking up one side of the robot off the ground and drop it, and see if you lose connection. The driving on the field, especially when trying to breach defenses will often be very violent. It's better to see it fail in the pit rather than in a critical match.

When doing this test it’s important to be ethernet tethered and not USB tethered, otherwise you are not testing all of the critical paths.

Check firmware and versions

Robot inspectors do this, but you should do it as well, it helps robot inspectors out and they appreciate it. And it guarantees that you are running with the most recent, bug fixed code. You wouldn't want to lose a match because of an out of date piece of control system software on your robot.

Troubleshooting

Driver station checks

We often see problems with the Drivers Station. You should:

- ALWAYS bring the laptop power cable to the field, it doesn't matter how good the battery is, you are allowed to plug in at the field.
- Check the power and sleep settings, turn off sleep and hybernate, screen savers, etc.
- Turn off power management for USB devices (dev manager)
- Turn off power management for ethernet ports (dev manager)
- Turn off windows defender
- Turn off firewall
- Close all apps except for DS/Dashboard when out on the field.
- Verify that there is nothing unnecessary running in the application tray in the start menu (bottom right side)

Handy tools

There never seems to be enough light inside robots, at least not enough to scrutinize the critical connection points, so consider using a handheld LED flashlight to inspect the connections on your robot. They're available from home depot or any hardware/automotive store.

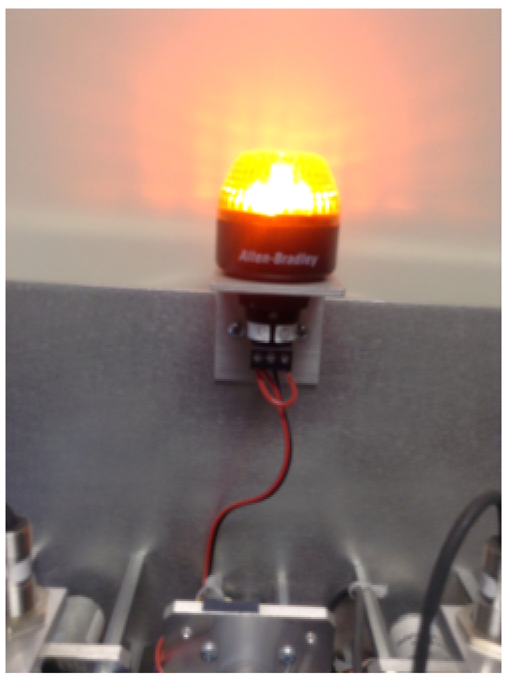
Troubleshooting

Wago tool is nice to for redoing weidmuller connections with stranded wires. Often I'll do one to show the team, and then have them do the rest using the WAGO tool to press down the white-plunger while they insert the stranded wire. The angle of the WAGO tool makes this particularly helpful.

Status Light Quick Reference

Many of the components of the FRC Control System have indicator lights that can be used to quickly diagnose problems with your robot. This guide shows each of the hardware components and describes the meaning of the indicators. Photos and information from Innovation FIRST and Cross the Road Electronics.

Robot Signal Light (RSL)



- Solid ON - Robot On and Disabled
- Blinking - Robot On and Enabled
- Off - Robot Off, roboRIO not powered or RSL not wired properly.

Troubleshooting

RoboRIO

Power

- Green - Power is good
- Amber - Brownout protection tripped, outputs disabled
- Red - Power fault, check user rails for short circuit

Status

- On while the controller is booting, then should turn off
- 2 blinks - Software error, reimage roboRIO
- 3 blinks - Safe Mode, restart roboRIO, reimage if not resolved
- 4 blinks - Software crashed twice without rebooting, reboot roboRIO, reimage if not resolved
- Constant flash or stays solid on - Unrecoverable error

Radio

Not currently implemented

Troubleshooting

Comm

- Off - No Communication
- Red Solid - Communication with DS, but no user code
- Red Blinking - E-stop
- Green Solid - Good communication with DS

Mode

- Off - Outputs disabled (robot in Disabled, brown-out, etc.)
- Amber/Orange - Autonomous Enabled
- Green - Teleop Enabled
- Red - Test Enabled

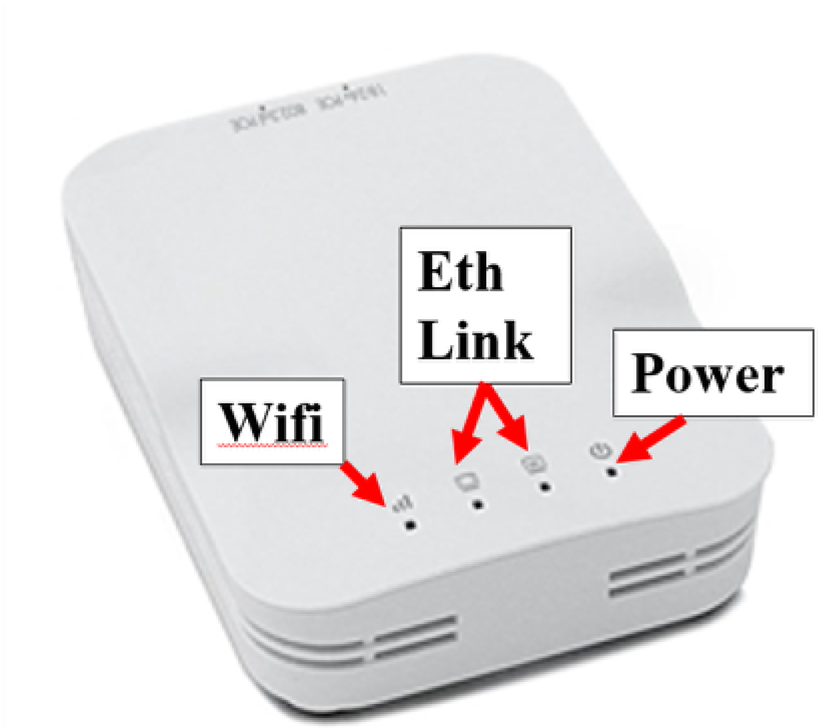
RSL

See above

OpenMesh Radio

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

WiFi light only works after radio has been power cycled.



Troubleshooting

Power

- Blue - On or Powering Up
- Blue Blinking - Powering Up

Eth Link

- Blue - Link Up
- Blue Blinking - Link Up + Traffic Present

WiFi

- Off - Bridge Mode Unlinked or Non-FRC Firmware
- Red - AP Mode Unlinked
- Yellow\Orange - AP Mode Linked
- Green - Bridge Mode Linked

Power Distribution Panel

LED Fault Table

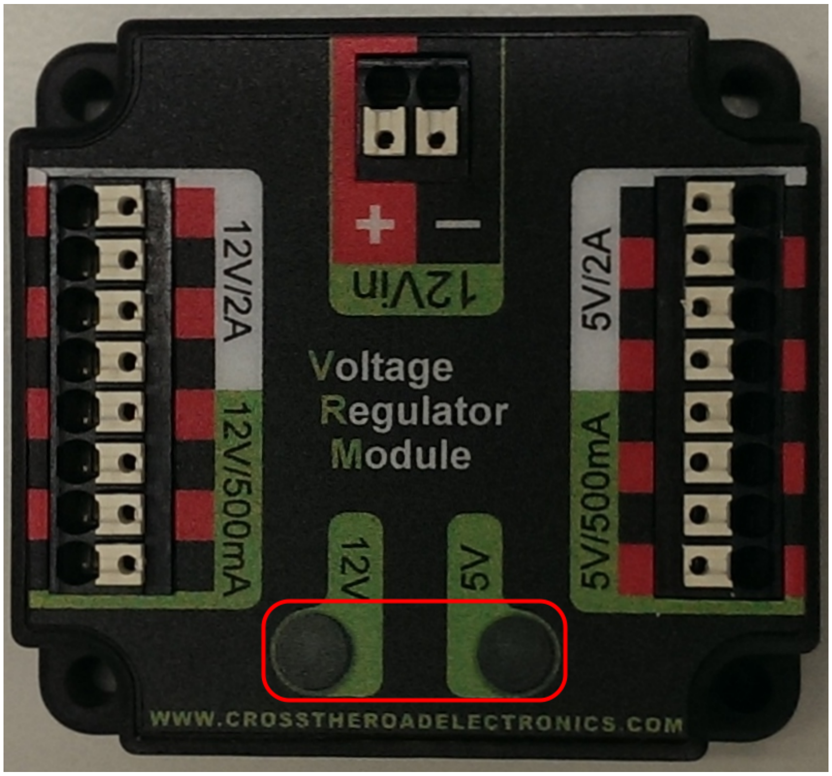
LED	Strobe	Slow	Long
Green	No Fault - Robot Enabled	No Fault - Robot Disabled	NA
Orange	NA	Sticky Fault	NA
Red	NA	No CAN Comm	NA

*If PCM LED contains more than one color, see LED Special States Table

LED Special States Table

LED Colors	Problem
Red/ Orange	Damaged Hardware
Green/ Orange	In Bootloader
No LED	No Power / Incorrect Polarity

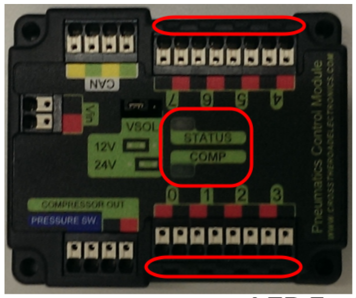
Voltage Regulator Module



The status LEDs on the VRM indicate the state of the two power supplies. If the supply is functioning properly the LED should be lit bright green. If the LED is not lit or is dim, the output may be shorted or drawing too much current.

Troubleshooting

Pneumatics Control Module



LED Fault Table

LED	Strobe	Slow	Long
Green	No Fault - Robot Enabled	No Fault - Robot Disabled	NA
Orange	NA	Sticky Fault	NA
Red	NA	No CAN Comm OR Solenoid Fault (Blinks Solenoid Index)	Compressor Fault

*If PCM LED contains more than one color, see LED Special States Table

LED Special States Table

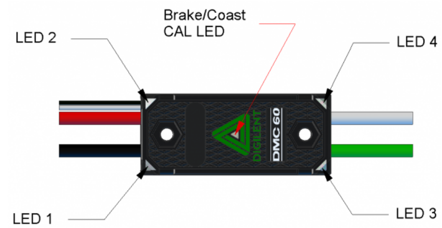
LED Colors	Problem
Red/ Orange	Damaged Hardware
Green/ Orange	In Bootloader
No LED	No Power / Incorrect Polarity

Solenoid Channel LEDs - These LEDs are lit red if the Solenoid channel is enabled and not lit if it is disabled.

Comp - This is the Compressor LED. This LED is green when the compressor output is active (compressor is currently on) and off when the compressor output is not active.

Status - The status LED indicates device status as indicated by the two tables above. For more information on resolving PCM faults see the PCM User Manual. Note that the No CAN Comm fault will not occur only if the device cannot see communicate with any other device, if the PCM and PDP can communicate with each other, but not the roboRIO you will NOT see a No Can Comm fault.

Digilent DMC-60



At power-on the RGB LEDs will display a progressive blue color, which continually gets brighter. This lasts for approximately five seconds. During this time the motor controller will not respond to an input signal, nor will the output drivers be enabled. After the initial power-on has completed the device will begin normal operation and what gets displayed on the RGB LEDs will be a function of the input signal being applied, as well as the current fault state. Assuming that no faults have occurred the RGB LEDs will function as follows:

Servo Input Signal Applied	LED State
No input signal or invalid input pulse width	Alternate between top (LED1 and LED2) and bottom (LED3 and LED4) LEDs being on and off. When on, the LEDs display color is orange.
Neutral input pulse width	All 4 LEDs on solid orange
Positive input pulse width	LEDs blink green in a clockwise circular pattern (LED1→LED2→LED3→LED4→LED1). The rate at which the LEDs update is proportional to the duty cycle of the output and increases with increased duty cycle. At 100% duty cycle, all four LEDs turn on solid green.
Negative input pulse width	LEDs blink red in a counter-clockwise circular pattern (LED1→LED4→LED3→LED2→LED1). The rate at which the LEDs update is proportional to the duty cycle of the output and increases with increased duty cycle. At 100% duty cycle, all four LEDs turn on solid red.

9 Fault Indicators

When a fault condition is detected the output duty cycle is reduced to 0% and a fault is signaled. The output will remain disabled for 3 seconds. During this time the onboard LEDs (LED1, LED2, LED3, and LED4) are used to indicate the fault condition. The fault condition is indicated by toggling between the top (LED1 and LED2) and bottom (LED3 and LED4) LEDs being on and off. The top LEDs will be Red during them on state. The color of the bottom LEDs depends on which faults are presently active. The table below describes how the color of the bottom LEDs maps to the presently active faults.

Color	Over Temperature	Under Voltage
Green	✓	X
Blue	X	✓
Cyan/Aqua	✓	✓

When the center LED is off the device is operating in coast mode. When the center LED is illuminated the device is operating in brake mode. The Brake/Coast mode can be toggled by pressing down on the center of the triangle and then releasing the button.

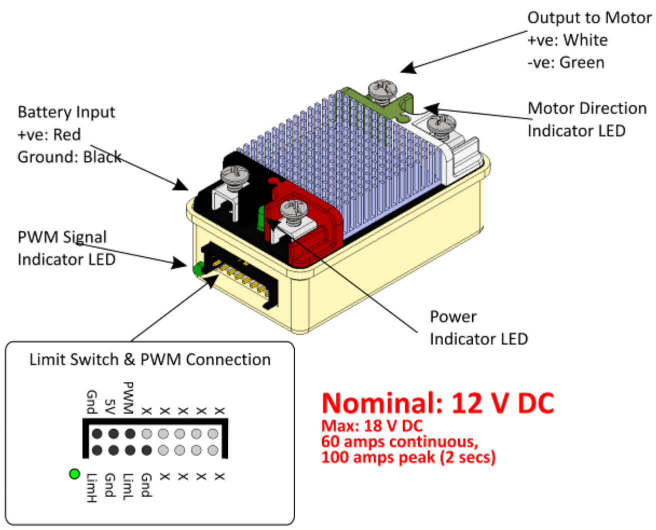
Jaguar speed controllers

LED State	Module Status
Normal Operating Conditions	
Solid Yellow	Neutral (speed set to 0)
Fast Flashing Green	Forward
Fast Flashing Red	Reverse
Solid Green	Full-speed forward
Solid Red	Full-speed reverse
Fault Conditions	
Slow Flashing Yellow	Loss of servo or Network link
Fast Flashing Yellow	Invalid CAN ID
Slow Flashing Red	Voltage, Temperature, or Limit Switch fault condition
Slow Flashing Red and Yellow	Current fault condition

LED State	Module Status
Calibration Conditions	
Fast Flashing Red and Green	Calibration mode active
Fast Flashing Red and Yellow	Calibration mode failure
Slow Flashing Green and Yellow	Calibration mode success
Slow Flashing Red and Green	Calibration mode reset to factory default settings success
Other Conditions	
Slow Flashing Green	Waiting in CAN Assignment mode

Troubleshooting

Mindsensors SD 540



Power LED

This LED will turn Red when Power is supplied.

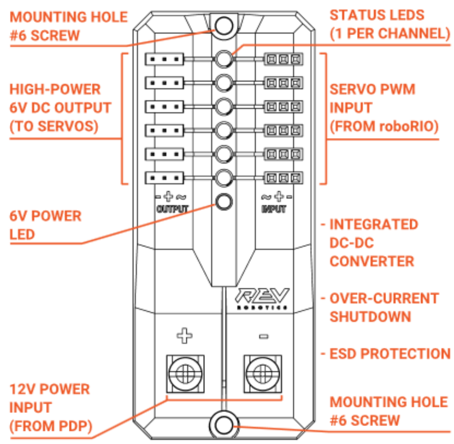
Motor LED

This LED turns Red in Forward direction and Green in Reverse direction.

PWM Signal LED

This LED turns Red when no valid PWM signal is detected, and turns Green when valid PWM signal is detected.

REV Robotics Servo Power Module



STATUS LEDs

Each channel has a corresponding status LED that will indicate the sensed state of the connected PWM signal. The table below describes each state's corresponding LED pattern.

State	Pattern
No Signal	Blinking Amber
Left/Reverse Signal	Solid Red
Center/Neutral Signal	Solid Amber
Right/Forward Signal	Solid Green

6V Power LED off, dim or flickering with power applied = Over-current shutdown

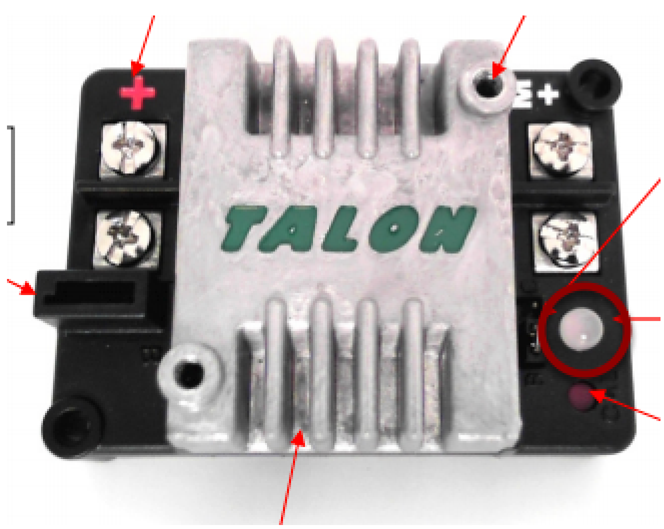
REV Robotics SPARK

2.6 STATUS LED

The SPARK can display information about its current mode of operation via its tri-colored STATUS LED. The STATUS LED is located next to the motor output terminals and is labeled as STATUS with raised lettering on the SPARK housing.

Figure 2-6 shows the status codes associated with each operating state of the SPARK.

Talon speed controllers



The LED is used to indicate the direction and percentage of throttle and state of calibration. The LED may be one of three colors; red, orange or green. A solid green LED indicates positive output voltage equal to the input voltage of the Talon. A solid Red LED indicates an output voltage that is equal to the input voltage multiplied by -1(input voltage = 12 volts, output equals -12 volts). The LED will blink it’s corresponding color for any throttle less than 100% (red indicates negative polarity, green indicates positive). The rate at which the led blinks is proportional to the percent throttle. The faster the LED blinks the closer the output is to 100% in either polarity.

The LED will blink orange any time the Talon is in the disabled state. This will happen if the PWM input signal is lost, or in FRC, when the robot is disabled. If the Talon is in the enabled state and the throttle is within the 4% dead band, the LED will remain solid orange.

Flashing Red/Green indicate ready for calibration. Several green flashes indicates successful calibration, and red several times indicates unsuccessful calibration.

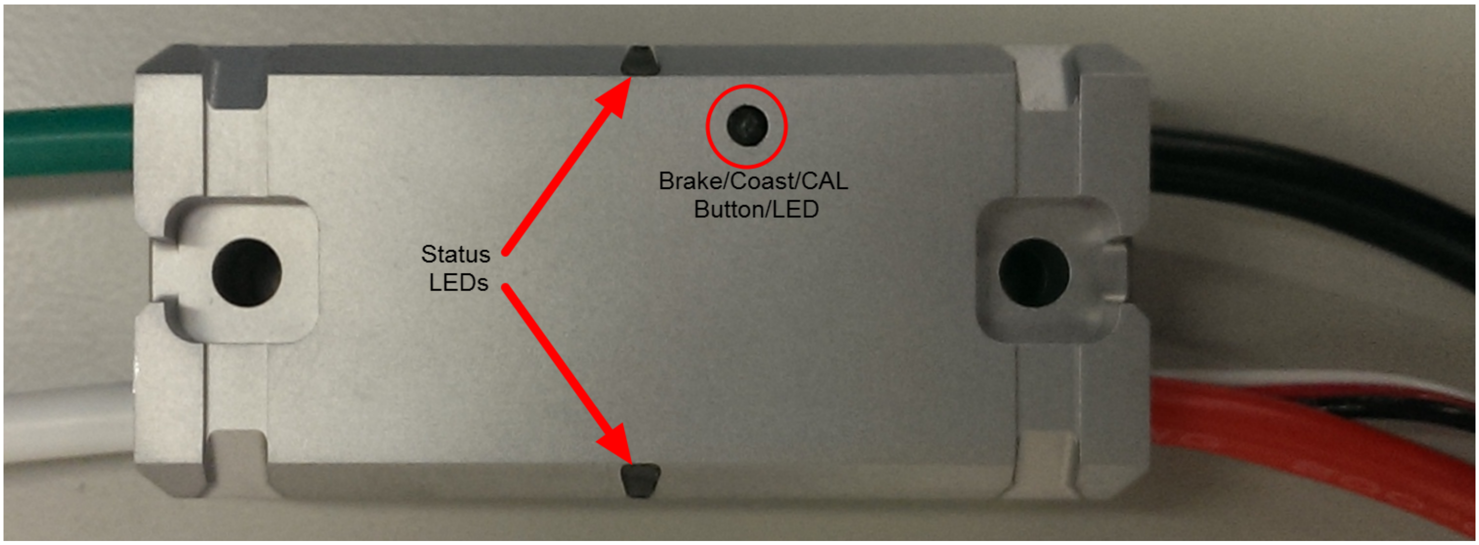
Victor speed controllers

LED Indicator Status:

Green - full forward
Orange - neutral / brake
Red - full reverse
Flashing orange - no PWM signal
Flashing red/green - calibration mode

Flashing green - successful calibration
Flashing red - unsuccessful calibration

Victor-SP speed controllers



Brake/Coast/Cal Button/LED - Red if the controller is in brake mode, off if the controller is in coast mode

Status

The Status LEDs are used to indicate the direction and percentage of throttle and state of calibration. The LEDs may be one of three colors; red, orange or green. Solid green LEDs indicate positive output voltage equal to the input voltage of the Victor-SP. Solid Red LEDs indicate an output voltage that is equal to the input voltage multiplied by -1(input voltage = 12 volts, output equals -12 volts). The LEDs will blink in the corresponding color for any throttle less than 100% (red indicates negative polarity, green indicates positive). The rate at which the LEDs blink is proportional to the percent throttle. The faster the LEDs blink the closer the output is to 100% in either polarity.

The LEDs will blink orange any time the Victor-SP is in the disabled state. This will happen if the PWM input signal is lost, or in FRC, when the robot is disabled. If the Victor-SP is in the enabled state and the throttle is within the 4% dead band, the LED will remain solid orange.

Flashing Red/Green indicate ready for calibration. Several green flashes indicates successful calibration, and red several times indicates unsuccessful calibration.

