

Robot Troubleshooting Guide

Revision History	
Revision	Description
25-26.1	Initial 2025-26 Season Release

Control System Introduction	3
Point-to-Point Control.....	3
Focus of this Document.....	4
Wi-Fi Technology	5
Wi-Fi Direct Group Owner.....	5
Wireless Access Point.....	5
Programming Laptop	5
Configuration Activity.....	6
Troubleshooting Wi-Fi Direct Connections	11
Monitoring and Troubleshooting the Wireless Environment	13
The Wireless Spectrum	13
Monitoring the Wireless Spectrum	14
Wi-Fi Analyzer	15
Mac OS Airport Utility	16
NetScout (formerly Fluke) AirCheck™ Wi-Fi Tester	16
MetaGeek inSSIDer.....	17
Wireshark	18
Troubleshooting the Wireless Environment at an Event	19
Ping Times	19
Is the Wi-Fi Channel Too Busy?	21
Potential Sources of Wi-Fi Interference	22
Potential Sources of Non-Wi-Fi Interference	22
Are There Too Many Robots Operating on the Same Channel?	22
Is there a Wi-Fi Suppressor Operating in the Vicinity?	22
Are the Wireless Radio Signals Being Blocked by Metal?.....	23
Is There Malicious Activity Occurring?.....	24
Determining if Wi-Fi Interference Warrants a Match Replay.....	24
Accommodating a Large Number of Robots at an Event	26
Wi-Fi Event Planning Guide	26
Distributing Robots Across Multiple Channels	26
Wi-Fi Channel Overlap	26
Factors to Consider when Selecting Wi-Fi Channels.....	27
UnPairing Then Re-Pairing the Driver Station to the Robot Controller.	28
Changing the Channel Using an Approved Motorola Smartphone.....	31
Mitigating Disruptions Due to Electrostatic Shocks.....	33
Troubleshooting Common Issues	33
FIRST Tech Challenge Driver Station	33
Robot Controller.....	35
Neglecting to Insert waitForStart() Statement	36
Uninterruptible Threads	37

REV Robotics Control and Expansion Hubs.....	38
Useful Tips and Tricks.....	40
Use a Pair of Android Devices to Monitor Wi-Fi Channel	40
Use the Log Files to Help Troubleshoot Problems.....	40
Wireshark	41
Creating a Capture Filter for DEAUTH Packets.....	41
Viewing WLAN Traffic Statistics	44
Getting Additional Help	46
Tech Tips on Using Log Files.....	46
Introduction.....	46
Verify the Date and Time.....	46
The FIRST Tech Challenge Log Files.....	46
Viewing the FIRST Tech Challenge Robot Controller Log File	47
Using the Android Debug Bridge for Troubleshooting	52
Using Android Studio to View Log Messages.....	56

Control System Introduction

The *FIRST* Tech Challenge uses an Android-based Control System for its robot competition. This document provides tips and recommended procedures for avoiding potential problems with the Android-based Control System. It also provides information to help troubleshoot and resolve common problems with the system.

Point-to-Point Control

The Control System that is used for the *FIRST* Tech Challenge competitions uses a point-to-point communication model. Each team has an Android device that acts as a *Driver Station* (or DS). The Driver Station establishes a secure and unique wireless connection with a second Android device that is mounted on the robot, and which is known as the Robot Controller (or RC).



Figure 1 - Each DS-RC pair has its own unique wireless connection

With the Android-based Control System, it is the teams that are responsible for bringing, maintaining, and troubleshooting the wireless Control System for their robot. At an event, each team will have a Driver Station and a Robot Controller. The two components will be paired with a secure and unique wireless connection.



Figure 2 - Each Team will have its own DS, RC and wireless network connection

Early versions of the FTC control system used an Android smartphone as the primary controller. This Android device was connected to an input/output (I/O) module through a USB connection. Most teams

use the Control Hub as the primary controller. The Control Hub has a built-in Android device and is connected to the I/O module through an internal serial connection.



Figure 3 - A Control Hub is a Robot Controller with an internal Android device

Starting with the 2021-2022 season, the REV Driver Hub is approved for use as the Driver Station instead of an Android smartphone. The Driver Hub has built-in 2.4 and 5GHz Wi-Fi enabling the ability for wireless connection to a robot with additional security through 802.11w. The REV Driver Hub can configure a REV Control Hub (directly) or Expansion Hub (via Control Hub or RC phone).



Figure 4 - Expansion Hub or Control Hub configured with REV Driver Hub (shown)

Focus of this Document

Although teams will be responsible for providing and maintaining their Robot Control System, they may occasionally encounter problems which require assistance from a *FIRST* Technical Advisor (FTA), Control System Advisor (CSA), and/or a Wi-Fi Technical Advisor (WTA). Also, there are steps that an Event Host, FTA, CSA, and/or WTA can take before and during an event to help mitigate wireless issues with the Control System.

This document provides information on steps that can be performed before and during a *FIRST* Tech Challenge competition to help ensure that the wireless systems run smoothly. This document also provides tips on how to diagnose/troubleshoot commonly encountered problems.

This document is not intended to teach users how to operate the *FIRST* Tech Challenge Control System. This document assumes that the reader has a basic understanding on how to configure and use the components of the system. For information on how to use the Android Control System, please visit the [FIRST Tech Challenge Programming Resources](#) web page and view the training documents listed. Also, REV Robotics, the manufacturer of the [Driver Hub](#), [Control Hub](#) and [Expansion Hub](#), has excellent [online documentation](#) including detailed instructions on using and troubleshooting the [Control Hub](#), [Expansion Hub](#), and [Driver Hub](#).

Wi-Fi Technology

The Driver Station and Robot Controller are Android devices that run special *FIRST* Tech Challenge apps to create a unique and secure wireless connection between the two devices. For this connection, the REV Control Hub uses Wireless Access Point (WAP) technology, while the standalone phone-based Robot Controller uses Wi-Fi Direct (P2P) technology. There are some minor, subtle differences between how these two technologies connect the devices together wirelessly. Note that the FTC Driver Station app is able to connect to both types of Robot Controllers.

Wi-Fi Direct Group Owner

For a Wi-Fi Direct P2P connection, one of the peer-to-peer devices acts like a Wi-Fi access point and is referred to as the *group owner*. The group owner establishes a Wi-Fi Direct group that the other devices can connect to. The other peer-to-peer device is referred to as the *client* device. For the *FIRST* Tech Challenge application, the Robot Controller phone is the device that acts as the group owner for the P2P connection. The Driver Station device is the client device, and it connects to the Wi-Fi Direct group through the FTC Driver Station app using Android's P2P technology. A Wi-Fi Direct connection requires that a user manually accept (using the P2P group owner's touch screen) the initial connection request from a P2P client.

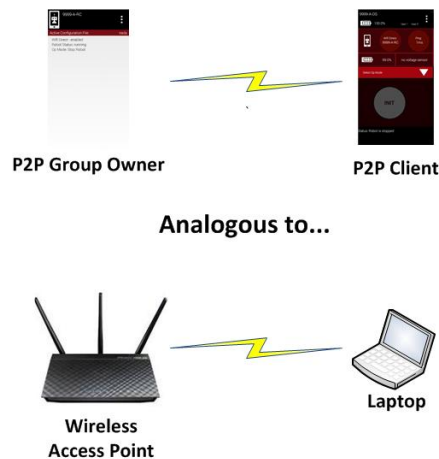


Figure 5 - The P2P group owner is analogous to a Wi-Fi access point

Wireless Access Point

A Control Hub is slightly different from a phone-based Robot Controller. A Control Hub acts as an actual wireless access point. A Driver Station device connects to the Control Hub's Wi-Fi network like it would to any other Wi-Fi network. The user only needs to provide the correct password in order to access the wireless network – no manual acceptance step on the access point is required.

Programming Laptop

During a typical *FIRST* Tech Challenge match, only a team's Driver Station is connected to the Wi-Fi Direct Group or the wireless access point (WAP) that is established by the team's Robot Controller. Away from the competition field, however, a team might have additional devices connected to this Wi-Fi Direct Group. For example, when a team edits an OpMode using the [FTC Blocks Development Tool](#) or the [FTC OnBot Java Development Tool](#), their developer's laptop will also be connected to the Robot Controller's wireless network.



Figure 6 - A team might also have a developer's laptop connected when they are away from the competition field

Note that the wireless connection between the developer's laptop and the Robot Controller does not violate the prohibition in the Competition Manual on teams setting up their own wireless network. For this case, the developer's laptop is connected to the existing Wi-Fi Direct Group or wireless access point that is also used by the Driver Station to communicate with the Robot Controller.

Configuration Activity

The Android operating system has a built-in configuration screen or activity that can be used to view and configure the Wi-Fi Direct settings.

Note: that for the *FIRST* Tech Challenge apps, you typically do NOT want to use the Android Wi-Fi Direct menu to pair your devices. Instead, you should use the **Pair with Robot Controller** activity that is available from the **Settings** menu of the FTC Driver Station app to pair/unpair your devices.