Troubleshooting

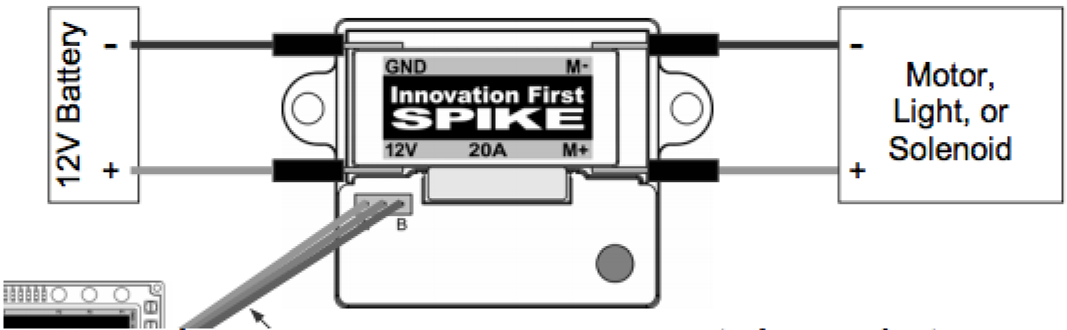
Talon-SRX speed controllers

Blink Codes During Calibration	
Status LEDs Blink Code	**Talon SRX State**
Flashing Red/Green	Calibration Mode
Blinking Green	Successful Calibration
Blinking Red	Failed Calibration

Blink Codes During Normal Operation		
LEDs	**Colors**	**Talon SRX State**
Both	Blinking Green	Forward throttle is applied. Blink rate is proportional to Duty Cycle
Both	Blinking Red	Reverse throttle is applied. Blink rate is proportional to Duty Cycle
None	None	No Power is being applied to Talon SRX
LEDs Alternate[1]	Off/Orange	CAN bus detected, robot disabled
LEDs Alternate[1]	Off/Slow Red	CAN bus/PWM is not detected
LEDs Alternate[1]	Off/Fast Red	Fault Detected
LEDs Alternate[1]	Red/Orange	Damaged Hardware
LEDs Strobe "towards" (M+)[2]	Off/Red	Forward Limit Switch or Forward Soft Limit
LEDs Strobe "towards" (M-)[2]	Off/Red	Reverse Limit Switch or Reverse Soft Limit
LED1 Only "closest" to M+/V+	Green/Orange	In Boot-loader

B/C CAL Blink Codes	
B/C CAL Button Color	**Talon SRX State**
Solid Red	Brake Mode
Off	Coast Mode

Spike relay configured as a motor, light, or solenoid switch

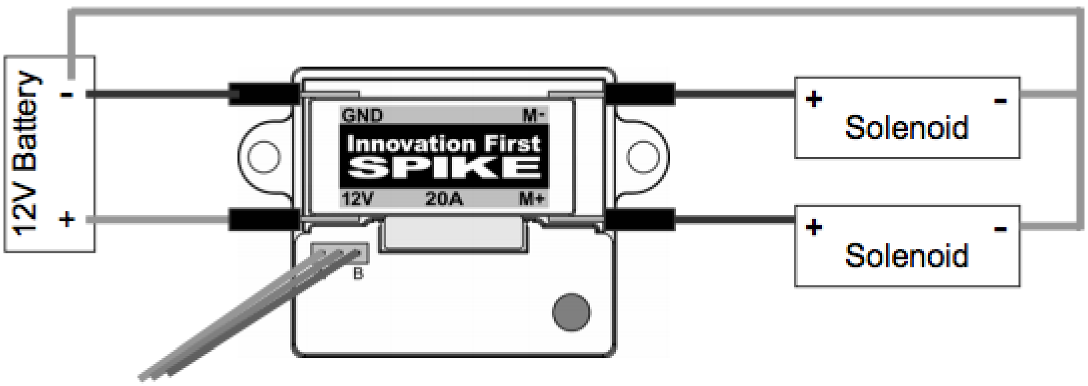


INPUTS		OUTPUTS			
Fwd(Wht)	Rev(Red)	M+	M-	Indicator	Motor Function
0	0	GND	GND	Orange	OFF / Brake Condition (default)
1	0	+12v	GND	Green	Motor rotates in one direction
0	1	GND	+12v	Red	Motor rotates in opposite direction
1	1	+12v	+12v	Off	OFF / Brake Condition

Notes:

1. 'Brake' refers to the dynamic stopping of the motor due to the shorting of the motor inputs. This condition is not optional when going to an off state.
2. The INPUT Fwd and Rev are defined as follows: 0 (Off) and 1 (On).

Spike relay configured as for one or two solenoids



INPUT		OUTPUTS			
Fwd(Wht)	**Rev(Red)**	**M+**	**M-**	**Indicator**	**Solenoid Function**
0	0	GND	GND	Orange	Both Solenoids OFF (default)
1	0	+12v	GND	Green	Solenoid connected to M+ is ON
0	1	GND	+12v	Red	Solenoid connected to M- is ON
1	1	+12v	+12v	Off	Both Solenoids ON

Note:

1. The INPUT Fwd and Rev are defined as follows: 0 (Off) and 1 (On).

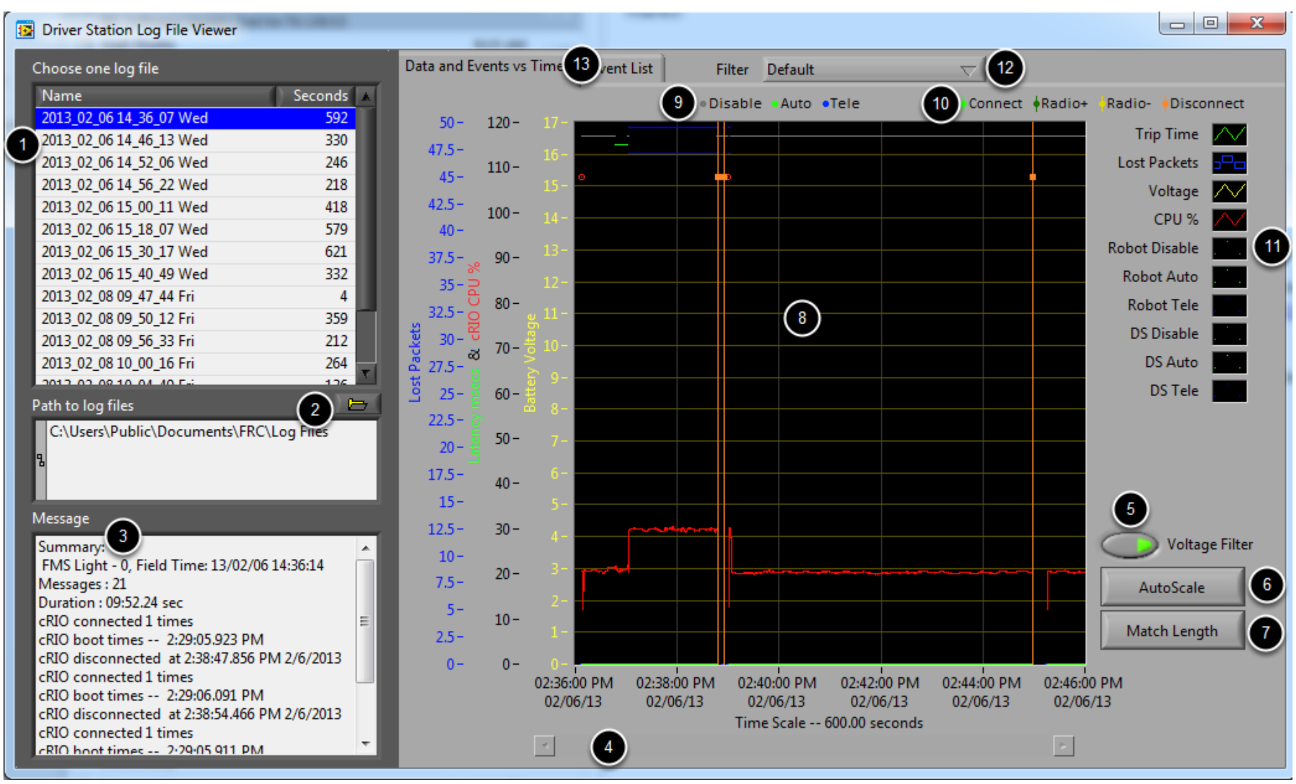
Driver Station Log File Viewer

In an effort to provide information to aid in debugging, the FRC Driver Station creates log files of important diagnostic data while running. These logs can be reviewed later using the FRC Driver Station Log Viewer. The Log Viewer can be found via the shortcut installed in the Start menu or in the FRC Driver Station folder in Program Files.

Event Logs

A new addition to the Driver Station logging this year is the Event Log. The Driver Station now logs all messages sent to the Messages box on the Diagnostics tab (not the User Messages box on the Operation tab) into a new Event Log file. When viewing Log Files with the Driver Station Log File Viewer, the Event Log and DSLog files are overlaid in a single display.

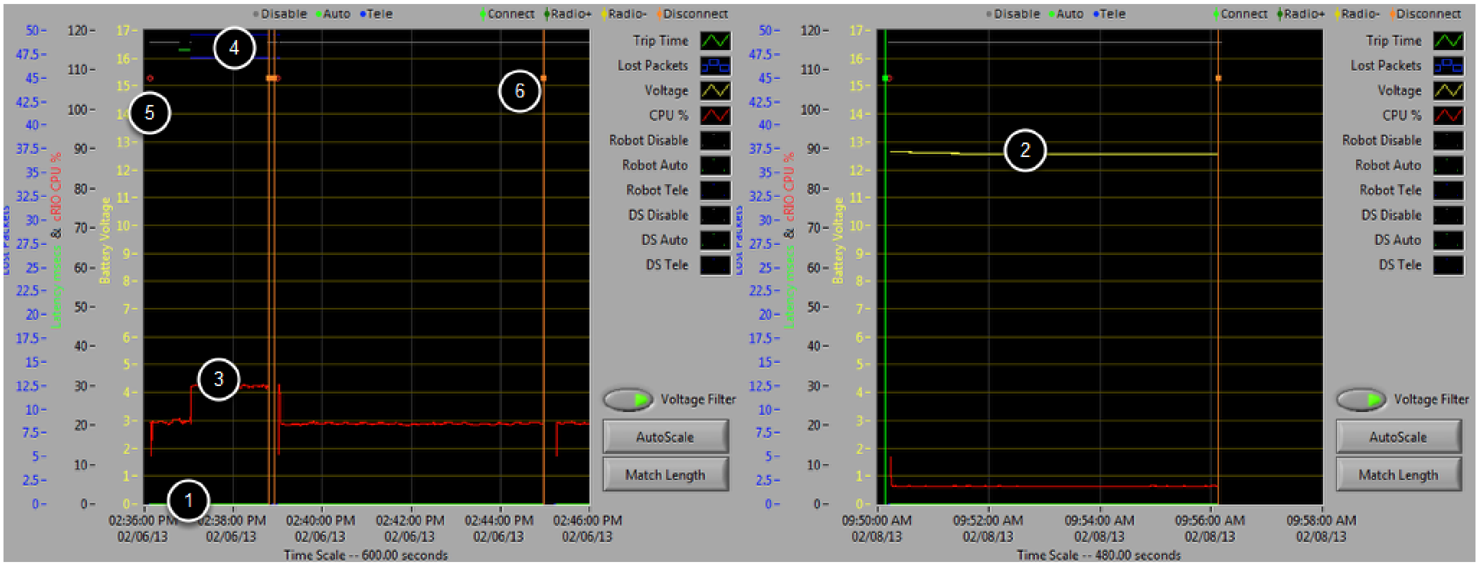
Log Viewer UI



The Log Viewer contains a number of controls and displays to aid in the analysis of the Driver Station log files:

1. File Selection Box - This window displays all available log files in the currently selected folder. Click on a log file in the list to select it.
2. Path to Log Files - This box displays the current folder the viewer is looking in for log files. This defaults to the folder that the Driver Station stores log files in. Click the folder icon to browse to a different location.
3. Message Box - This box displays a summary of all messages from the Event Log. When hovering over an event on the graph this box changes to display the information for that event.
4. Scroll Bar - When the graph is zoomed in, this scroll bar allows for horizontal scrolling of the graph.
5. Voltage Filter - This control turns the Voltage Filter on and off (defaults to on). The Voltage Filter filters out data such as CPU %, robot mode and trip time when no Battery Voltage is received (indicating that the DS is not in communication with the roboRIO). This does not filter out data when the battery voltage being received is 0 due to a missing jumper on the Analog Module or no power provided to the Analog Module.
6. AutoScale - This button zooms the graph out to show all data in the log.
7. Match Length - This button scales the graph to approximately the length of an FRC match (2 minutes and 30 seconds shown). It does not automatically locate the start of the match, you will have to scroll using the scroll bar to locate the beginning of the Autonomous mode.
8. Graph - This display shows graph data from the DS Log file (voltage, trip time, roboRIO CPU%, Lost Packets, and robot mode) as well as overlaid event data (shown as dots on the graph with select events showing as vertical lines across the entire graph). Hovering over event markers on the graph displays information about the event in the Messages window in the bottom left of the screen.
9. Robot Mode Key - Key for the Robot Mode displayed at the top of the screen
10. Major event key - Key for the major events, displayed as vertical lines on the graph
11. Graph key - Key for the graph data
12. Filter Control - Drop-down to select the filter mode (filter modes explained below)
13. Tab Control - Control to switch between the Graph (Data and Events vs. Time) and Event List displays.

Using the Graph Display

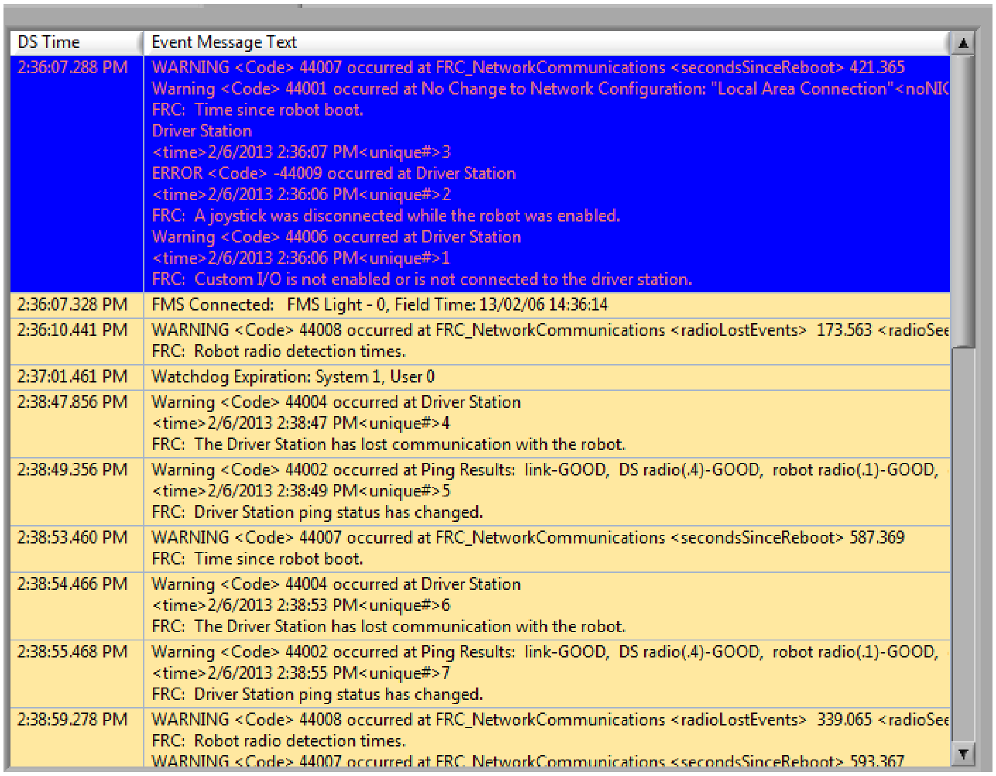


The Graph Display contains the following information:

1. Graphs of Trip Time in ms (green line) and Lost Packets per second (displayed as blue vertical bars). In these example images Trip Time is a flat green line at the bottom of the graph and there are no lost packets
2. Graph of Battery voltage displayed as a yellow line.
3. Graph of roboRIO CPU % as a red line
4. Graph of robot mode and DS mode. The top set of the display shows the mode commanded by the Driver Station. The bottom set shows the mode reported by the robot code. In this example the robot is not reporting it's mode during the disabled and autonomous modes, but is reported during Teleop.
5. Event markers will be displayed on the graph indicating the time the event occurred. Errors will display in red; warnings will display in yellow. Hovering over an event marker will display information about the event in the Messages box at the bottom left of the screen.
6. Major events are shown as vertical lines across the graph display.

To zoom in on a portion of the graph, click and drag around the desired viewing area. You can only zoom the time axis, you cannot zoom vertically.

Event List



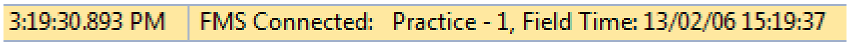
The Event List tab displays a list of events (warnings and errors) recorded by the Driver Station. The events and detail displayed are determined by the currently active filter (images shows "All Events, All Info" filter active).

Filters

Three filters are currently available in the Log Viewer:

1. Default: This filter filters out many of the errors and warnings produced by the Driver Station. This filter is useful for identifying errors thrown by the code on the Robot.
2. All Events and Time: This filter shows all events and the time they occured.
3. All Events, All Info: This filter shows all events and all recorded info. At this time the primary difference between this filter and "All Events and Time" is that this option shows the "unique" designator for the first occurrence of a particular message.

Identifying Logs from Matches



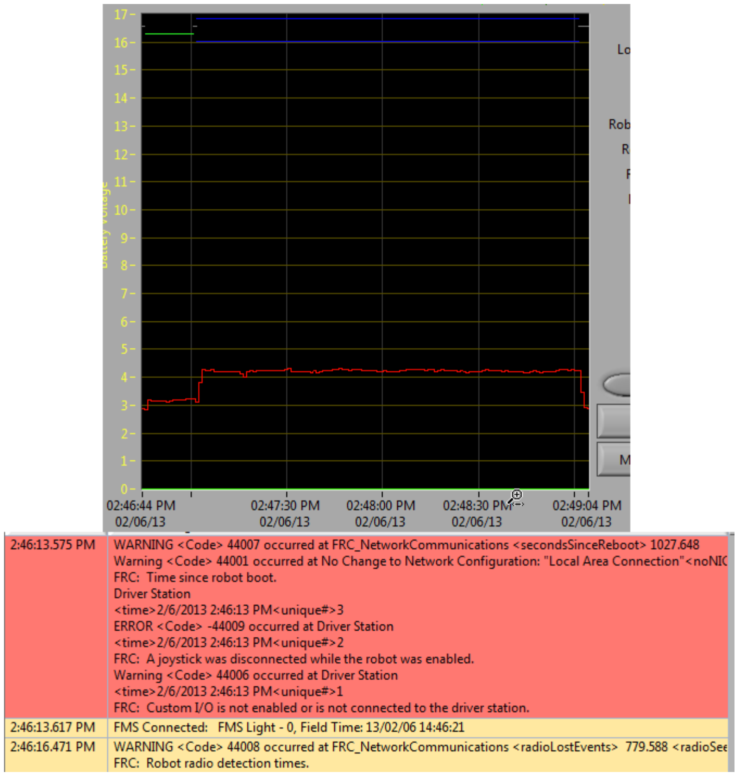
A common task when working with the Driver Station Logs is to identify which logs came from competition matches. Logs which were taken during a match can now be identified using the FMS Connected event which will display the match type (Practice, Qualification or Elimination), match number, and the current time according to the FMS server. In this example, you can see that the FMS server time and the time of the Driver Station computer are fairly close, approximately 7 seconds apart.

Identifying Common Connection Failures with the Log Viewer

When diagnosing robot issues, there is no substitute for thorough knowledge of the system and a methodical debugging approach. If you need assistance diagnosing a connection problem at your events it is strongly recommended to seek assistance from your FTA and/or CSA. The goal of this section is to familiarize teams with how some common failures can manifest themselves in the DS Log files. Please note that depending on a variety of conditions a particular failure show slightly differently in a log file.

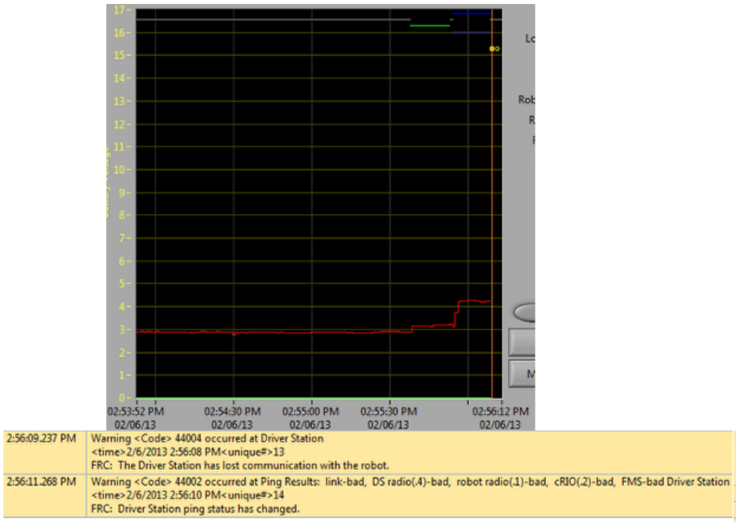
Note that all log files shown in this section have been scaled to match length using the Match Length button and then scrolling to the beginning of the autonomous mode. Also, many of the logs do not contain battery voltage information, the platform used for log capture was not properly wired for reporting the battery voltage.

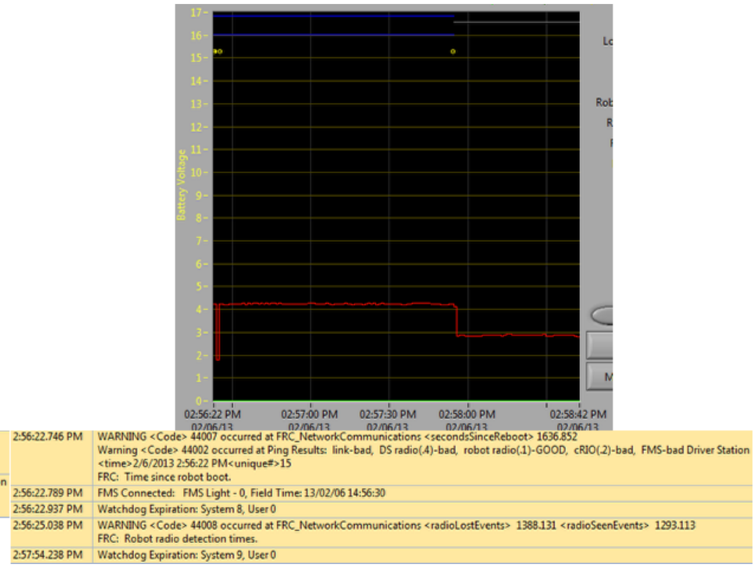
"Normal" Log



This is an example of a normal match log. The errors and warnings contained in the first box are from when the DS first started and can be ignored. This is confirmed by observing that these events occurred prior to the "FMS Connected:" event. The last event shown can also be ignored, it is also from the robot first connecting to the DS (it occurs 3 seconds after connecting to FMS) and occurs roughly 30 seconds before the match started.

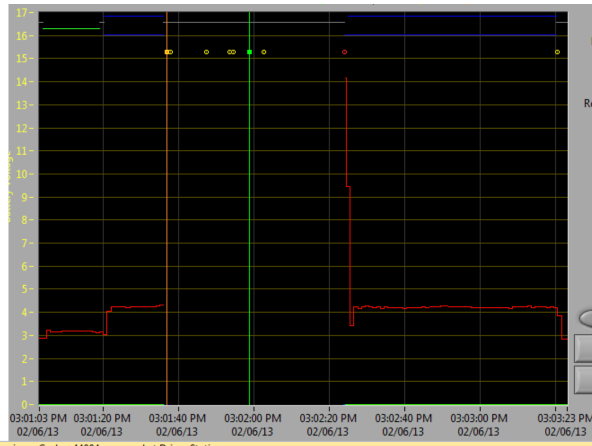
Disconnected from FMS





When the DS disconnects from FMS, and therefore the robot, during the match it may segment the log into pieces. The key indicators to this failure are the last event of the first log, indicating that the connection to FMS is now "bad" and the second event from the 2nd log which is a new FMS connected message followed by the DS immediately transitioning into Teleop Enabled. The most common cause of this type of failure is an ethernet cable with no latching tab or a damaged ethernet port on the DS computer.

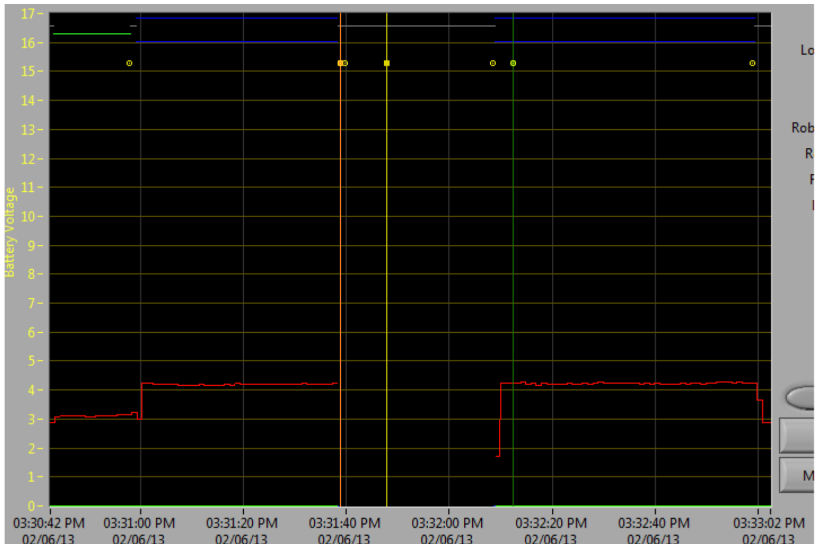
roboRIO Reboot



3:01:36.869 PM	Warning <Code> 44004 occurred at Driver Station <time>2/6/2013 3:01:36 PM<unique#>4 FRC: The Driver Station has lost communication with the robot.
3:01:37.871 PM	Warning <Code> 44002 occurred at Ping Results: link-GOOD, DS radio(.4)-GOOD, robot radio(.1)-GOOD, cRIO(.2)-bad, FMS-GOOD Driver Station <time>2/6/2013 3:01:37 PM<unique#>5 FRC: Driver Station ping status has changed.
3:01:47.281 PM	Warning <Code> 44002 occurred at Ping Results: link-GOOD, DS radio(.4)-GOOD, robot radio(.1)-GOOD, cRIO(.2)-GOOD, FMS-GOOD Driver Station <time>2/6/2013 3:01:46 PM<unique#>6 FRC: Driver Station ping status has changed.
3:01:53.689 PM	Warning <Code> 44002 occurred at Ping Results: link-GOOD, DS radio(.4)-GOOD, robot radio(.1)-GOOD, cRIO(.2)-bad, FMS-GOOD Driver Station <time>2/6/2013 3:01:52 PM<unique#>7 FRC: Driver Station ping status has changed.
3:01:54.490 PM	Warning <Code> 44002 occurred at Ping Results: link-GOOD, DS radio(.4)-GOOD, robot radio(.1)-GOOD, cRIO(.2)-GOOD, FMS-GOOD Driver Station <time>2/6/2013 3:01:53 PM<unique#>8 FRC: Driver Station ping status has changed.
3:02:02.512 PM	WARNING <Code> 44007 occurred at FRC_NetworkCommunications <secondsSinceReboot> 3.682 FRC: Time since robot boot.
3:02:24.128 PM	ERROR <Code> -44003 occurred at WPI_CameraIssue HTTP Request with Authentication.vi>>WPI_CameraIssue Get.vi>>WPI_CameraGet Image Appea <time>21:30:35 02/06/2013 FRC: Operation failed due to a communication failure with the camera.
3:02:24.151 PM	Warning <Code> 44003 occurred at Driver Station <time>2/6/2013 3:02:24 PM<unique#>9 FRC: No robot code is currently running.

The "Time since robot boot" message is the primary indicator in a connection failure caused by the roboRIO rebooting. In this log the DS loses connection with the roboRIO at 3:01:36 as indicated by the first event. The second event indicates that the ping initiated after the connection failed was successful to all devices other than the roboRIO. At 3:01:47 the roboRIO begins responding to pings again, one additional ping fails at 3:01:52. At 3:02:02 the Driver Station connects to the roboRIO and the roboRIO reports that it has been up for 3.682 seconds. This is a clear indicator that the roboRIO has rebooted. The code continues to load and at 3:02:24 the code reports an error communicating with the camera. A warning is also reported indicating that no robot code is running right before the code finishes starting up.

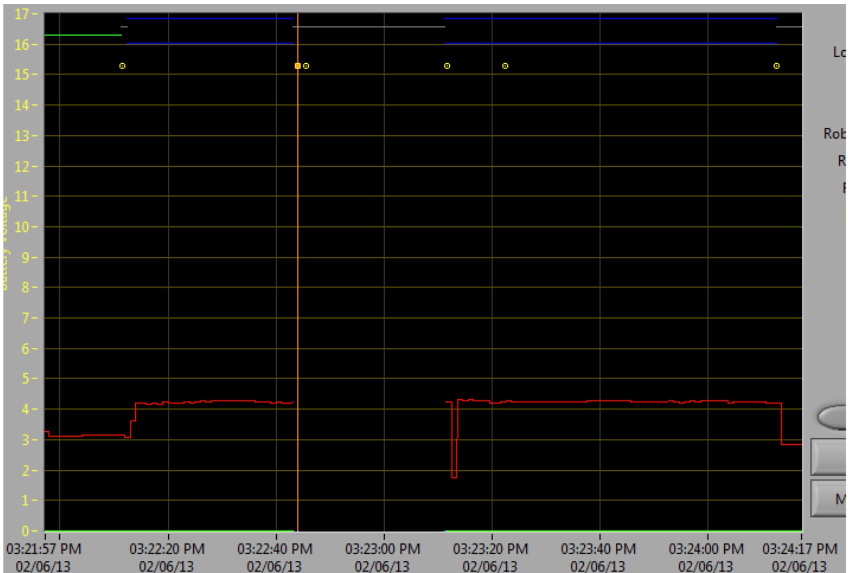
Ethernet cable issue on robot



3:30:57.799 PM	Watchdog Expiration: System 5, User 0
3:31:38.800 PM	Warning <Code> 44004 occurred at Driver Station <time>2/6/2013 3:31:38 PM<unique#>10 FRC: The Driver Station has lost communication with the robot.
3:31:39.801 PM	Warning <Code> 44002 occurred at Ping Results: link-GOOD, DS radio(.4)-GOOD, robot radio(.1)-GOOD, cRIO(.2)-bad, FMS-GOOD Driver Station <time>2/6/2013 3:31:39 PM<unique#>11 FRC: Driver Station ping status has changed.
3:32:08.449 PM	WARNING <Code> 44007 occurred at FRC_NetworkCommunications <secondsSinceReboot> 1809.393 FRC: Time since robot boot.
3:32:12.399 PM	WARNING <Code> 44008 occurred at FRC_NetworkCommunications <radioLostEvents> 24.505,1492.752 <radioSeenEvents> 0.000,260.086 FRC: Robot radio detection times.
3:32:59.018 PM	Watchdog Expiration: System 7, User 0

An issue with the ethernet cable on the robot is primarily indicated by the ping to the roboRIO going to bad and Radio Lost and Radio Seen events when the roboRIO reconnects. The "Time since robot boot" message when the roboRIO reconnects will also indicate that the roboRIO has not rebooted. In this example, the robot Ethernet cable was disconnected at 3:31:38. The ping status indicates that the D-Link radio is still connected. When the robot reconnects at 3:32:08 the "Tim since robot boot" is 1809 seconds indicating that the roboRIO clearly did not reboot. At 3:32:12 the robot indicates that it lost the radio 24.505 seconds ago and it returned 0.000 seconds ago. These points are plotted as vertical lines on the graph, yellow for radio lost and green for radio seen. Note that the times are slightly offset from the actual events as shown via the disconnection and connection, but help to provide additional information about what is occurring.

Radio reboot



3:22:11.489 PM	Watchdog Expiration: System 2, User 0
3:22:44.030 PM	Warning <Code> 44004 occurred at Driver Station <time>2/6/2013 3:22:43 PM<unique#>6 FRC: The Driver Station has lost communication with the robot.
3:22:45.531 PM	Warning <Code> 44002 occurred at Ping Results: link-GOOD, DS radio(.4)-GOOD, robot radio(.1)-bad, cRIO(.2)-bad, FMS-GOOD Driver Station <time>2/6/2013 3:22:45 PM<unique#>7 FRC: Driver Station ping status has changed.
3:23:11.620 PM	WARNING <Code> 44007 occurred at FRC_NetworkCommunications <secondsSinceReboot> 1272.775 FRC: Time since robot boot.
3:23:11.690 PM	Watchdog Expiration: System 3, User 0

A reboot of the robot radio is typically characterized by a loss of connection to the radio for ~25-30 seconds. In this example, the radio briefly lost power at 3:22:44, causing it to start rebooting. The event at 3:22:45 indicates that the ping to the radio failed. At 3:23:11, the DS regains communication with the roboRIO and the roboRIO indicates it gas been up for 1272.775 seconds, ruling out a roboRIO reboot. Note that the network switch on the radio comes back up very quickly so a momentary power loss may not result in a "radio lost"/"radio seen" event pair. A longer disturbance may result in radio events being logged by the DS. In that case, the distinguishing factor which points towards a radio reboot is the ping status of the radio from the DS. If the radio resets, the radio will be unreachable. If the issue is a cabling or connection issue on the robot, the radio ping should remain "GOOD".

RoboRIO Brownout and Understanding Current Draw

In order to help maintain battery voltage to preserve itself and other control system components such as the radio during high current draw events, the roboRIO contains a staged brownout protection scheme. This article describes this scheme, provides information about proactively planning for system current draw, and describes how to use the new functionality of the PDP as well as the DS Log File Viewer to understand brownout events if they do happen on your robot.

roboRIO Brownout Protection

The roboRIO uses a staged brownout protection scheme to attempt to preserve the input voltage to itself and other control system components in order to prevent device resets in the event of large current draws pulling the battery voltage dangerously low.

Stage 1 - Output Disable

Voltage Trigger - 6.8V

When the voltage drops below 6.8V, the controller will enter the brownout protection state. The following indicators will show that this condition has occurred:

- Power LED on the roboRIO will turn Amber
- Background of the voltage display on the Driver Station will turn red
- Mode display on the Driver Station will change to Voltage Brownout
- The CAN\Power tab of the DS will increment the 12V fault counter by 1.
- The DS will record a brownout event in the DS log.

The controller will take the following steps to attempt to preserve the battery voltage:

- PWM outputs will be disabled. For PWM outputs which have set their neutral value (all speed controllers in WPILib) a single neutral pulse will be sent before the output is disabled.
- 6V User Rail disabled (this is the rail that powers servos on the PWM header bank)

- GPIO configured as outputs go to High-Z
- Relay Outputs are disabled (driven low)
- CAN-based motor controllers are sent an explicit disable command

The controller will remain in this state until the voltage rises to greater than 7.5V or drops below the trigger for the next stage of the brownout

Stage 2 - User Voltage Rail Disable

Voltage Trigger - 6.3V

When the voltage drops below 6.3V, the User Voltage Rails are disabled. This includes the 5V pins (or 3.3V is the jumper has been set) in the DIO connector bank, the 5V pins in the Analog bank, the 3.3V pins in the SPI and I2C bank and the 5V and 3.3V pins in the MXP bank.

The controller will remain in this state until the voltage rises above 6.3V (return to Stage 2) or drops below the trigger for the next stage of the brownout

Stage 3 - Device Blackout

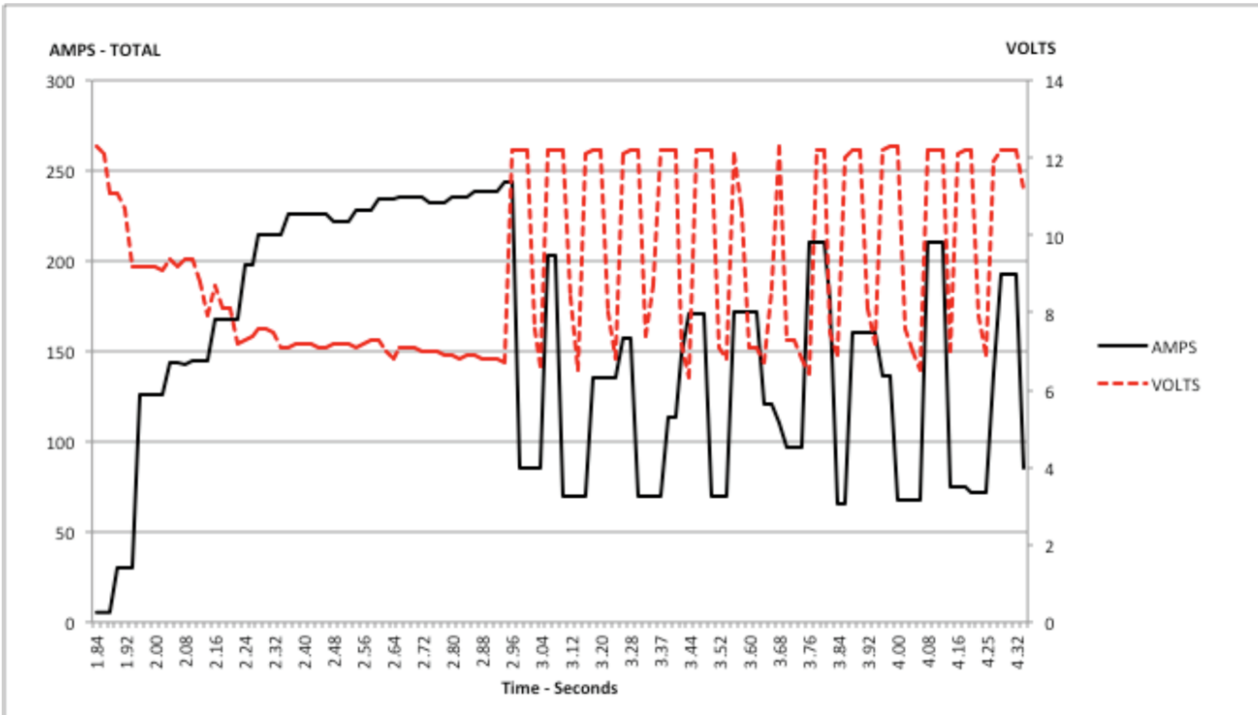
Voltage Trigger - 4.5V

Below 4.5V the device may blackout. The exact voltage may be lower than this and depends on the load on the device.

The controller will remain in this state until the voltage rises above 4.65V when the device will begin the normal boot sequence.

Avoiding Brownout - Proactive Current Draw planning

PLOT 1 – AMPS and VOLTS v. Time – 2.5 Second Window



The key to avoiding a brownout condition is to proactively plan for the current draw of your robot. The best way to do this is to create some form of power budget. This can be a complex document that attempts to quantify both estimated current draw and time in an effort to most completely understand power usage and therefore battery state at the end of a match, or it can be a simple inventory of current usage. To do this:

1. Establish the max "sustained" current draw (with sustained being loosely defined here as not momentary). This is probably the most difficult part of creating the power budget. The exact current draw a battery can sustain while maintaining a voltage of 7+ volts is dependent on a variety of factors such as battery health and state of charge. As shown in the NP18-12 data sheet, the terminal voltage chart gets very steep as state of charge decreases, especially as current draw increases. This datasheet shows that at 3CA continuous load (54A) a brand new battery can be continuously run for over 6 minutes while maintaining a terminal voltage of over 7V. As shown in the image above (used with permission from Team 234's Drive System Testing document), even with a fresh battery, drawing 240A for more than a second or two is likely to cause an issue. This gives us some bounds on setting our sustained current draw. For the purposes of this exercise, we'll set our limit at 180A.

2. List out the different functions of your robot such as drivetrain, manipulator, main game mechanism, etc.
3. Start assigning your available current to these functions. You will likely find that you run out pretty quickly. Many teams gear their drivetrain to have enough torque to slip their wheels at 40-50A of current draw per motor. If we have 4 motors on the drivetrain, that eats up most, or even exceeds, our power budget! This means that we may need to put together a few scenarios and understand what functions can (and need to be) be used at the same time. In many cases, this will mean that you really need to limit the current draw of the other functions if/while your robot is maxing out the drivetrain (such as trying to push something). Benchmarking the "driving" current requirements of a drivetrain for some of these alternative scenarios is a little more complex, as it depends on many factors such as number of motors, robot weight, gearing, and efficiency. Current numbers for other functions can be done by calculating the power required to complete the function and estimating efficiency (if the mechanism has not been designed) or by determining the torque load on the motor and using the torque-current curve to determine the current draw of the motos.
4. If you have determined mutually exclusive functions in your analysis, consider enforcing the exclusion in software. You may also use the current monitoring of the PDP (covered in more detail below) in your robot program to provide output limits or exclusions dynamically (such as don't run a mechanism motor when the drivetrain current is over X or only let the motor run up to half output when the drivetrain current is over Y).

Measuring Current Draw using the PDP

The FRC Driver Station works in conjunction with the roboRIO and PDP to extract logged data from the PDP and log it on your DS PC. A viewer for this data is still under development.

In the meantime, teams can use their robot code and manual logging, a LabVIEW front panel or the SmartDashboard to visualize current draw on their robot as mechanisms are developed. In LabVIEW, you can read the current on a PDP channel using the PDP Channel Current VI found on the Power pallete. For C++ and Java teams, use the PowerDistributionPanel class as described in the Power Distribution Panel article. Plotting this information over time (easiest with a LV Front Panel or with the SmartDashboard by using a Graph indicator can provide information to compare against and update your power budget or can locate mechanisms which do not seem to be performing as expected (due to incorrect load calculation, incorrect efficiency assumptions, or mechanism issues such as binding).

Identifying Brownouts

The easiest way to identify a brownout is by clicking on the CAN\Power tab of the DS and checking the 12V fault count. Alternately, you can review the Driver Station Log after the fact using the Driver Station Log Viewer. The log will identify brownouts with a bright orange line, such as in the image above (note that these brownouts were induced with a benchtop supply and may not reflect the duration and behavior of brownouts on a typical FRC robot).

Support Resources

In addition to the documentation here, there are a variety of other resources available to FRC teams to help understand the Control System and software.

Other Documentation

In addition to this site there are a few other places teams may check for documentation:

- NI FRC Community Documents Section
- USFIRST.org Technical Resources Page
- VEXPro Jaguar Page
- CTRE Product Pages

Forums

Stuck? Have a question not answered by the documentation? Official Support is provided on these forums:

- NI FRC Community Discussion Section (roboRIO, LabVIEW and Driver Station software questions)
- USFIRST.org Control System Forum (wiring, hardware and Driver Station questions)
- USFIRST.org Programming Forum (programming questions for C++, Java, or LabVIEW)

NI Phone Support

Have a LabVIEW, roboRIO, or Driver Station question? NI provides phone support for FRC teams during the build season at 866-511-6285 1:00-7:00 PM CST Monday - Friday.

CTRE Support

Support for Cross The Road Electronics components (Pneumatics Control Module, Power Distribution Panel, Talon SRX, and Voltage Regulator Module) is provided via the e-mail address support@crosstheroadelectronics.com

Bug Reporting

Found a bug? Let us know by reporting it in the Issues section of the appropriate WPILibSuite project on Github: https://github.com/wpilibsuite

Specific Issues

Updating your roboRIO firmware

Before updating firmware on 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.

Note that the firmware is separate from the roboRIO image. You will need to update your roboRIO firmware once after receiving the roboRIO and should only need to do so again if instructed specifically by an update.

Note: The shipping roboRIO firmware is acceptable. This procedure should only be needed if you run into an issue imaging a roboRIO that has older firmware.

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 must be wired for power and connected to the computer via USB in order to update the firmware.

USB Connection

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. Do not update via the Ethernet connection.

Make sure that nothing (including the radio) is connected to the Ethernet port when updating the roboRIO firmware.

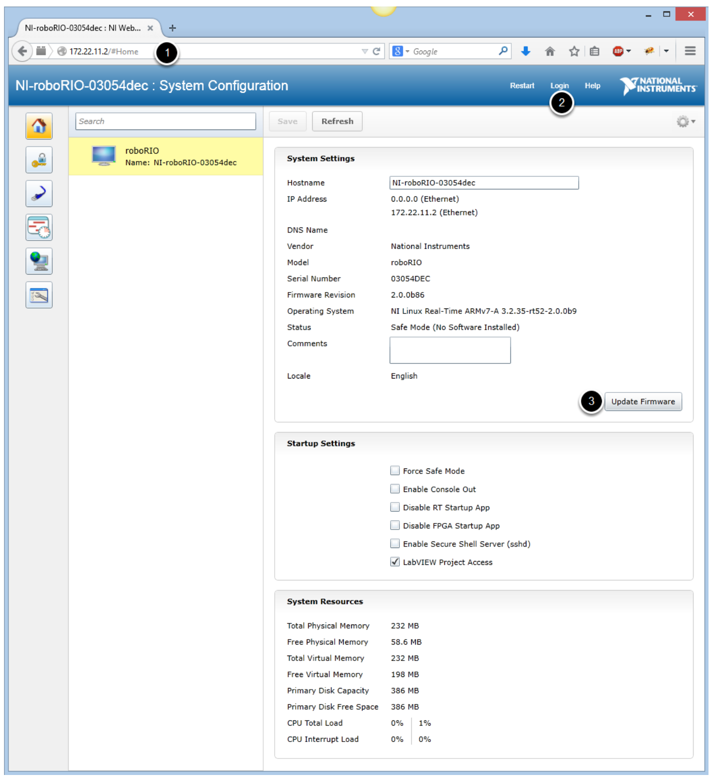
Driver Installation

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. If the Driver Installation fails, make sure you have installed the 2016 FRC Update Suite and that you have rebooted the computer. If that does not work, try a different USB cable, a bad USB cable can result in the driver installation failing.

Firmware Update Using Web Browser

The embedded web server on the roboRIO requires using a browser which supports [Silverlight](). Note that Google Chrome has removed Silverlight support, using Internet Explorer is recommended.