It is useful to be familiar with Android's Wi-Fi Direct configuration activity. As an FTA/CSA you might need to use this screen to check on the configuration of an Android device, and to clear/erase remembered groups or do other tasks to help get a robot back in action.

Accessing the Wi-Fi Direct Configuration Activity

On your Android device, launch the **Settings** activity then click on the **Wi-Fi** item.

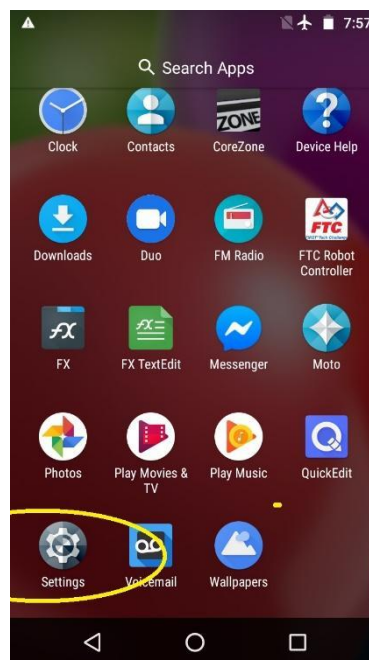


Figure 7 - - Launch the Settings menu

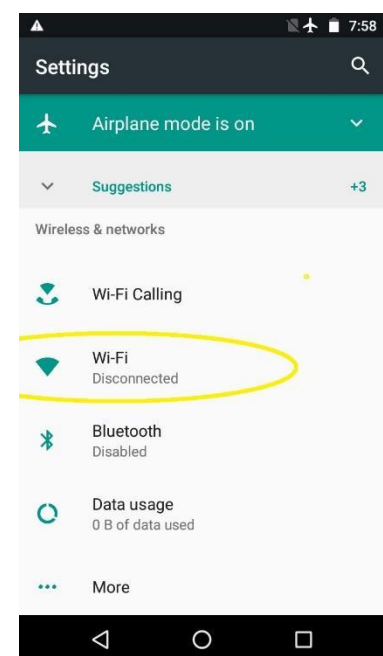


Figure 8 - Click on "Wi-Fi"

To access the Wi-Fi Direct menu, touch the three dots in the top right-hand corner of the screen to display a short pop-up menu. Select **Advanced** from the pop-up menu.

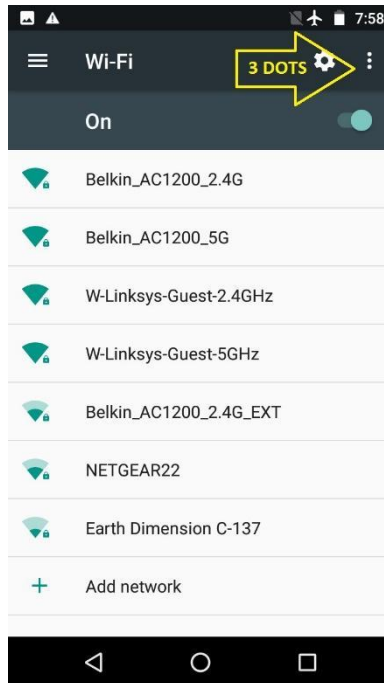


Figure 9 - Click on the 3 dots

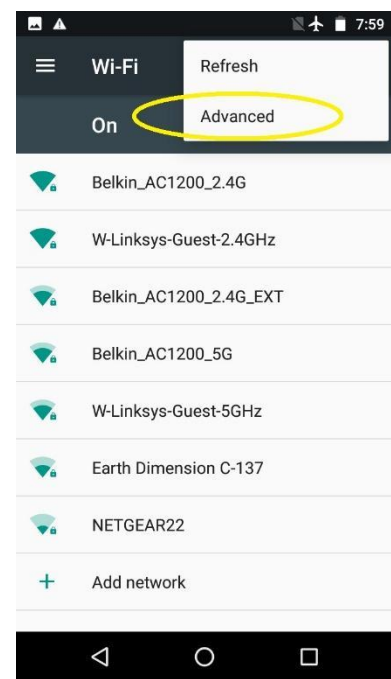


Figure 10 - Choose Advanced

In the Advanced Wi-Fi menu, select **Wi-Fi Direct**. Note that the screenshots in this document were generated using a Moto e4 phone running Android 7.1.1. The screen images and menu text might vary from device to device.

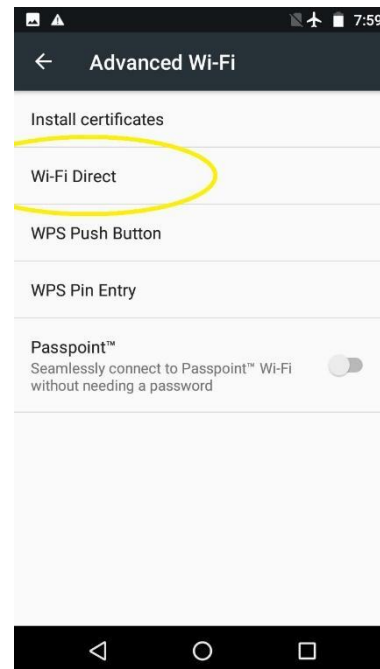


Figure 11 - Select Wi-Fi Direct

This screen shows the Wi-Fi Direct group name, if any, along with any connected device(s) and remembered group(s). When troubleshooting, it is usually best to clear all the items that this screen will allow. Sometimes the “connections” listed here have become corrupted. Clearing connections here, and

re-establishing later (from the FTC Driver Station app), is a fast, simple and reliable way to ensure good communications.

Note: that the Wi-Fi Direct group cannot be renamed at this stage. All connections must be cleared first. Also note, you should request and be granted permission from the team before you clear any Wi-Fi Direct groups.

To begin, touch one of the remembered group names, then click OK to forget that group. Repeat for any other remembered groups listed

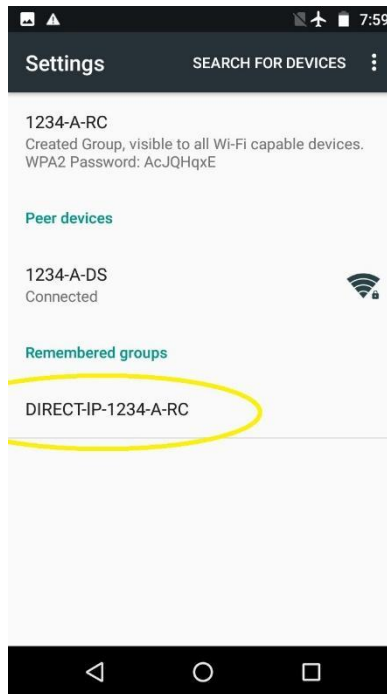


Figure 12 - Touch any group name(s)

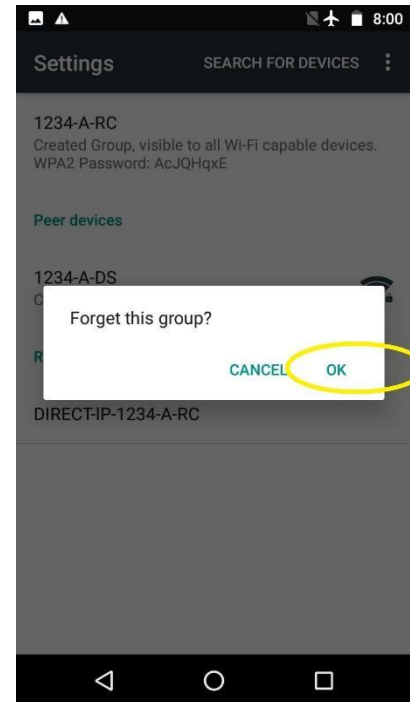


Figure 13 - Forget each group

Next, do the same thing with any peer devices and the Created Group.