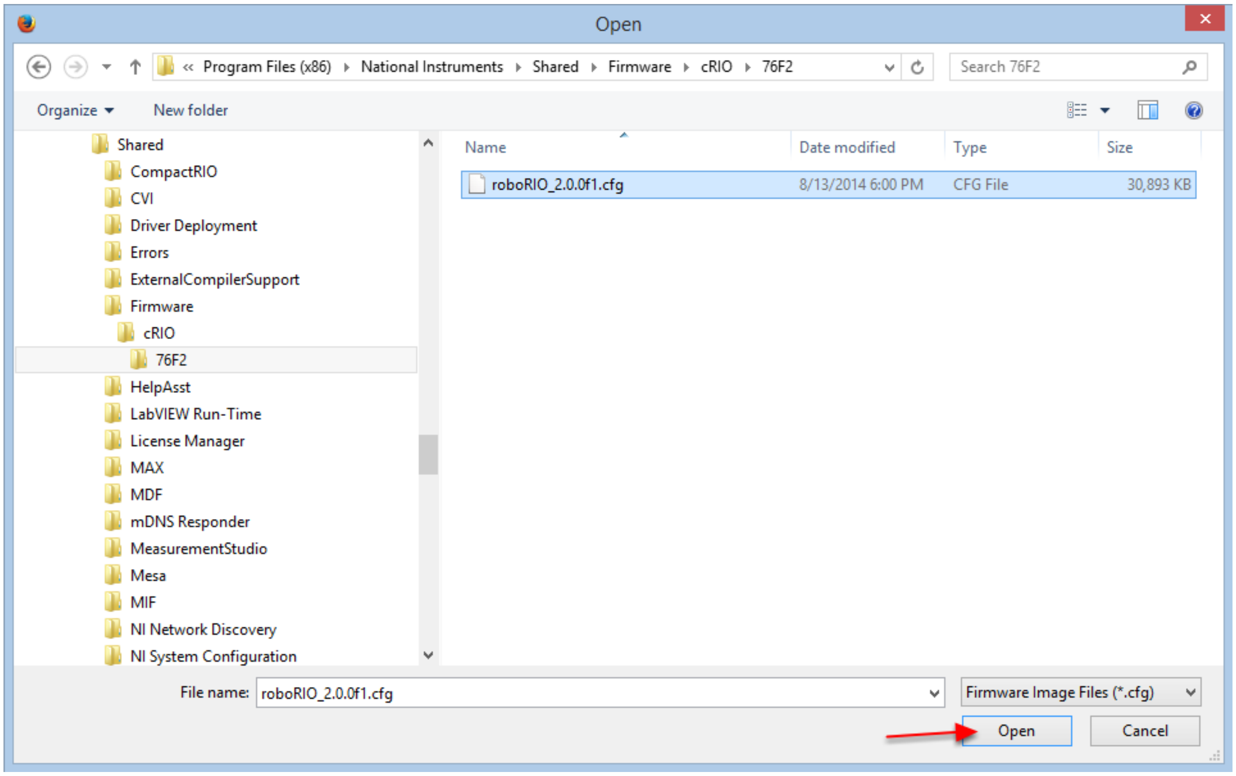
Launch Compatible Browser and Navigate to roboRIO



1. Enter the address 172.22.11.2 in your browser's address bar and press enter.
2. Click Login. Enter "admin" in the Username field and leave the Password field blank.
3. Click Update Firmware

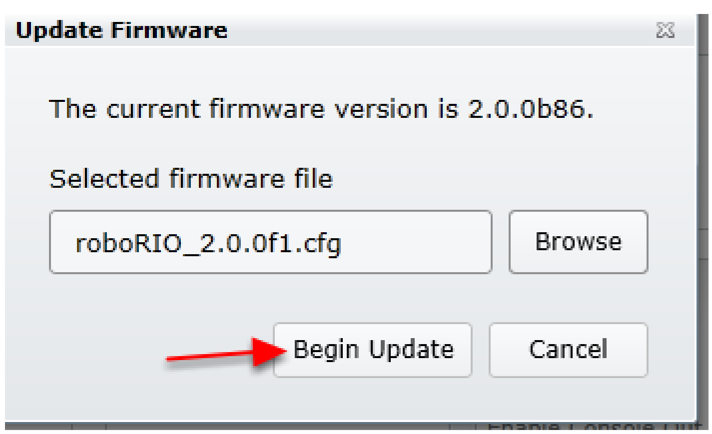
Troubleshooting

Locate Firmware



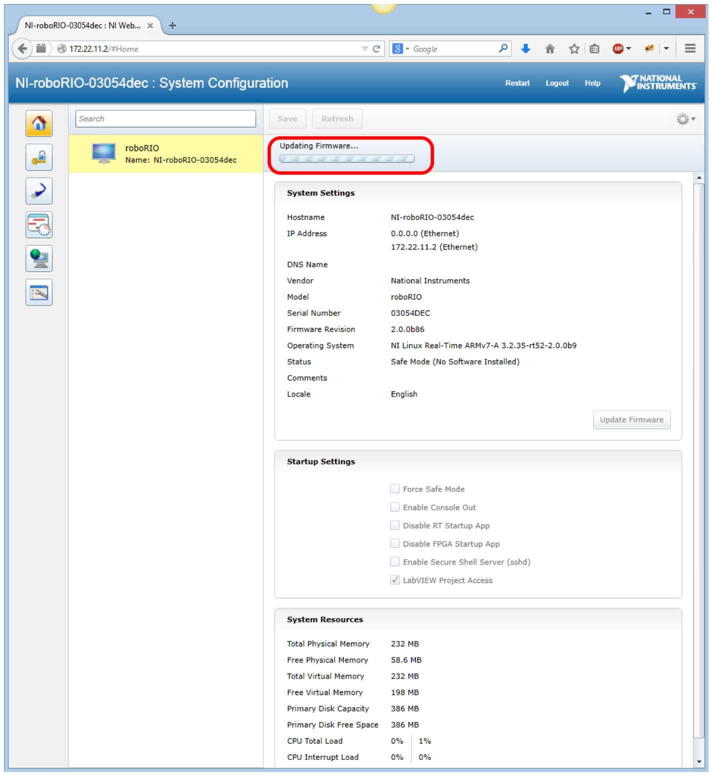
Browse to **Program Files\National Instruments\Shared\Firmware\cRIO\76F2** *Program Files(x86) on 64 bit* and select the latest firmware, then click **Open** (this image shows version 2.0.0f1).

Begin Update



Verify that the firmware selected is correct, then click **Begin Update**

Updating



A progress indicator will appear below the toolbar. Do not close the browser or power off the roboRIO until the process completes and the message appears "The firmware update completed successfully".

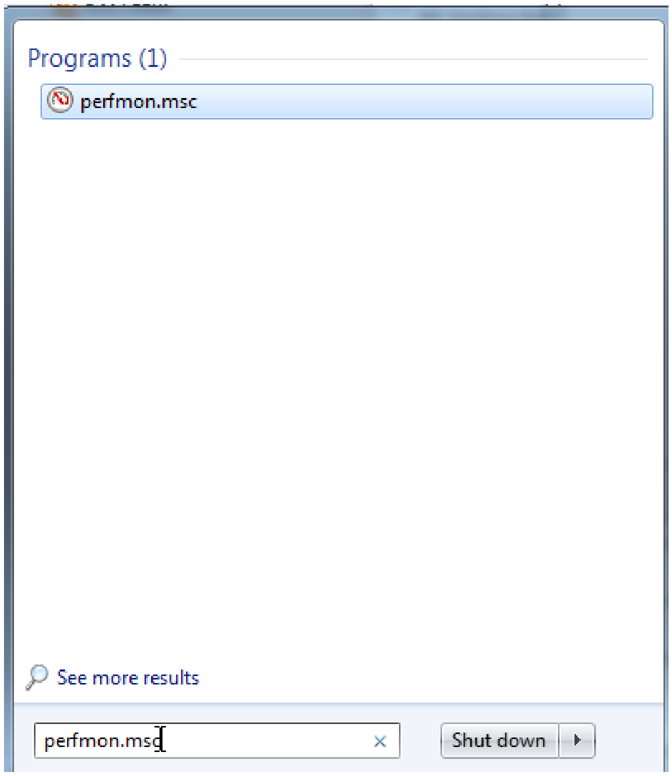
Measuring Bandwidth Usage

On the 2013 FRC Field (and at home when the DAP-1522 is configured using the FRC Bridge Configuration Utility) each team is limited to 7Mb/s of network traffic (see the FMS Whitepaper for more details). The FMS Whitepaper provides information on determining the bandwidth usage of the Axis camera, but some teams may wish to measure their overall bandwidth consumption. This document details how to make that measurement.

Measuring Bandwidth Using the Perfomance Monitor (Win 7 only)

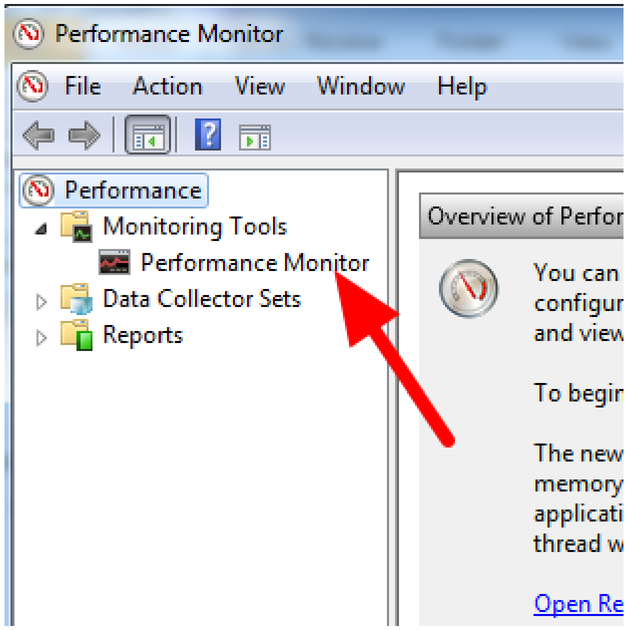
Windows 7 contains a built-in tool called the Performance Monitor that can be used to monitor the bandwidth usage over a network interface.

Launching the Performance Monitor



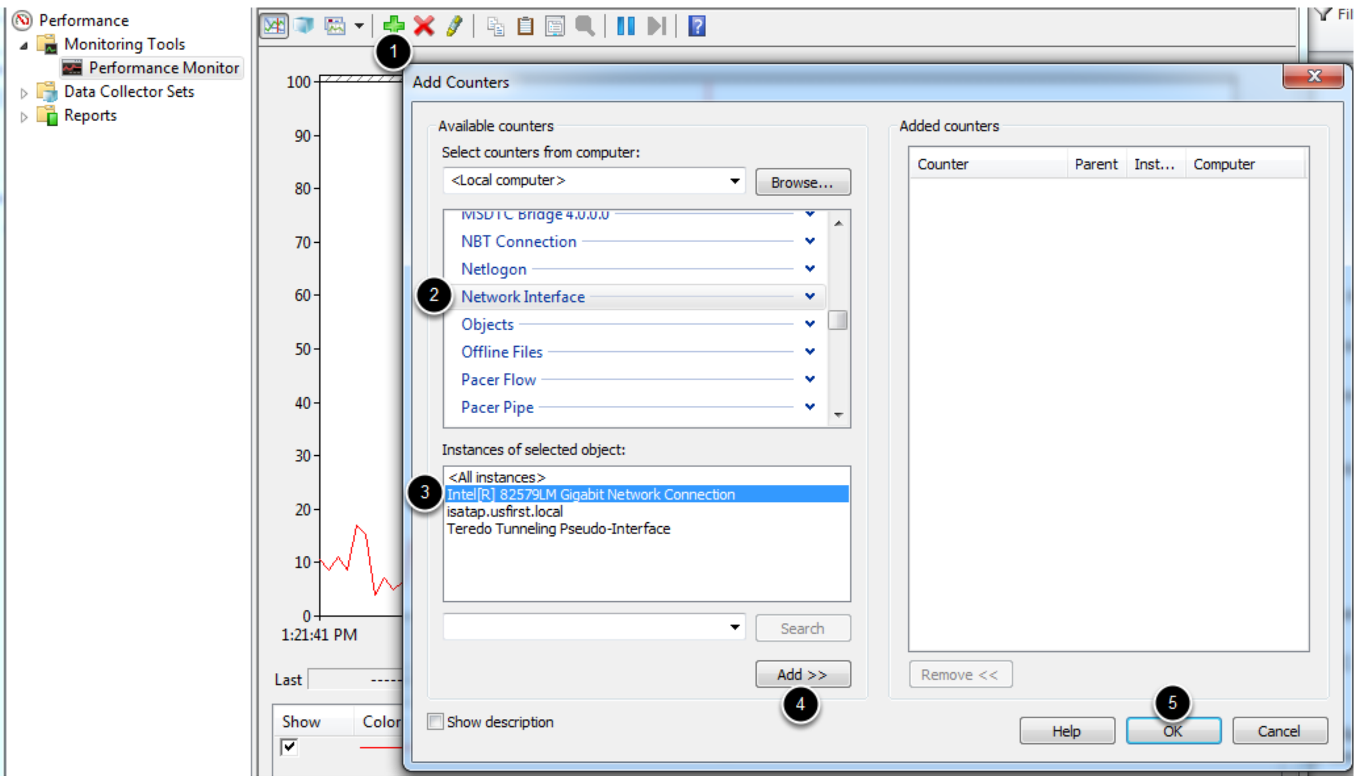
Click **Start** and in the search box, type **perfmon.msc** and press Enter.

Open Real-Time Monitor



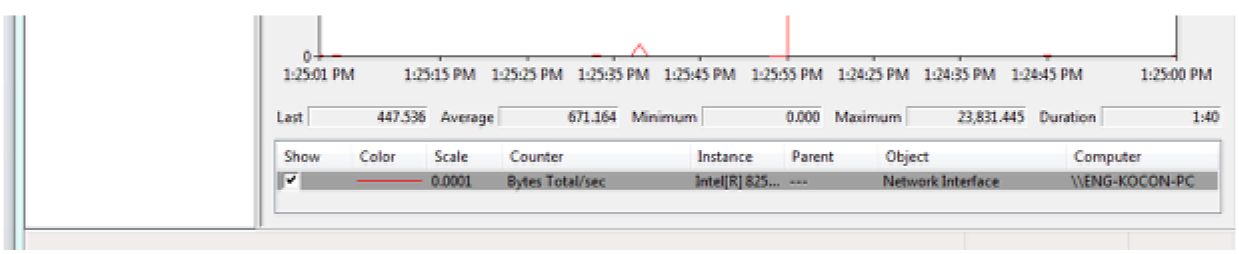
In the left pane, click Performance Monitor to display the real-time monitor.

Add Network Counter



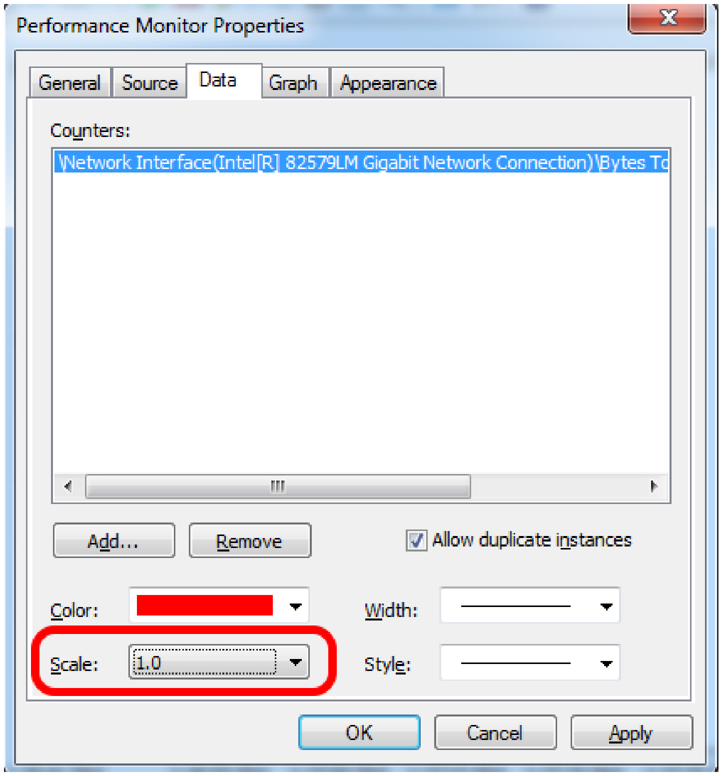
1. Click the green plus near the top of the screen to add a counter
2. In the top left pane, locate and click on **Network Interface** to select it
3. In the bottom left pane, locate the desired network interface (or use All instances to monitor all interfaces)
4. Click **Add>>** to add the counter to the right pane.
5. Click **OK** to add the counters to the graph.

Remove extra counters



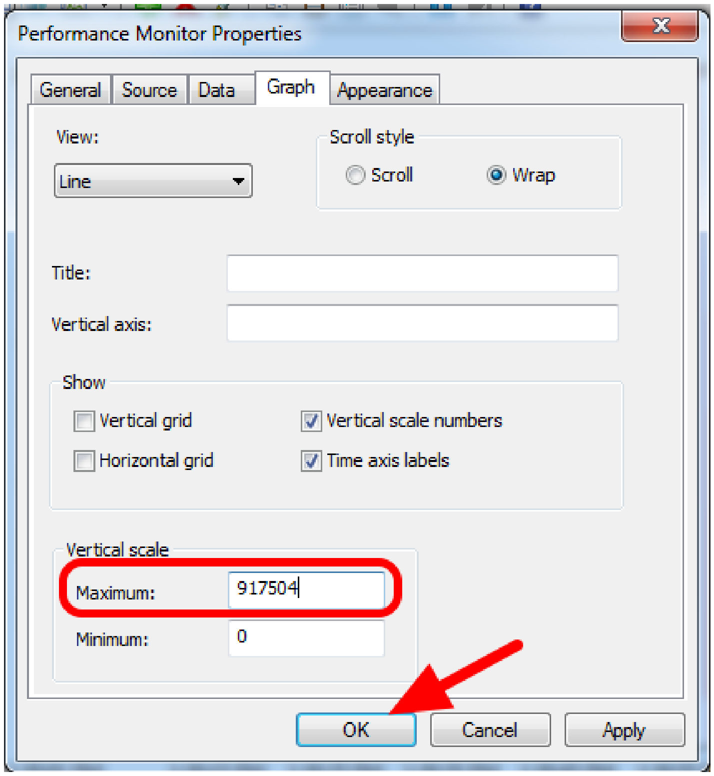
In the bottom pane, select each counter other than **Bytes Total/sec** and press the **Delete** key. The **Bytes Total/sec** entry should be the only entry remaining in the pane.

Configure Data Properties



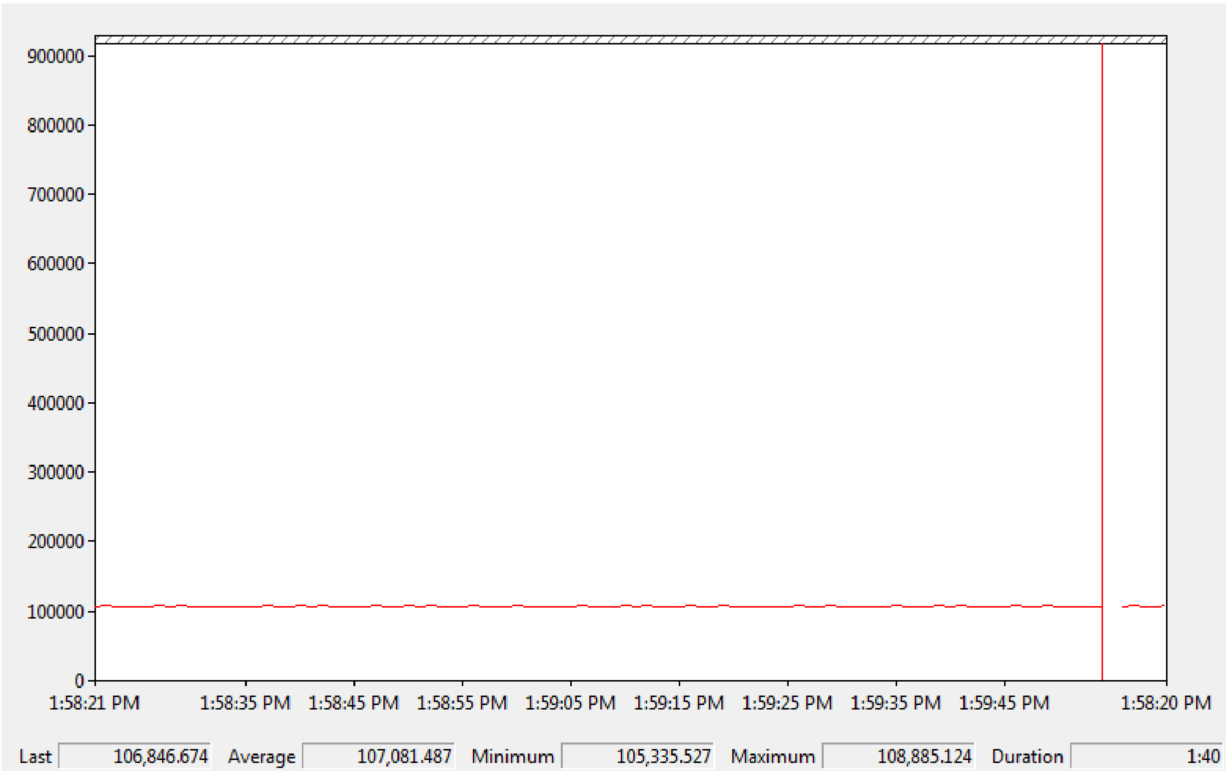
Press **Ctrl+Q** to bring up the Properties window. Click on the dropdown next to **Scale** and select **1.0**. Then click on the **Graph** tab.

Configure Graph Properties



In the **Maximum Box** under **Vertical Scale** enter 917504 (this is 7Megabits converted to Bytes). If desired, turn on the horizontal grid by checking the box. The click **OK** to close the dialog.

Viewing Bandwidth Usage

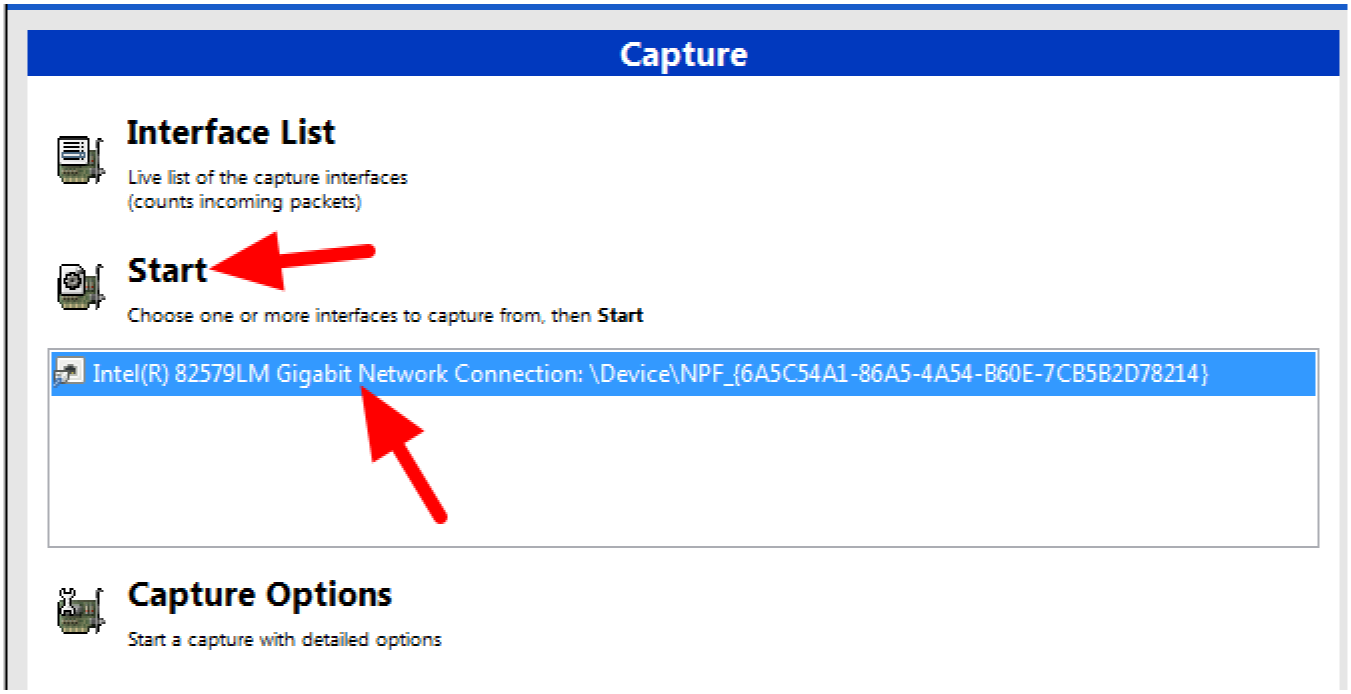


You may now connect to your robot as normal over the selected interface (if you haven't done so already). The graph will show the total bandwidth usage of the connection, with the bandwidth cap at the top of the graph. The Last, Average, Min and Max values are also displayed at the bottom of the graph. Note that these values are in Bytes/Second meaning the cap is 917,504. With just the Driver Station open you should see a flat line at ~100000 Bytes/Second.

Measuring Bandwidth Usage using Wireshark

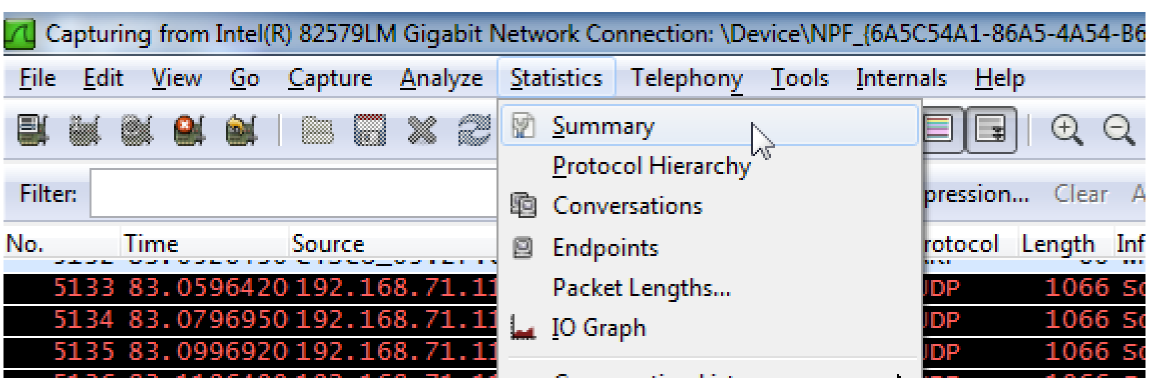
If you are not using Windows 7, you will need to install a 3rd party program to monitor bandwidth usage. One program that can be used for this purpose is Wireshark. Download and install the latest version of Wireshark for your version of Windows. After installation is complete, locate and open Wireshark. Connect your computer to your robot, open the Driver Station and any Dashboard or custom programs you may be using.

Select the interface and Start capture



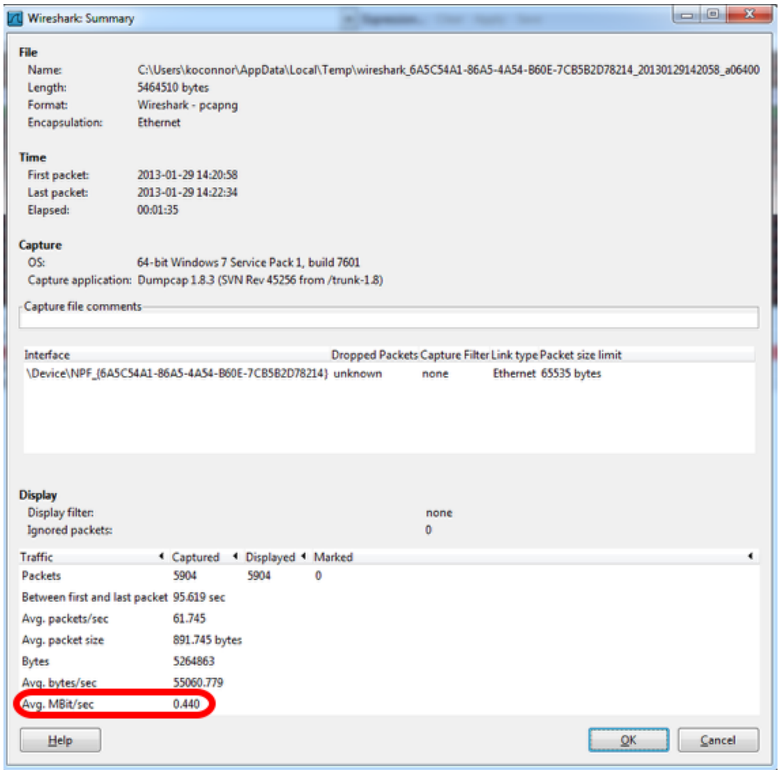
In the Wireshark program on the left side, select the interface you are using to connect to the robot and click **Start.**

Open Statistics Summary



Let the capture run for at least 1 minute, then click **Statistics>>Summary**.

View Bandwidth Usage



Average bandwidth usage, in Megabits/Second is displayed near the bottom of the summary window. The bandwidth cap on the field is 7 Megabits/second.

RoboRIO Network Troubleshooting

The roboRIO and the 2015 FRC tools use dynamic IP addresses (DHCP) for network connectivity. This article describes steps for troubleshooting networking connectivity between your PC and your roboRIO

Ping roboRIO

The first step to identifying roboRIO networking issues is to isolate if it is an application issue or a general network issue. To do this, click Start->type **cmd**->press Enter to open the command prompt. Type **ping roboRIO-####-FRC.local** where #### is your team number (with no leading zeroes) and press enter. If the ping succeeds, the issue is likely with the specific application, verify your team number configuration in the application, and check your firewall configuration.

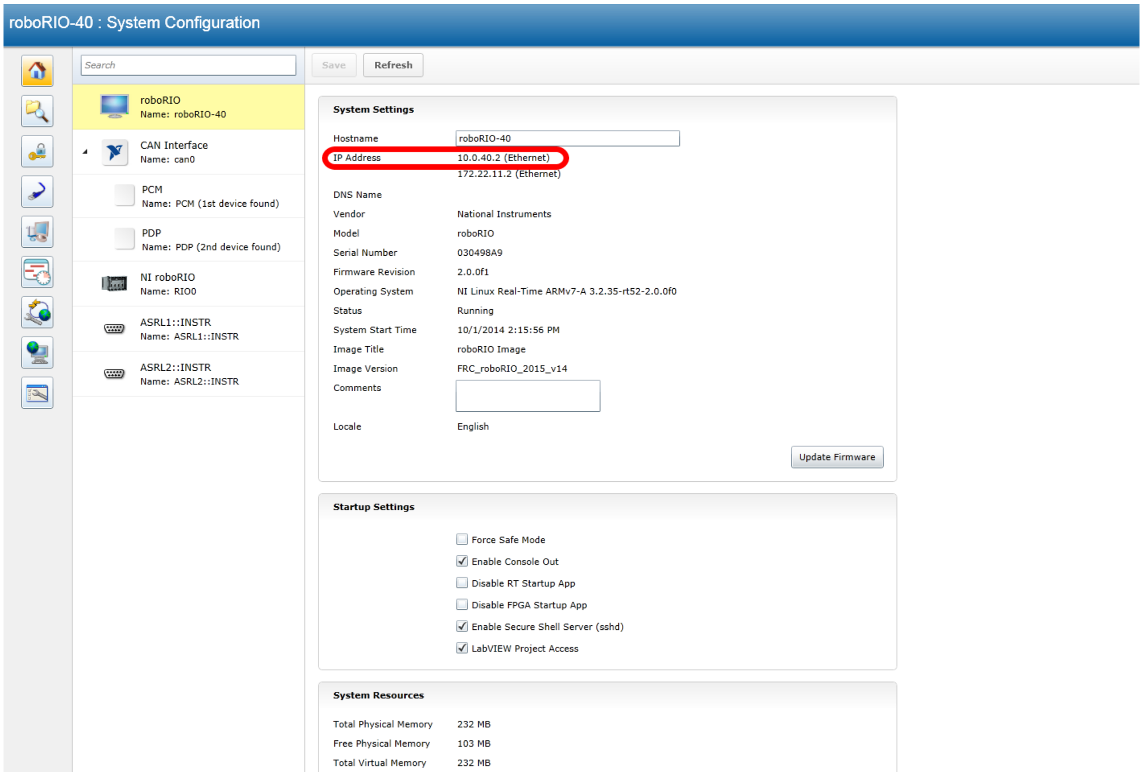
USB Connection Troubleshooting

If you are attempting to troubleshoot the USB connection, try pinging the roboRIO's IP address. As long as there is only one roboRIO connected to the PC, it should be configured as 172.22.11.2. If this ping fails, make sure you have the roboRIO connected and powered, and that you have installed the NI FRC Update Suite. This update installs the roboRIO drivers needed for the USB connection.

If this ping suceeds, but the .local ping fails, it is likely that either the roboRIO hostname is configured incorrectly,, or you are connected to a DNS server which is attempting to resolve the .local address.

- Verify that your roboRIO has been imaged for your team number. This sets the hostname used by mDNS.
- Disconnect your computer from all other networks including Ethernet and WiFi. It is possible that one of these networks contains a DNS server that is attempting to resolve the .local address.

Ethernet Connection



If you are troubleshooting an Ethernet connection, it may be helpful to first make sure that you can connect to the roboRIO using the USB connection. Using the USB connection, open the roboRIO webdashboard and verify that the roboRIO has an IP address on the ethernet interface. If you are tethering to the roboRIO directly this should be a self-assigned 169.*.*.* address, if you are connected to the OM5P-AN radio, it should be an address of the form 10.TE.AM.XX where TEAM is your four digit FRC team number. If the only IP address here is the USB address, verify the physical roboRIO ethernet connection.

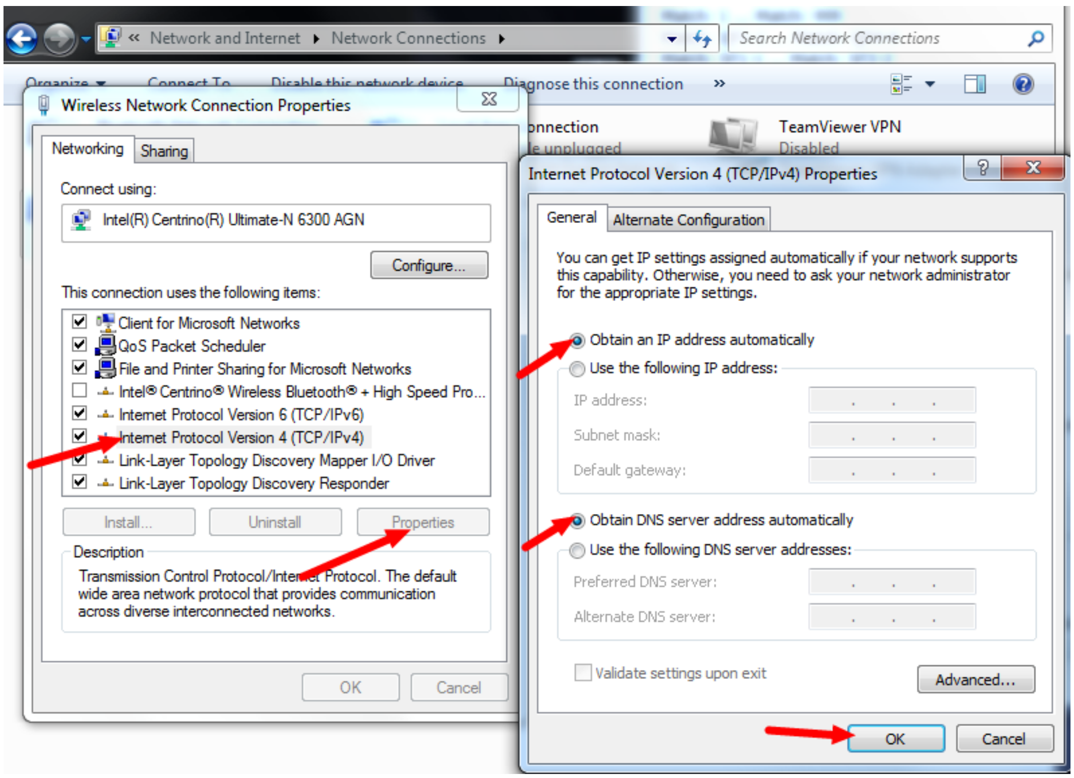
Ping the roboRIO IP address

If there is an IP address in the step above, try pinging this IP address using the command prompt as described above. If this works, you have an issue resolving the mDNS address on your PC. The two most common causes are not having an mDNS resolver installed on the system and a DNS server on the network that is trying to resolve the .local address using regular DNS.

- Verify that you have an mDNS resolver installed on your system. On Windows, this is typically fulfilled by the NI FRC Update Suite. For more information on mDNS resolvers, see the RoboRIO Networking article.

- Disconnect your computer from any other networks and make sure you have the OM5P-AN configured as an access point, using the FRC Bridge Configuration Utility. Removing any other routers from the system will help verify that there is not a DNS server causing the issue.

Ping fails



If pinging the IP address directly fails, you may have an issue with the network configuration of the PC. The PC should be configured to Obtain an Address Automatically (also known as DHCP). To check this, click Start->Control Panel->Network Connections->Change adapter settings, then right click on the appropriate interface (usually Local Area Connection for Ethernet or Wireless Network Connection for wireless) and select Properties. Click Internet Protocol Version 4, then click Properties. Make sure both radio buttons are set to Obtain automatically.

Other things to check

Other possibilities that may cause issues include:

- Proxies. Having a proxy enabled may cause issues with the roboRIO networking.

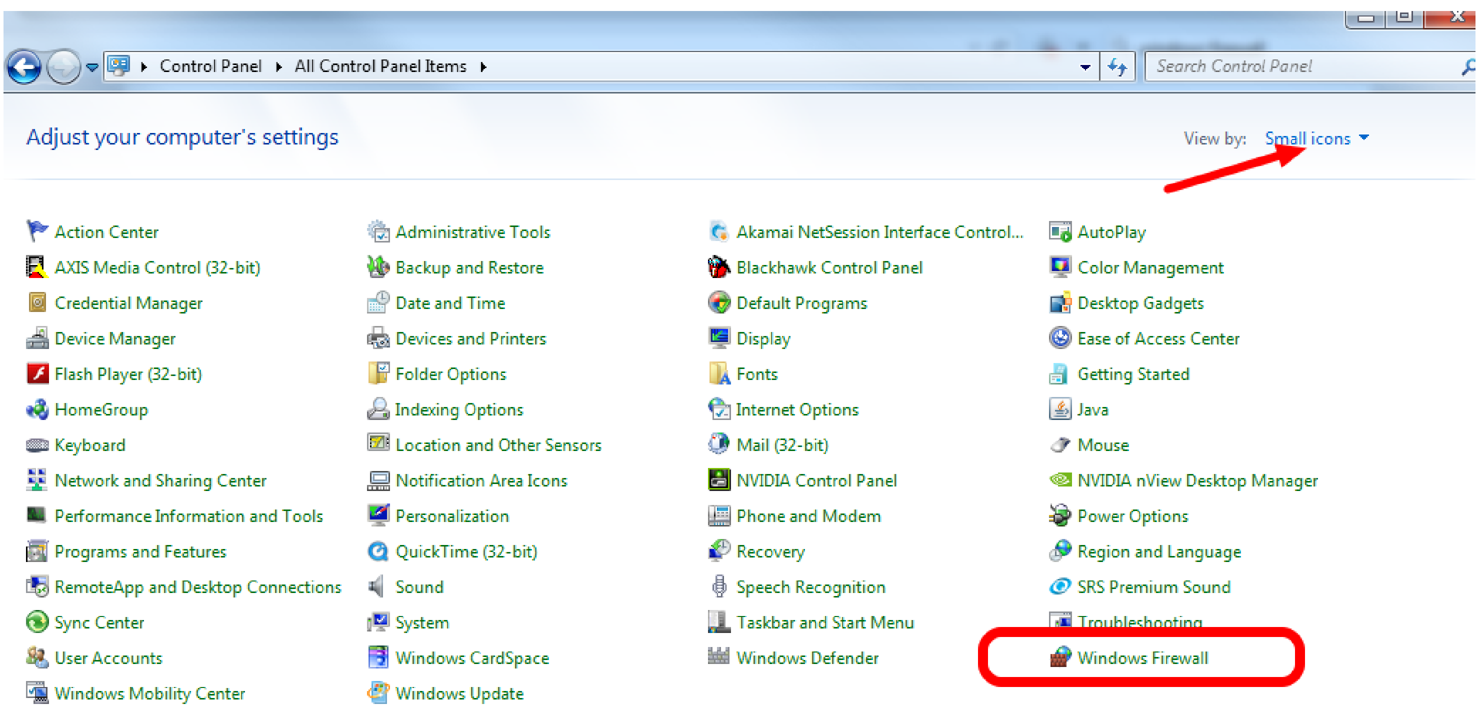
Windows Firewall Configuration

Many of the programming tools used in FRC need network access for various reasons. Depending on the exact configuration, the Windows Firewall may potentially interfere with this access for one or more of these programs. This document describes procedures for Windows 7, but Windows 8 should be similar.

Disabling Windows Firewall

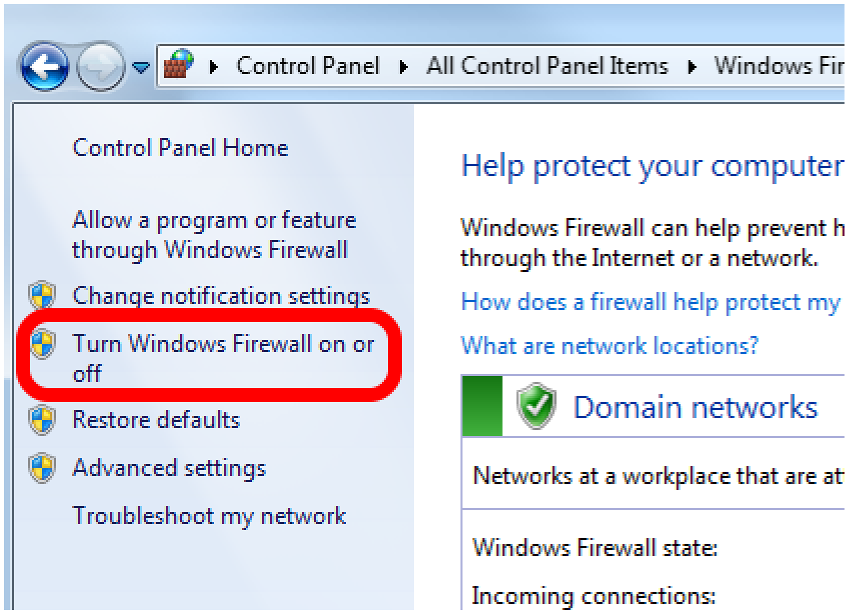
The easiest solution is to disable the Windows Firewall. Teams should beware that this does make the PC potentially more vulnerable to malware attacks if connecting to the internet.

Control Panel



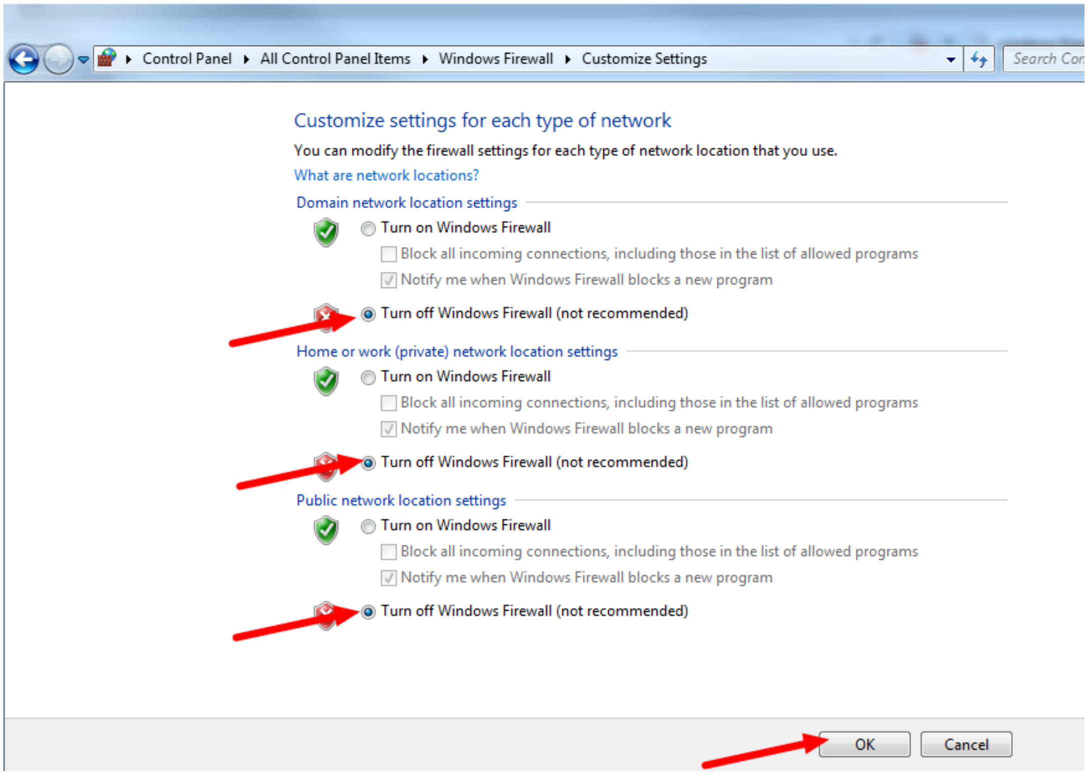
Click Start->Control Panel to open the Control Panel. Click the dropdown next to **View by:** and select **Small icons** then click **Windows Firewall**.

Turn Windows Firewall on or off



In the left pane, click **Turn Windows Firewall on or off,** and click yes or enter your Administrator password if a dialog appears.

Disable the Firewall

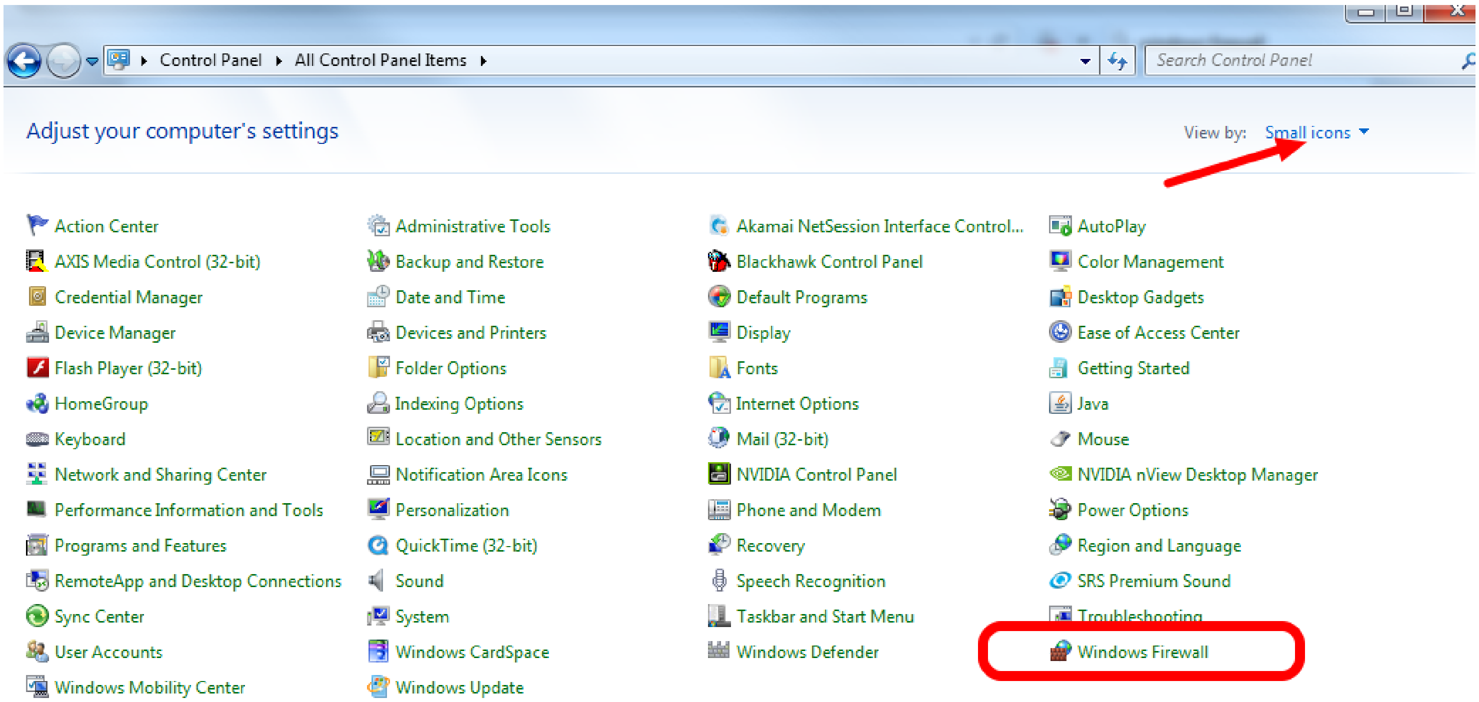


For each category, select the radio button to **Turn off Windows Firewall**. Then click **OK.**

Configure the firewall

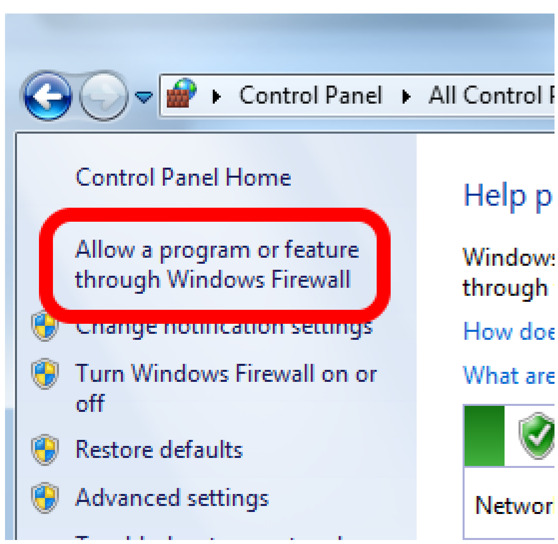
Alternatively, you can add exceptions to the Firewall for any FRC programs you are having issues with.

Control Panel



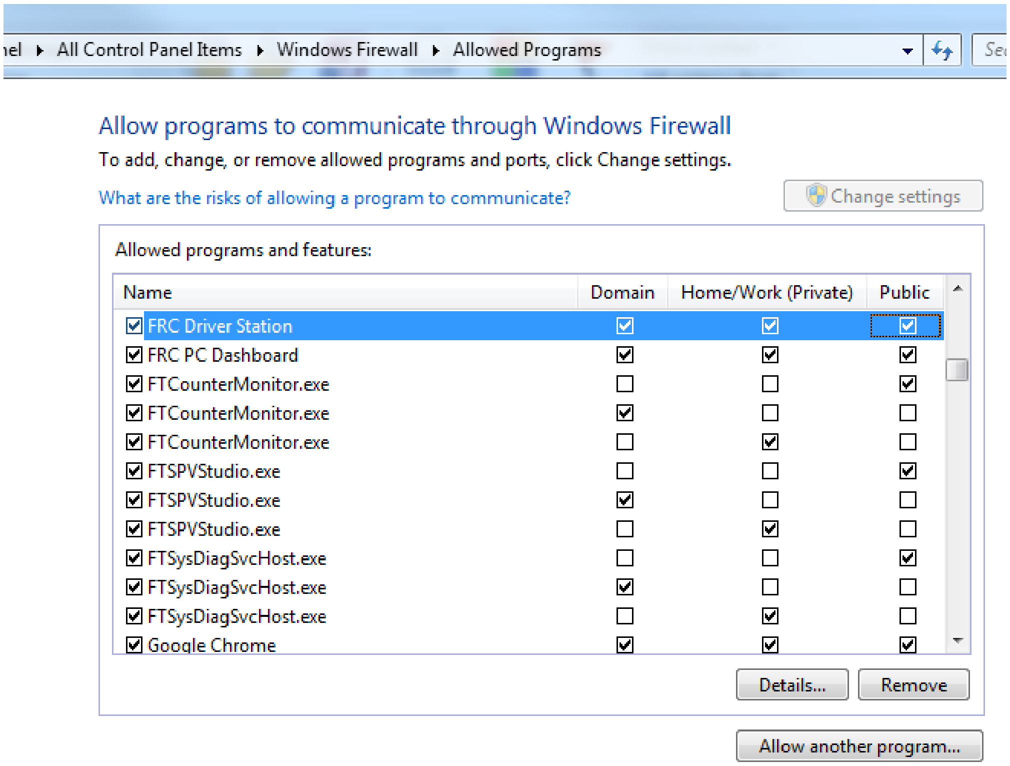
Click Start->Control Panel to open the Control Panel. Click the dropdown next to **View by:** and select **Small icons** then click **Windows Firewall**.

Allow a program...



In the left pane, click **Allow a program or feature through Windows Firewall**

Allowed Programs

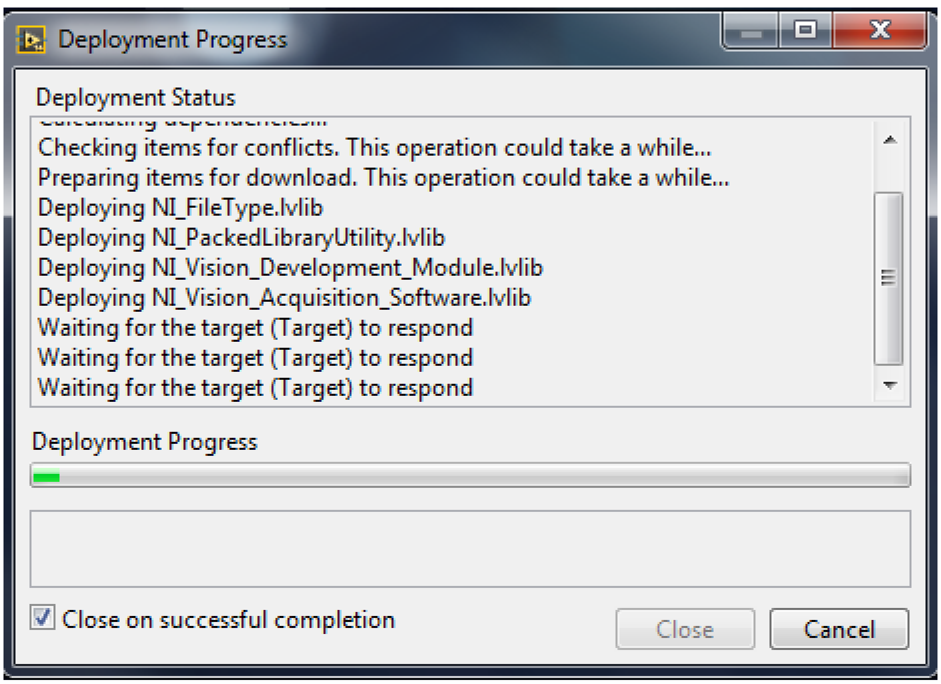


For each FRC program you are having an issue with, make sure that it appears in the list and that it has a check in each of the 3 columns. If you need to change a setting, you made need to click the **Change settings** button in the top right before changing the settings. If the program is not in the list at all, click the **Allow another program...** button and browse to the location of the program to add it.

Waiting for Target to Respond - Recovering from bad loops

If you download LabVIEW code which contains an unconstrained loop (a loop with no delay) it is possible to get the roboRIO into a state where LabVIEW is unable to connect to download new code. This document explains the process required to load new, fixed, code to recover from this state.

The Symptom

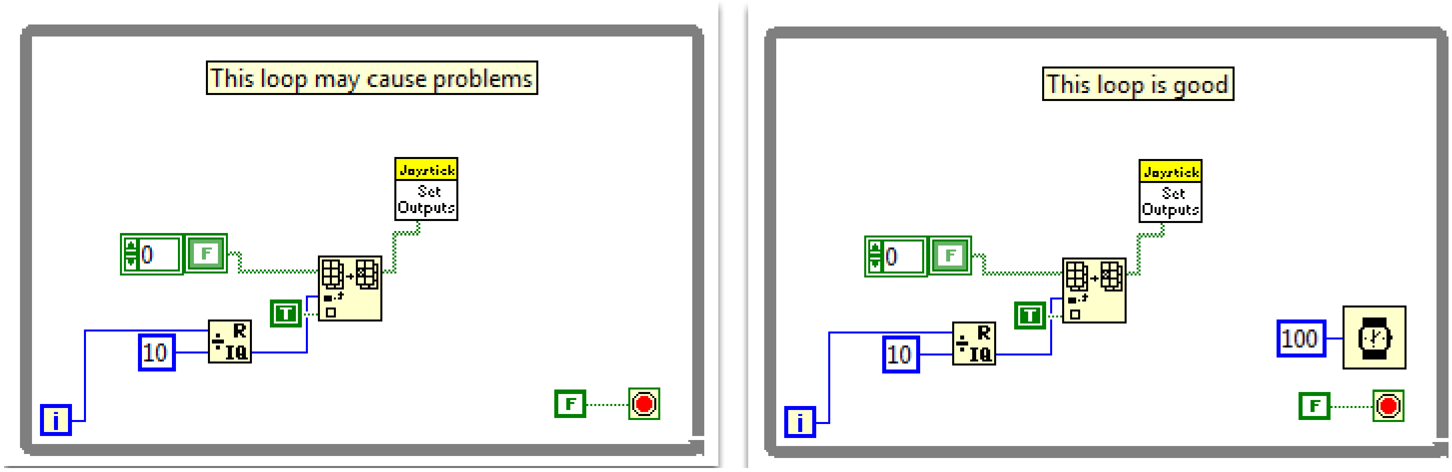


The primary symptom of this issue is attempts to download new robot code hang at the "Waiting for the target (Target) to respond" step as shown above. Note that there are other possible causes of this symptom (such as switching from a C++\Java program to LabVIEW program) but the steps described here should resolve most or all of them.

Click Cancel to close the download dialog.

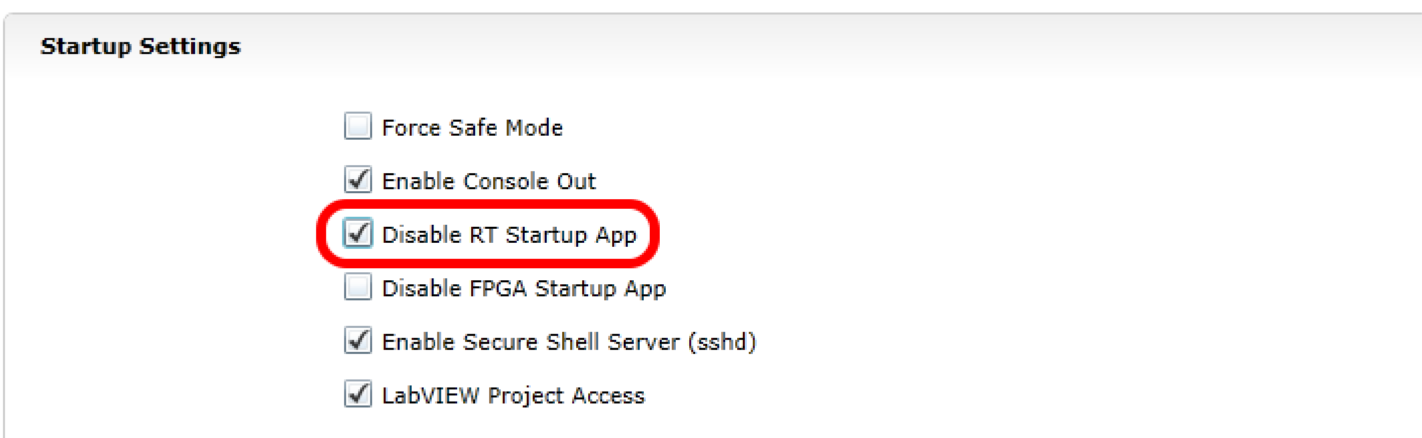
Troubleshooting

The Problem



One common source of this issue is unconstrained loops in your LabVIEW code. An unconstrained loop is a loop which does not contain any delay element (such as the one on the left). If you are unsure where to begin looking, Disabled.VI, Periodic Tasks.VI and Vision Processing.VI are the common locations for this type of loop. To fix the issue with the code, add a delay element such as the Wait (ms) VI from the Timing palette, found in the right loop.

Set No App

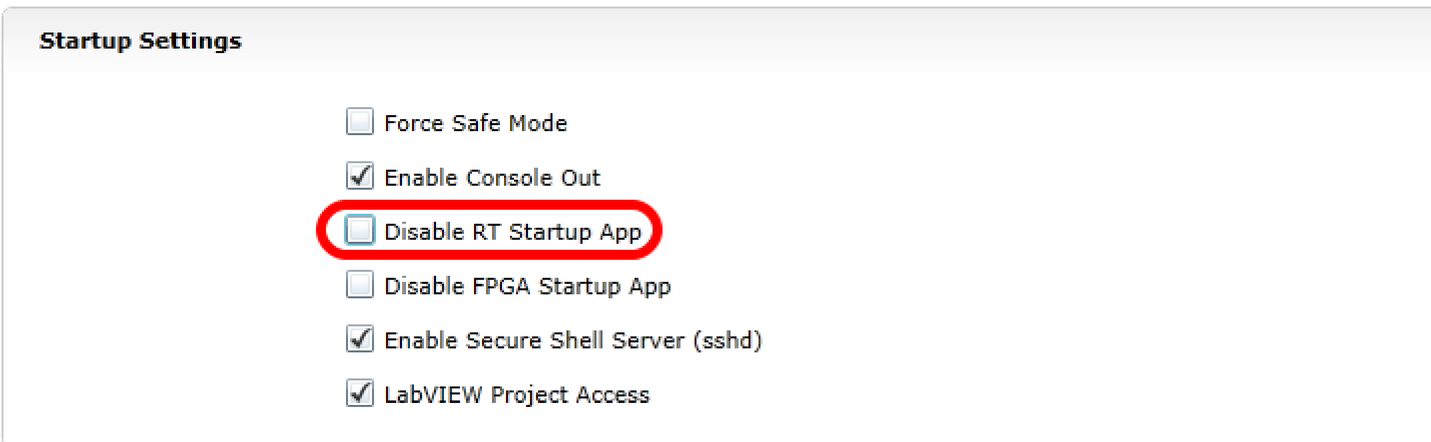


Using the roboRIO webserver (see the article RoboRIO Webdashboard for more details). Check the box to "Disable RT Startup App".

Reboot

Reboot the roboRIO, either using the Reset button on the device or by click Restart in the top right corner of the webpage.

Clear No App



Using the roboRIO webserver (see the article RoboRIO Webdashboard for more details). Uncheck the box to "Disable RT Startup App".

Load LabVIEW Code

Load LabVIEW code (either using the Run button or Run as Startup). Make sure to set LabVIEW code to Run as Startup before rebooting the roboRIO or you will need to follow the instructions above again.

Recovering a roboRIO using Safe Mode

Occasionally a roboRIO may become corrupted to the point that it cannot be recovered using the normal boot and imaging process. Booting the roboRIO into Safe Mode may allow the device to be successfully re-imaged.

Booting into Safe Mode

To boot the roboRIO into Safe Mode:

1. Apply power to the roboRIO
2. Press and hold the Reset button until the Status LED lights up (~5 seconds) then release the Reset button
3. The roboRIO will boot in Safe Mode (indicated by the Status LED flashing in groups of 3)

Recovering the roboRIO

The roboRIO can now be imaged by using the roboRIO Imaging Tool as described in Imaging your roboRIO.

About Safe Mode

In Safe Mode, the roboRIO boots a separate copy of the operating system into a RAM Disk. This allows you to recover the roboRIO even if the normal copy of the OS is corrupted. While in Safe Mode, any changes made to the OS (such as changes made by accessing the device via SSH or Serial) will not persist to the normal copy of the OS stored on disk.

At The Event

IP Networking at the Event

This document describes the IP configuration used at events, both on the fields and in the pits, potential issues and workaround configurations.

TE.AM IP Notation

The notation TE.AM is used as part of IPs in numerous places in this document. This notation refers to splitting your four digit team number into two digit pairs for the IP address octets.

Example: 10.TE.AM.2

Team 12 - 10.0.12.2

Team 122 - 10.1.22.2

Team 1212 - 10.12.12.2

Team 3456 - 10.34.56.2

On the Field

This section describes networking when connected to the Field Network for match play

DHCP (typical configuration)

The Field Network runs a DHCP server with pools for each team that will hand our addresses in the range of 10.TE.AM.20 and up with subnet masks of 255.0.0.0

- OpenMesh OM5P-AN or OM5P-AC radio - Static 10.TE.AM.1 programmed by Kiosk
- roboRIO - DHCP 10.TE.AM.W assigned by field
- Driver Station - DHCP ("Obtain an IP address automatically") 10.TE.AM.X assigned by field
- IP camera (if used) - DHCP 10.TE.AM.Y assigned by field (note, this will not currently work with the SmartDashboard camera viewer)
- Other devices (if used) - DHCP 10.TE.AM.Z assigned by field

Static (workaround configuration)

It is also possible to configure static IPs on your devices 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. These addresses are 10.TE.AM.1 and 10.TE.AM.4 for the OpenMesh radio and the field access point and anything 10.TE.AM.20 and up which may be assigned to a device still configured for DHCP. The roboRIO network configuration can be set from the webdashboard.

- OpenMesh radio - Static 10.TE.AM.1 programmed by Kiosk
- roboRIO - Static 10.TE.AM.2 would be a reasonable choice, subnet mask of 255.255.255.0 (default)
- Driver Station - Static 10.TE.AM.5 would be a reasonable choice, **subnet mask must be 255.0.0.0**
- IP Camera (if used) - Static 10.TE.AM.11 would be a reasonable choice, subnet 255.255.255.0 should be fine
- Other devices - Static 10.TE.AM.6-.10 or .12-.19 (.11 if camera not present) subnet 255.255.255.0

In the Pits

In the pits at the event there is no DHCP server present in the typical configuration.

DHCP (typical configuration)

- OpenMesh radio - Static 10.TE.AM.1 programmed by Kiosk. You will not be able to communicate with the radio itself (the radio ping light in the DS will be off) but this is expected and not an issue.
- roboRIO - DHCP or Link Local, falls back to Link Local 169. address with no DHCP server
- Driver Station - DHCP ("Obtain an IP address automatically"), falls back to Link Local 169. address with no DHCP server
- IP camera (if used) - DHCP falls back to Link Local 169. address with no DHCP server (note, this will not currently work with the SmartDashboard camera viewer)
- Other devices (if used) - DHCP or Link Local falls back to Link Local 169. address with no DHCP server requires the device to support Link Local fallback and mDNS addressing (such as using the avahi service on a Linux device)

DHCP + DHCP Server

An alternative if one or more devices are set static but others are set DHCP is to provide a DCHP server that serves 10.TE.AM.X addresses such as a D-Link radio **(with the wireless turned off!)** or other router. **Make sure any devices used are not broadcasting wireless signals.**

In this case the addressing will look just like the field DHCP configuration.

Static (workaround configuration)

It is also possible to configure static IPs on your devices 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. These addresses are 10.TE.AM.1 and 10.TE.AM.4 for the OpenMesh radio and the field access point and anything 10.TE.AM.20 and up which may be assigned to a device still configured for DHCP. The roboRIO network configuration can be set from the webdashboard.

- OpenMesh radio - Static 10.TE.AM.1 programmed by Kiosk
- roboRIO - Static 10.TE.AM.2 would be a reasonable choice, subnet mask of 255.255.255.0 (default)
- Driver Station - Static 10.TE.AM.5 would be a reasonable choice, **subnet mask must be 255.0.0.0**
- IP Camera (if used) - Static 10.TE.AM.11 would be a reasonable choice, subnet 255.255.255.0 should be fine
- Other devices - Static 10.TE.AM.6-.10 or .12-.19 (.11 if camera not present) subnet 255.255.255.0

Troubleshooting

The most common issue is to have a mix of static and DHCP configured devices. This may result in things working on the field (when a DHCP server is present) but not working in the pit, when any DHCP devices fall back to Link Local addresses and cannot communicate with devices configured with a static IP.

Another issue is if the roboRIO is not powered cycled (or at least pull the ethernet cable and re-insert) after connecting to the field before trying to tether to the robot. In this case, the roboRIO still has a 10.TE.AM address (as it has had a continuous ethernet connection) and the DS computer (which has gone through an interface down and interface up cycle) will fall back to Link Local.

The last common issue is using a subnet mask of 255.255.255.0 on the DS PC. This configuration will not communicate with the FMS system which is on a 10.0.100 address.

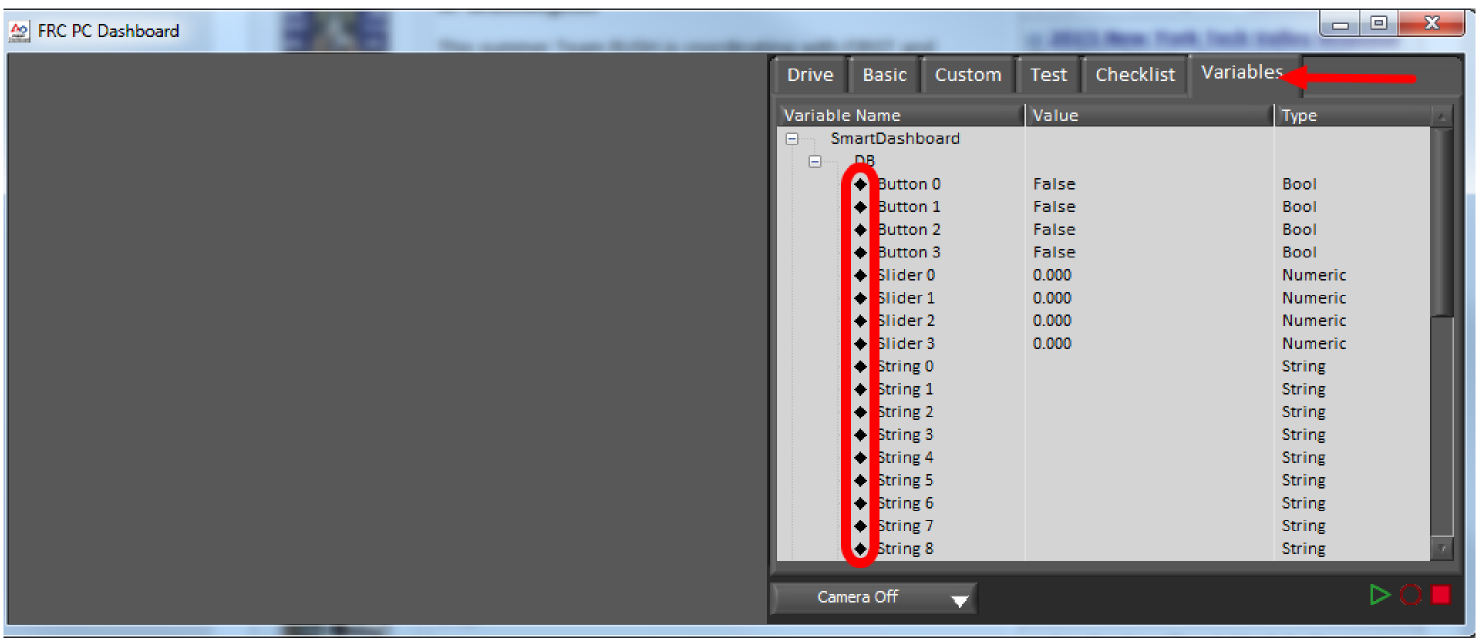
Troubleshooting Dashboard Connectivity

We have received a number of reports of Dashboard connectivity issues from events. This document will help explain how to recognize if the Dashboard is not connected to your robot, steps to troubleshoot this condition and a code modification you can make

LabVIEW Dashboard

This section discusses connectivity between the robot and LabVIEW dashboard

Recognizing Connectivity



If you have an indicator on your dashboard that you expect to be changing it may be fairly trivial to recognize if the Dashboard is connected. If not, there is a way to check without making any changes to your robot code. On the Variables tab of the Dashboard, the variables are shown with a black diamond when they are not synced with the robot. Once the Dashboard connects to the robot and these variables are synced, the diamond will disappear.

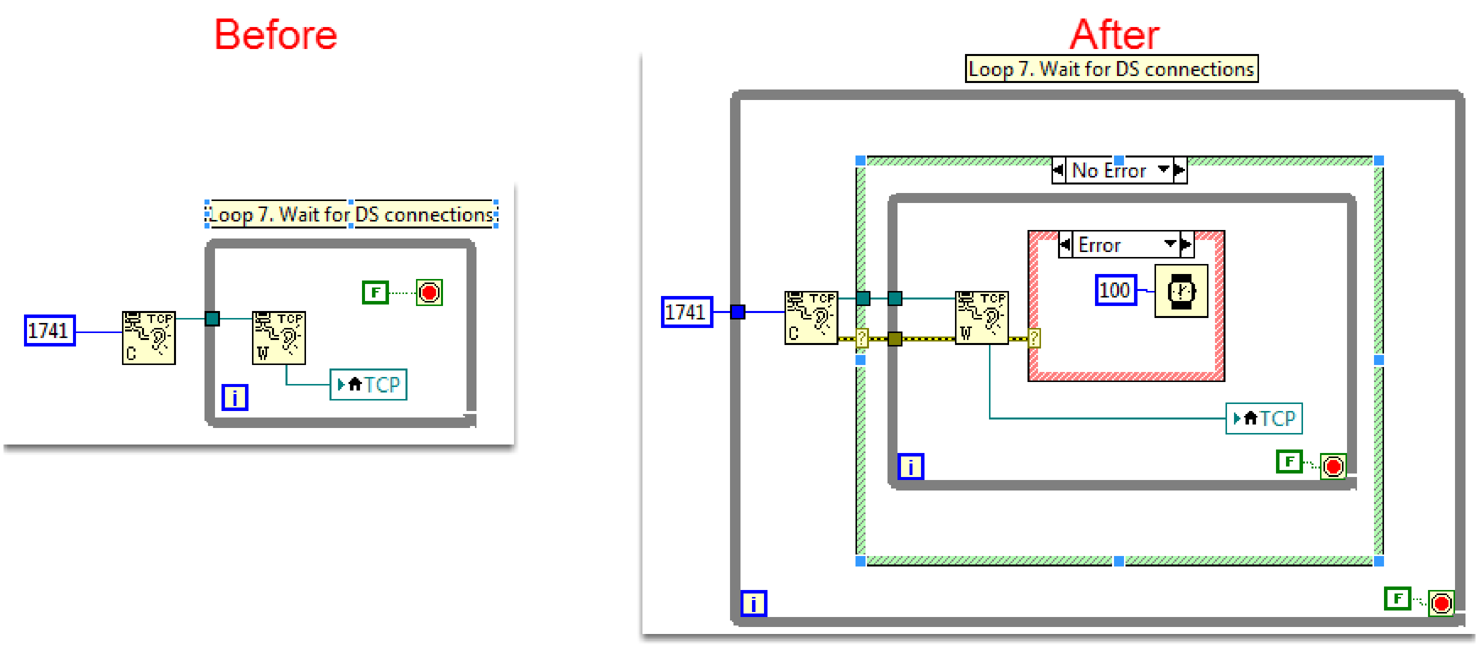
Troubleshooting Connectivity

If the Dashboard does not connect to the Robot (after the Driver Station has connected to the robot) the recommended troubleshooting steps are:

1. Close the Driver Station and Dashboard, then re-open the Driver Station (which should launch the Dashboard).

2. If that doesn't work, restart the Robot Code using the Restart Robot Code button on the Diagnostics tab of the Driver Station

Improving Reliability of a Custom Dashboard

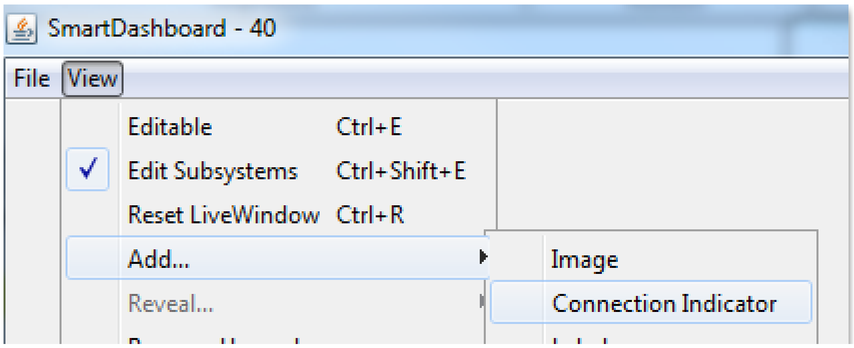


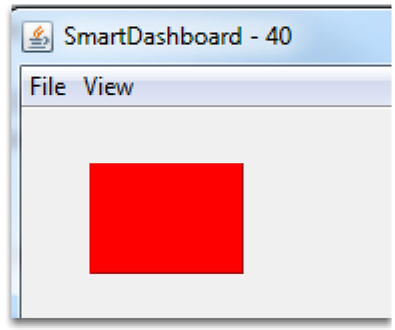
If you have created a custom LabVIEW dashboard there is a tweak you can make to the code to improve reliability of the initial connection. Locate the loop labeled Loop 7 in the Dashboard Main VI. Modify the loop according to the image above by adding a loop around the listener, 2 case statements, a Wait block and error wiring.

SmartDashboard

This section discusses connectivity between the robot and Java SmartDashboard

Recognizing Connectivity





The typical way to recognize connectivity with the Java SmartDashboard is to add a Connection Indicator widget and to make sure your code is writing at least one key during initialization or disabled to trigger the connection indicator. The connection indicator can be moved or re-sized if the Editable checkbox is checked.

Troubleshooting Connectivity

If the Dashboard does not connect to the Robot (after the Driver Station has connected to the robot) the recommended troubleshooting steps are:

1. Restart the SmartDashboard (there is no need to restart the Driver Station software for the Java SmartDashboard)

2. If that doesn't work, restart the Robot Code using the Restart Robot Code button on the Diagnostics tab of the Driver Station

3. If it still doesn't connect verify that the Team Number is set properly in the Dashboard and that your Robot Code writes a SmartDashboard value during initialization or disabled

Programming Radios for FMS Offseason

When using the FMS Offseason software, the typical networking setup is to use a single access point with a single SSID and WPA key. This means that the radios should all be programmed to connect to this network, but with different IPs for each team. The Team version of the FRC Bridge Configuration Utility has an FMS-Lite mode that can be used to do this configuration.

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.

The steps below describe installing and using the FRC Bridge Configuration Utility to program radios for this network configuration. If you have a machine where you have already used this tool to program a radio for FRC 2016, skip to "Launch the Software".

Pre-Requisites

The 2016 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.

Application Notes

The 2016 Radio Kiosk 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.

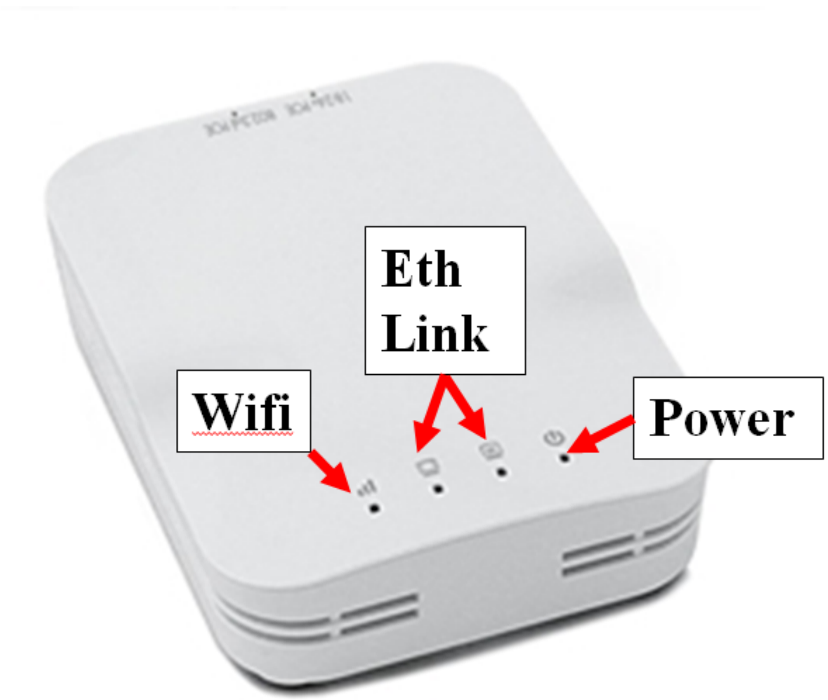
Troubleshooting

The 2016 Kiosk 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	
Red	Bridge Mode, unlinked
Yellow\Orange	Bridge Mode, Linked
Green	AP Mode
Off	Unprogrammed

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
- 7Mb/s bandwidth limit on the outbound side of the wireless interface
- 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).

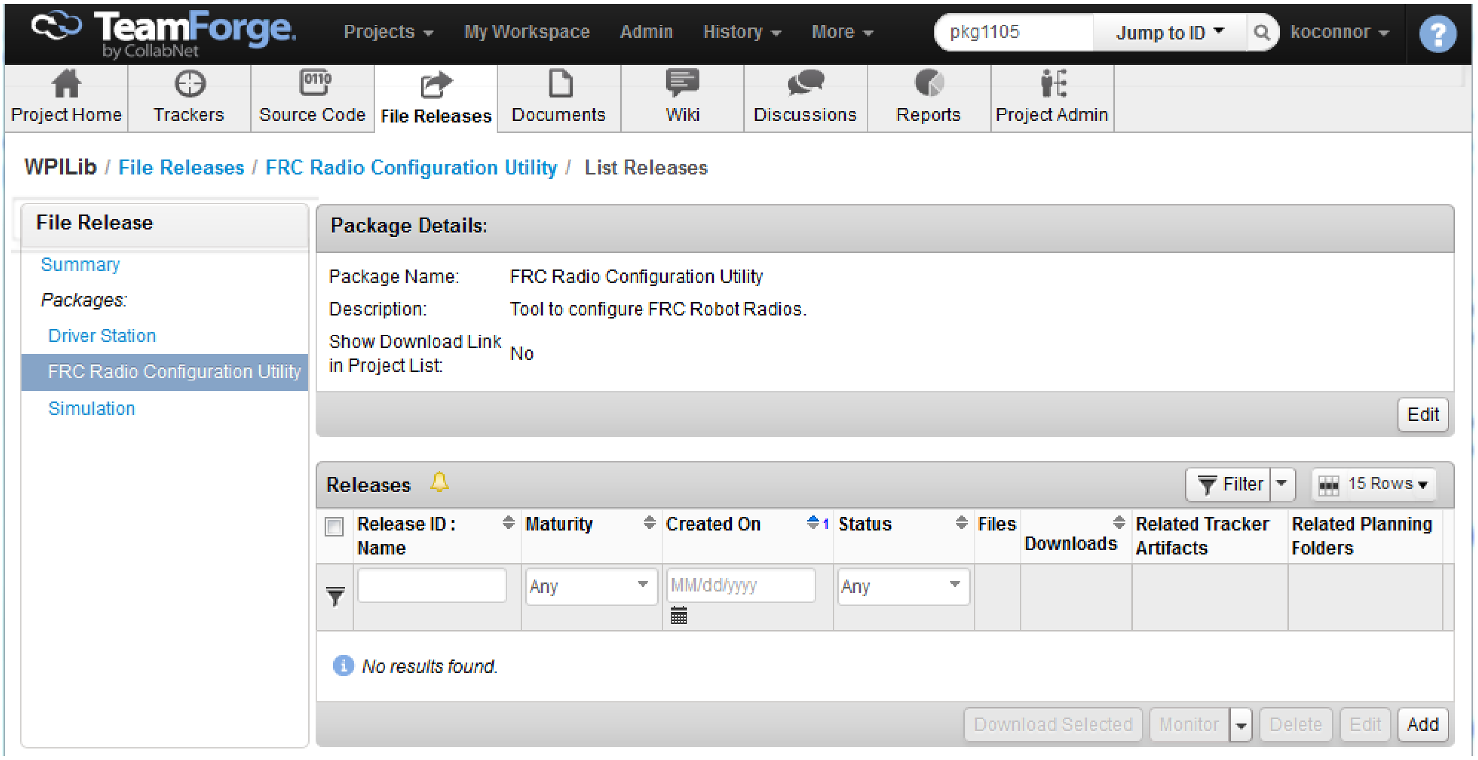
When programmed with the team version of the Radio Configuration Utility, the user accounts will be left at (or set to) the firmware defaults:

- Username: root
- Password: root

Troubleshooting

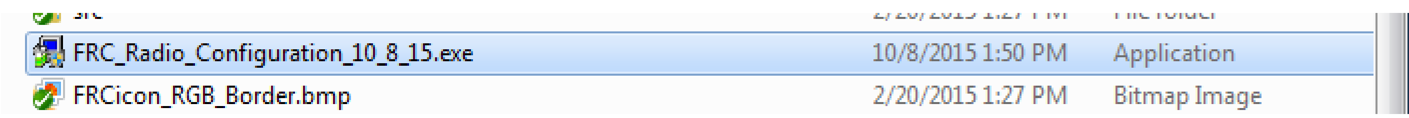
Note: It is not recommended to modify the configuration manually

Download the software



Download the latest FRC Radio Configuration Utility Installer from the WPILib project File Releases.

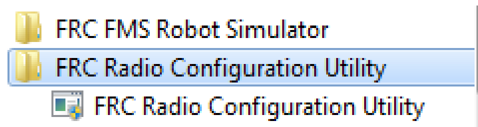
Install the software



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

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

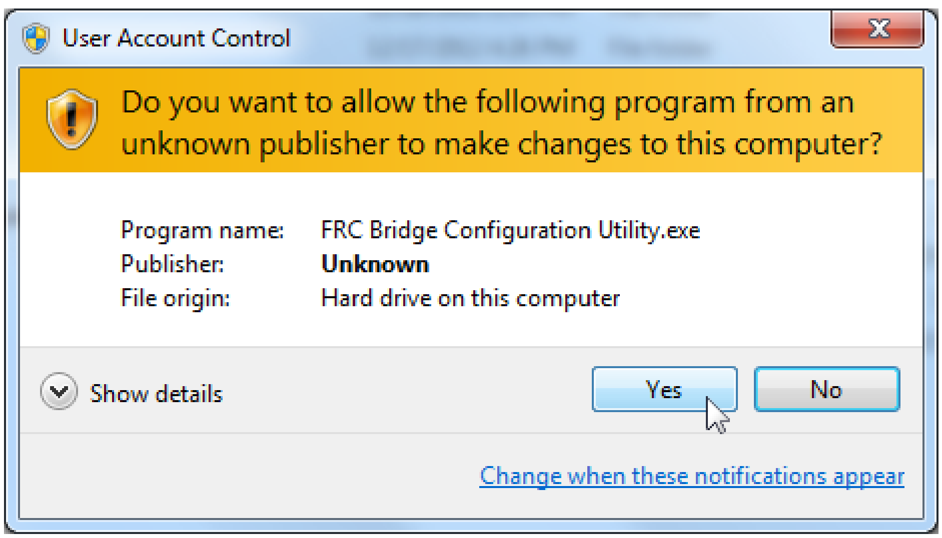
Launch the software



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

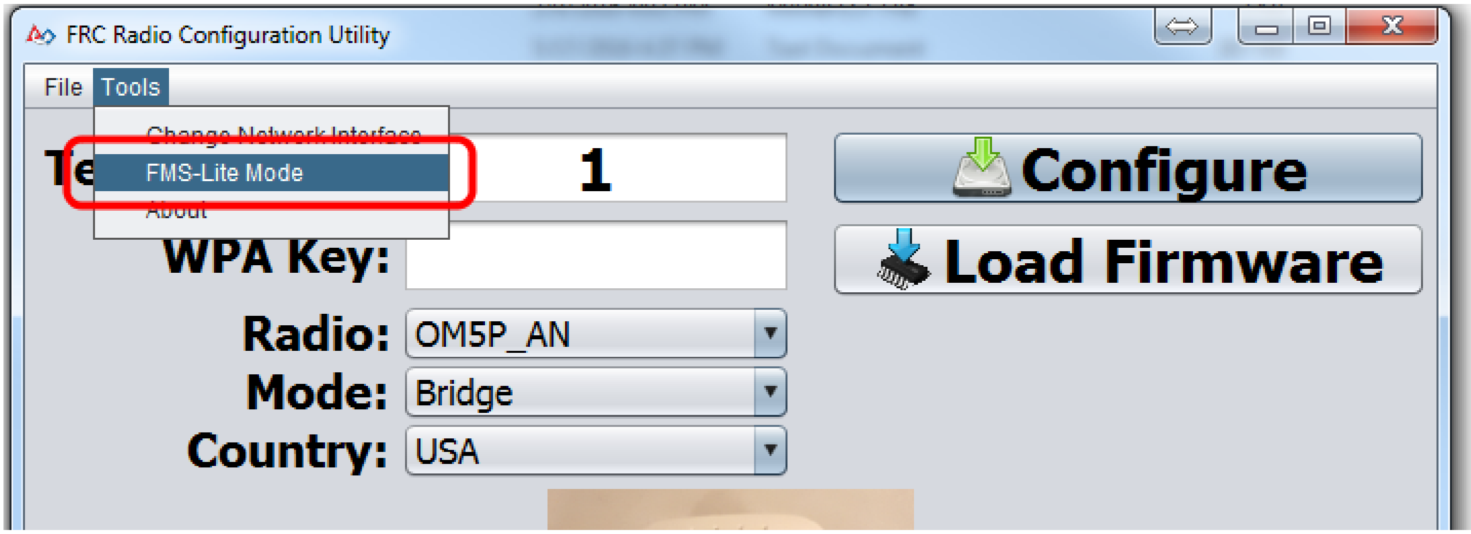
Note: If you need to locate the program it is installed to C:\Program Files (x86)\FRC Radio Configuration Utility. For 32-bit machines the path is C:\Program Files\FRC Radio Configuration Utility\

Allow the program to make changes, if prompted



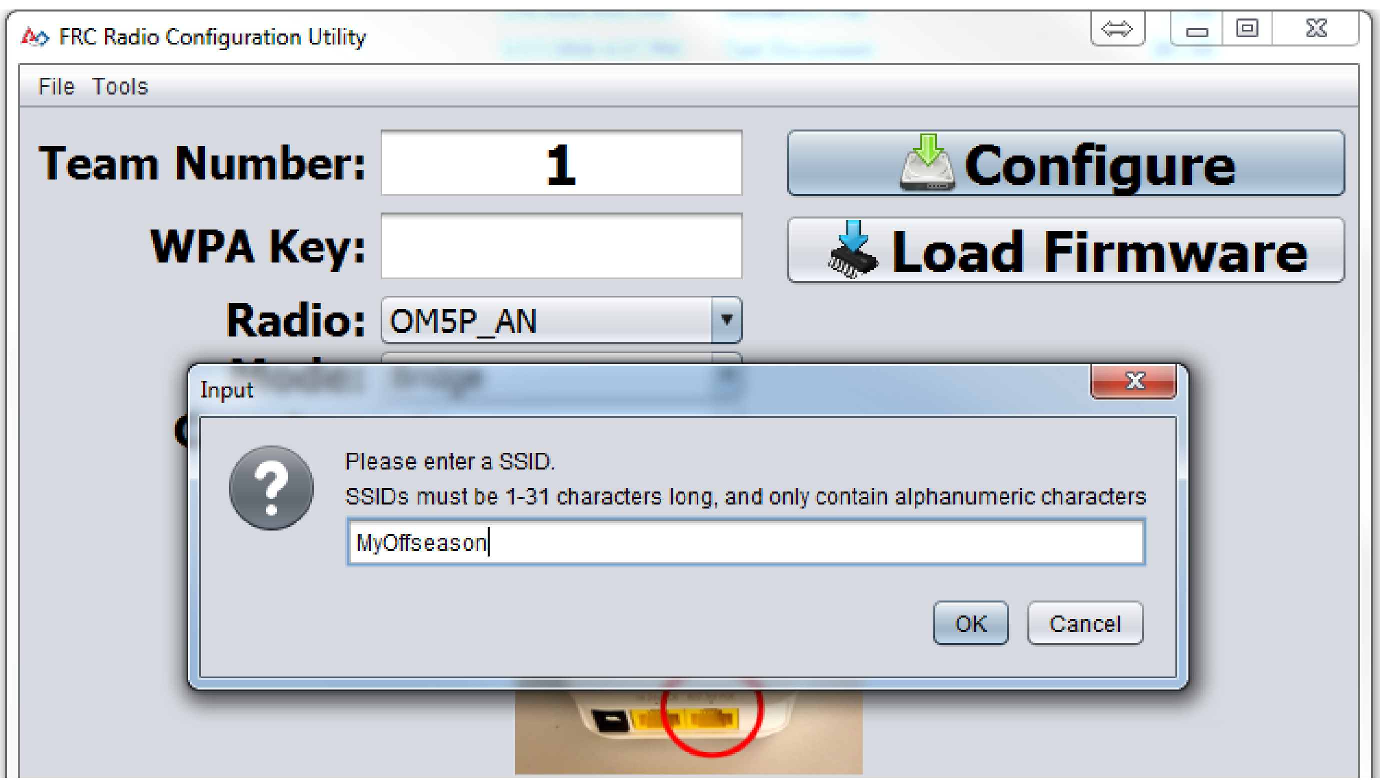
If the your computer is running Windows Vista or Windows 7, a prompt may appear about allowing the configuration utility to make changes to the computer. Click "Yes" if the prompt appears.

Enter FMS-Lite Mode



Click **Tools->FMS-Lite Mode** to enter FMS-Lite Mode.

Enter SSID

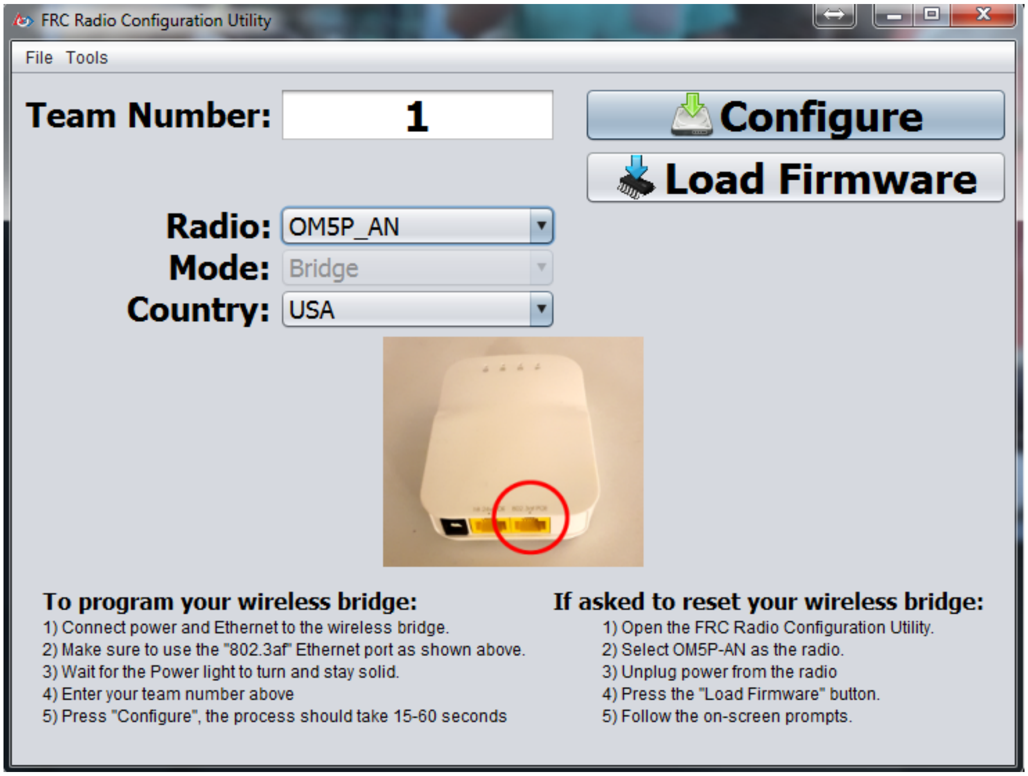


Enter the SSID (name) of your wireless network in the box and click OK.

Enter WPA Key

Enter the WPA key for your network in the box and click OK. Leave the box blank if you are using an unsecured network.

Program Radios



The Kiosk is now ready to program any number of radios to connect to the network entered. To program each radio, connect the radio to the Kiosk, set the Team Number in the box, and click Configure.

The kiosk will program D-Link Rev A or Rev B radios to work on an offseason FMS network by selecting the appropriate option from the "Radio" dropdown. **Note: Bandwidth limitations and QoS will not be configured on the D-Link radios in this mode.**

Changing SSID or Key

If you enter something incorrectly or need to change the SSID or WPA Key, go to the Tools menu and click FMS-Lite Mode to take the kiosk out of FMS-Lite Mode. When you click again to put the Kiosk back in FMS-Lite Mode, you will be re-prompted for the SSID and Key.

FMS Whitepaper

Overview

The Field Management System (FMS) is the electronics core of a *FIRST* Robotics Competition (FRC) playing field. It encompasses all the controls for the field electronics, team robots, and is used to manage the event by creating match schedules, managing all field hardware during a match (timers, team lights, estops, etc.), scoring the matches in real-time, posting information to the Audience screen, and uploading results data to the Internet.

FMS is based on Ethernet architecture. Components such as the Driver Station, or the touchscreens used by the referees, integrate with FMS through direct wired Ethernet interfaces. Devices like the ball counters used in *Breakaway* and *Rebound Rumble*, or the Estops and Stack Lights mounted in each Player Station, interface through Ethernet-based Input/Output (I/O) modules that are donated to *FIRST* by Rockwell Automation. The lights used to illuminate the tower bases in *Logo Motion,* the bridges in *Rebound Rumble,* and the high goals in *Aerial Assist* are controlled via Ethernet-enabled power supplies donated to *FIRST* by Philips Color Kinetics. The weight sensors used in *Ultimate Accen*t were read by a National Instruments cRIO-FRC II and interfaced over the Ethernet network to a Rockwell Automation PLC module, to give some examples.

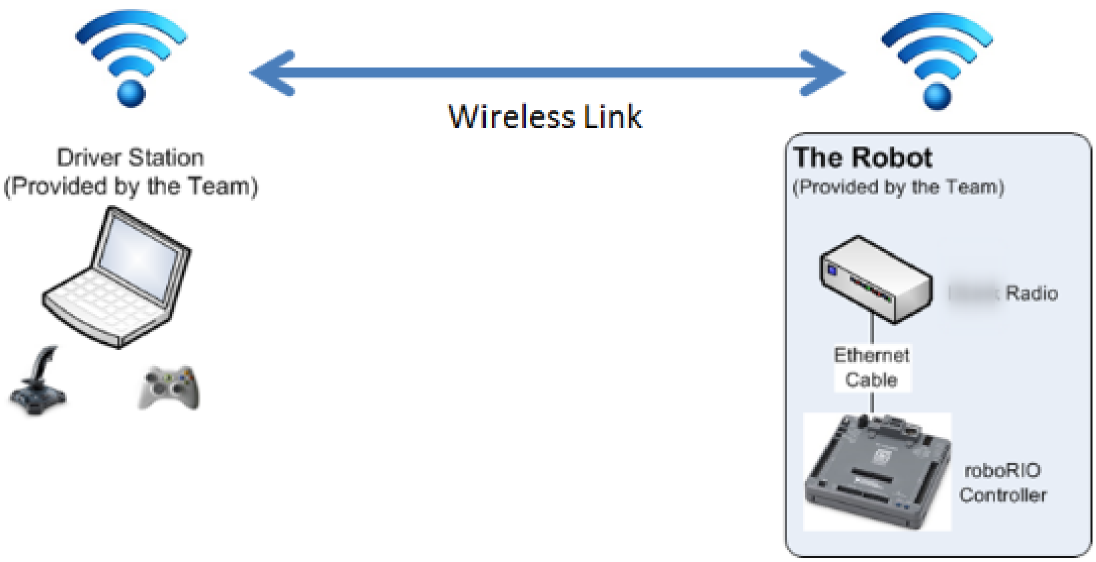
This white paper focuses on the electronics infrastructure needed to control the robots on the playing field. Specific details on the FMS software used during each season can be found in the Field Management System User Guide, publicly available on this site.

Frequently Asked Questions about the Field Management System and recommended best practices when operating on the competition field appear at the end of this document.

Any rules information referenced in this document is for explanation sake only, and the only source of official competition guidelines and rules is the FRC Manual.

This documentation was written using the 2015 revision of the FRC Robot Control System and Field Management System as the model platform. Some improvements to FMS have been introduced in 2015 to coincide with user experience enhancements available when using the roboRIO, however, this system topology has remained mainly the same since 2009.

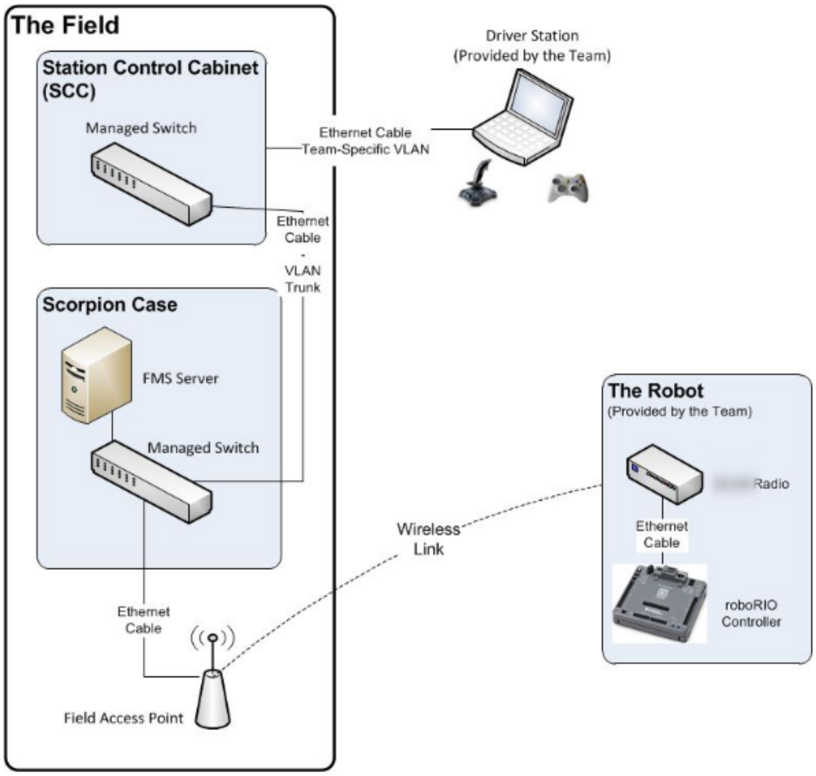
Driver Station Robot Communications - The Basics



The standard configuration for controlling an FRC robot is two core components, the robot itself with a roboRIO controller installed, and a netbook/ ultrabook/ laptop etc. running the FRC Driver Station (DS) software. The FRC Robot Control System is built such that the DS is the master controller, i.e. the status and actions of the robot are determined by commands from the DS.

Communication packets from the DS are broadcast via the integrated radio in the DS or a separate radio (like the WRT610N provided to teams in previous seasons), or through a tether (e.g wired Ethernet or USB.) When operating via wireless link, the DLink radio on the Robot receives the command packets from the DS and forwards them to the roboRIO robot controller. Status packets from the roboRIO are sent to the DS after each received command packet.

On-Field Communications Path for a Single Team



Driver Station Robot Communications

Communication packets from a Driver Station are routed through the managed switches in the Station Control Cabinet and Scorpion Case to the Field Access Point (AP), which then transmits the packets via wireless link to the appropriate Robot.

The Radio on the Robot receives the wireless transmissions from the Field AP, and forwards the packets to the roboRIO robot controller. Status packets from the roboRIO are sent to the DS (via the Field AP) after each received command packet.

Field Management System (FMS) Role

The FMS software (running on the FMS Server) communicates with each Driver Station via the managed switches in the Scorpion Case and Station Control Cabinets. This communication employs team-specific virtual local area networks (VLANs) which serve to isolate each teams data traffic.

The FMS software does not communicate with the robots directly. Instead, FMS gathers data from the Driver Station about the Robot's status, and tells the Driver Station which state (enabled/

disabled/ e-stopped) and mode (autonomous/ teleoperated) the robot should be in, as well as the player station and alliance color.

The FMS software configures the managed switches and Access Point before each match to ensure the data and communications for each team is kept separate from others (a process called "Pre-Start").

Frequently Asked Questions

Does FMS control the robot?

No, FMS does not communicate with the Robots directly. On the playing field, FMS communicates exclusively with each DS, sending it commands for enable/disable, auto/teleop, Estop, player station number and alliance color. The DS then sends this data to the Robot.

What does the flashing Player Station light mean?

The flashing light in the Player Station indicates that FMS does not have the necessary information to confirm your DS has a connection to your Robot. There are two typical ways for this condition to occur: the DS is not communicating with FMS (e.g. the Ethernet cable in the Player Station is not connected), or the DS is telling FMS it does not have communication with the Robot.

What information does FMS log?

FMS combines the status data from each DS along with the data that is monitored from the field components and stores this data in log files. The following data is logged every 500ms for each of the six Robots on the playing field during each match:

- Timestamp (local time)
- Match Number
- Team Number
- Match Time
- Alliance
- Mode (Auto/Teleop)
- DS in FMS Mode (yes/no)
- Robot Mode (enable/disable)
- Estop state (on/off)
- Robot Link (yes/no)
- Bandwidth consumption over the wireless link
- Strength of the signal transmitted by the Dlink radio

- Signal-to-Noise Ratio of the wireless link
- Average packet trip time between DS and Robot
- Number of missed packets between DS and Robot
- Total number of packets sent by DS to Robot
- Robot Battery Voltage

Can one team's DS control another team's Robot?

The FRC competition field is configured such that each team has its own virtual local area network (VLAN) within which all data is passed. The characteristics of a VLAN ensure that the command packets from one team's DS do not cause a response on another team's robot. These VLANs exist on both the wired and wireless side of the playing field's network; this is why for example, it's necessary for a team in Blue Player Station #1 to connect their DS into the corresponding cable for that station. On the wireless side of the network, the VLANs are configured in the Field Access Point by broadcasting an individual network name (SSID) for each of the six teams on the field, each with its own encryption passkey. These VLANs are configured prior to the start of each match so that only the six teams assigned in FMS may operate on the field. This is also the reason each team must configure their Robot's radio at each event they attend.

What happens when you plug your DS into the competition field?

Once the process of setting up all the VLANs on the competition field is complete, through a process FRC calls "Match Prestart", the FMS is ready to connect with the six DS's. When a team plugs their DS into the Ethernet cable for their assigned Player Station, it's first assigned an IP address by the playing field, then starts sending messages to FMS. When the DS receives reply messages from FMS it switches over into FMS Mode. It's at this point "FMS Connected" is displayed on the DS "Operation" tab. If the team has successfully connected to the field, but is in the wrong Player Station, a message is displayed on the DS indicating to which station the team needs to move.

When in FMS Mode, the DS continues to serve as the master controller for the Robot, but state (enable/disable/estop) and mode (auto/teleop) are dictated by the FMS. The FMS tells the DS what to do, and the DS then tells the Robot.

Is Practice Mode on the DS different from FMS mode?

Yes, the two modes do have some differences, but the majority of the functions are identical. Both operating modes step through the same states, the order is:

1. Disable - state prior to match start
2. Autonomous Enable - Autonomous period

3. Disable - end of Autonomous period
4. Disable - end of Autonomous period, prior to start of Teleop period
5. Teleop Enable - Teleop period
6. Disable - end of Teleop/match end

Joysticks are handled a bit differently between the two modes. In Practice Mode unplugging a joystick will result in the Robot being switched to Disabled. This is designed to be a safety feature, as the Robot may be running in a variety of environments that might not be equipped with barriers to safely contain the Robot.

Unplugging a joystick in FMS Mode will not result in the Robot being disabled. If a joystick becomes disconnected, simply plugging it back in will not result in it returning to normal operation. The user must press F1 on the DS to manually rescan the USB interface to re-detect the joystick.

Finally, the network port used by the DS to send command packets to the Robot is different in Practice Mode than the one used when in FMS Mode.

Why do I need to press F1 when a joystick is disconnected in FMS Mode?

While in Disable, the DS software periodically polls the USB interface for the presence of devices and adds/removes them from the list of joysticks on the DS "Setup" tab automatically. This polling is only done in Disable as it is computationally expensive and could compromise control of an enabled robot.

In FMS Mode, the DS's Enable/Disable state is dictated by FMS. When a joystick is disconnected during a match the robot is not disabled because FMS is continuously telling the DS to be enabled, and when the DS state is Enable, it does not poll the USB interface for device changes. Pressing F1 on the DS manually rescans the USB interface to redetect the change in joystick devices. If a joystick has disconnected from the DS or is otherwise unresponsive the recommended procedure is:

1. Unplug the joystick device's USB connector from the DS
2. Plug in the joystick device
3. Press F1 to rescan the USB devices

Is there anything different about the practice field vs. the competition field?

Yes. The practice field uses a different field access point, it does not employ the port filtering, bandwidth limits, or Quality-of-Service priorities used on the competition field, nor does it employ VLANs.

Are there any bandwidth limits on the playing field?

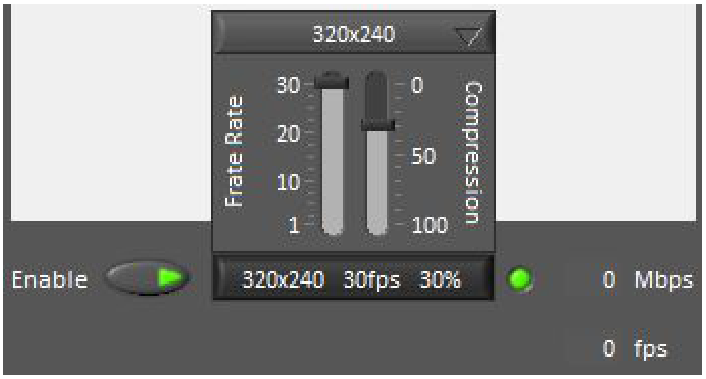
Yes, each team VLAN is limited to 7 megabits/second (Mbps). The bandwidth consumed by control packets sent from the DS to the robot, and the status packets sent from the robot to the DS are each ~40kilobits/second (kbps), or approximately 80kbps total. This leaves approximately 6.9Mbps remaining for camera traffic, NetworkTables variables, and/or any other data traffic a team chooses to employ on their VLAN. In most cases, the camera data traffic will consume the majority of the available bandwidth between the DS and the robot.

The chart below shows the bandwidth consumption for some typical camera resolutions and frame rates. This data was captured using an Axis 1011, but the data is also applicable to the Axis 206.

Bandwidth consumption for different camera resolutions

			Peak Data Rate - Mbps	
Stream Type	**Image Size**	**Frame Rate**	**Compression = 0**	**Compression = 30**
MJPEG	640x480	30	34	14
MJPEG	640x480	24	25.5	11.1
MJPEG	640x480	15	15.5	6.1
MJPEG	640x480	10	11	4.3
MJPEG	320x240	30	8	3.7
MJPEG	320x240	24	7.5	2.9
MJPEG	320x240	15	4.2	2.1
MJPEG	320x240	10	2.8	1.2
MJPEG	160x120	30	2.9	1
MJPEG	160x120	24	2	0.9
MJPEG	160x120	15	1.1	0.6
MJPEG	160x120	10	0.8	0.4

LabVIEW Dashboard camera stream controls



The Dashboard can be used to monitor the bandwidth consumed by the camera stream from the robot. The default LabVIEW Dashboard includes controls and indicators to assist the user in selecting the preferred camera stream and display the resulting bandwidth consumption.

Axis Camera bandwidth consumption overlay

The default SmartDashboard does not include integrated controls like those provided in the LabVIEW Dashboard, but provisions are available to display the bandwidth consumption of the camera stream. Including the “#b” option in the *Overlay settings/Include Text* field in the Axis camera to will overlay a bandwidth measurement (in kbps) on the camera stream. This option also works for the LabVIEW Dashboard.

What are the recommended settings for the Axis Camera?

- Image Resolution: 320x240
- Compression: 30

- Overlay settings/Include Text: #b
    - This setting will overlay a bandwidth measurement (in kbps) onto the camera stream as shown above.

What is Quality of Service (QoS)? Are there any QoS priorities on the competition field?

Typically, networks operate on a best-effort delivery basis, which means that all packet traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

Configuring QoS on the field network allows for the selection of specific network traffic, placing a priority level on it, and using congestion-management and congestion-avoidance techniques to provide preferential treatment. This makes network performance more predictable and bandwidth utilization more effective.

The FMS does have a QoS policy. The policy increases the priority of all control packets between to the DS and Robot to the same level as voice data, while keeping the priority of video data at the best-effort (or lowest priority) level. The “voice” level is the highest priority level packets can be set to without impacting the core functionality of the field network.

Prioritizing control packets at the same level used for voice data was chosen because the desired performance is similar in both applications. In order for Voice-Over-IP (VoIP) to be a realistic replacement for standard public switched telephone network (PSTN) telephony services, customers need to receive the same quality of voice transmission they receive with basic telephone services,meaning consistently high-quality voice transmissions. Like other real-time applications (e.g. controlling an FRC Robot), VoIP is extremely bandwidth- and delay-sensitive. For VoIP transmissions to be intelligible to the receiver, voice packets should not be dropped, excessively delayed, or suffer varying delay.

Which network ports are open on the competition field?

The ports that the teams are able to access on the competition field are as follows:

- UDP/TCP 1180 - 1190: This port range is typically used for camera data from the Robot to the DS when the camera is connected to the roboRIO via USB. This port is bidirectional on the field.
- TCP 1735: SmartDashboard, bidirectional
- UDP 1130: Dashboard-to-Robot control data, directional
- UDP 1140: Robot-to-Dashboard status data, directional
- HTTP 80: Camera connected via switch on the robot, bidirectional

- HTTP 443: Camera connected via switch on the robot, bidirectional
- UDP/TCP 554: Real-Time Streaming Protocol for h.264 camera streaming, bi-directional
- UDP/TCP 5800-5810: Team Use, bi-directional

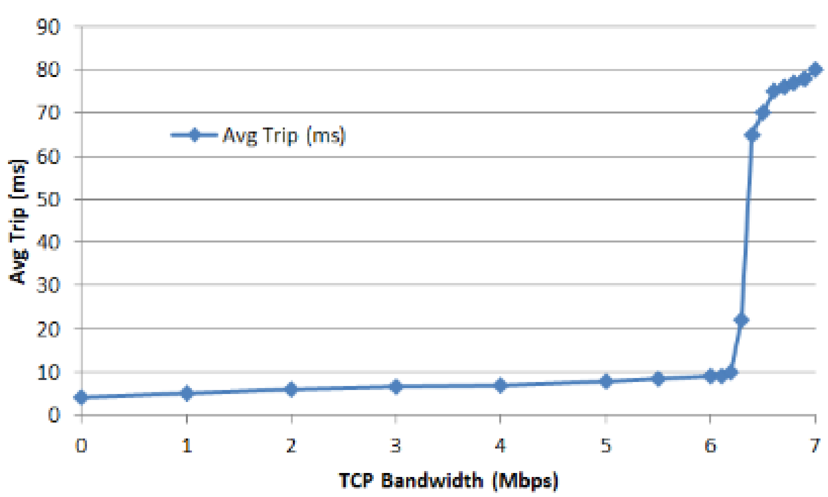
All these ports are open on the competition field, so a team can use them as they wish if they choose not to employ them as outlined above (i.e. TCP 1180 can be used to pass data back and forth between the Robot and the DS if the team chooses not to use the camera on USB).

What is the significance of Trip time?

Trip time is the roundtrip time required for a control packet to be delivered from the DS to the Robot and a corresponding status packet to be sent from the Robot to the DS. It's similar to a network ping, but it's built into the communications protocol.

The DS computes the average trip time of the last 10 packets, logs it, as well as forwards the most current average value in each status packet sent to the FMS, which occurs every 100ms. Typical trip times are under 10ms, but can increase due to network congestion as a result of high bandwidth usage, addition processing in the DS, or inefficient user code on the Robot. Trip times are monitored during each match by the FTA.

How is Trip time impacted by Bandwidth usage?



The figure above shows how Average Trip time of the DS-Robot packets increases as the total bandwidth on the VLAN is consumed. *NOTE: The data shown was captured prior to the 2014 season, but still applies in 2015. The only difference is that in 2015 the increase in Average Trip time occurs around 6.5Mbps due to the increased efficiency in control and status packet size between the DS and roboRIO.*

Troubleshooting

A significant increase in Average Trip Time occurs just above 6Mbps of data because there is already ~900Kbits/sec being consumed by the DS-Robot packets alone. Network congestion increases as you approach the 7Mbps limit of the VLAN and as a result the Average Trip Time increases. The QoS policy described above works to prioritize control packets specifically to allow for some reasonable level of Robot control to be available under such conditions, but performance cannot be guaranteed.

If you are experiencing high trip times, the first place to look should be the settings of your camera stream from the Robot to the DS.

- For teams using the LabVIEW Dashboard: Increasing the compression value, or reducing the image size or frame rate should result in the average trip time value decreasing.

For teams using the SmartDashboard: Check the settings in the Axis camera. The SmartDashboard implementation requests a MJPEG stream using the default camera settings. If they have not been manually configured, they are 640x480 @30fps with a compression value of 30. Under these conditions the camera stream alone requires ~14Mbps of bandwidth, which is twice (2X) the available limit on the competition field. Change the camera settings to match the recommended values detailed above to determine if performance improves.