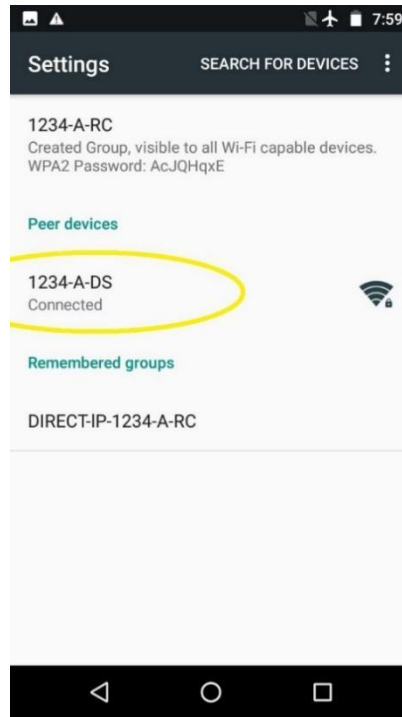


Figure 14 - Peer Devices

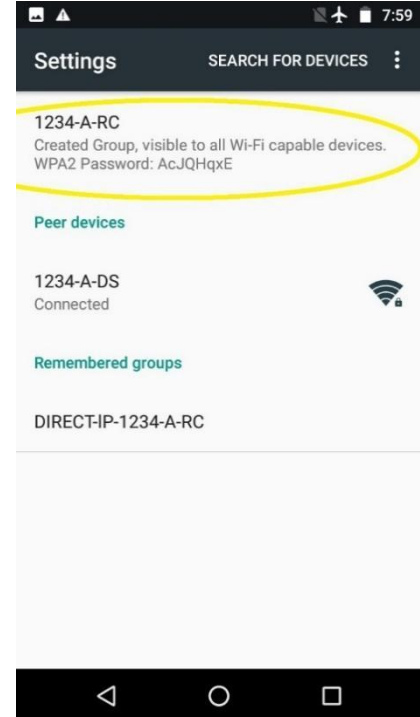


Figure 15 - Created Group

Touching either of these items brings up the following "Disconnect?" screen – select OK. If one of these items cannot be disconnected after several tries, continue with the other items. A persistent item like this will generally not cause a problem later.

When the disconnecting is complete, the screen will list only the Device Name, which was also the Wi-Fi Direct Group Name.

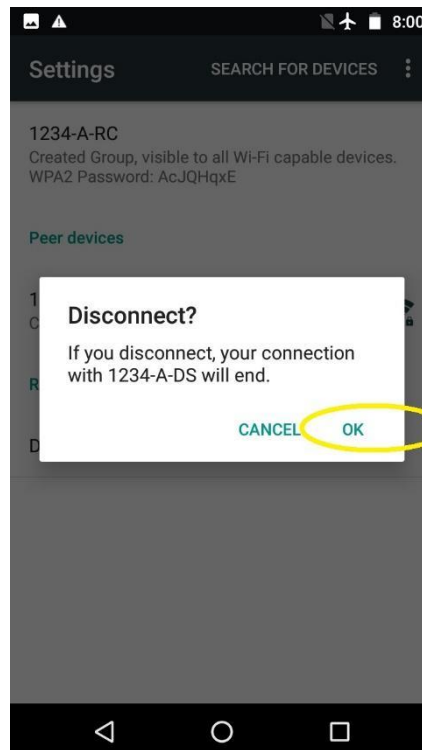


Figure 16 - Select OK to disconnect.

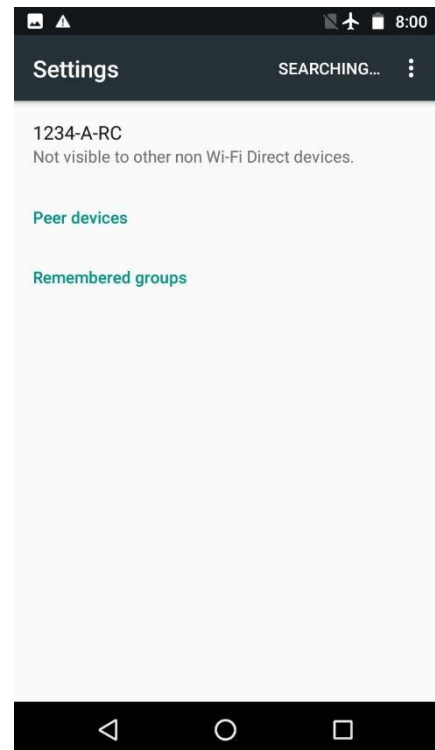


Figure 17 - All clear

In this disconnected state, the Device Name can be changed. Touch the 3 dots again at the top right corner. Now the Configure Device selection is live and can be clicked.

Device naming must follow the rules described in the Competition Manual.

At this screen, Motorola phones offer three features not present on previous FTC phone models.

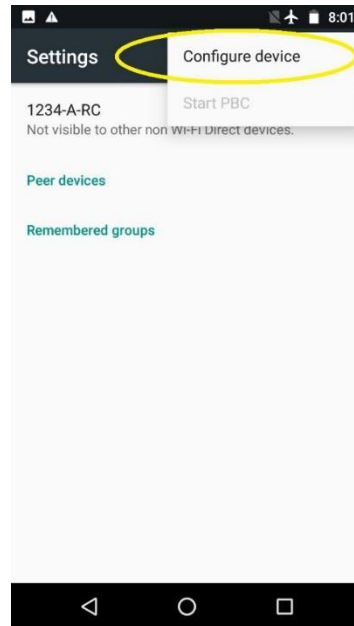


Figure 18 - Configure Device

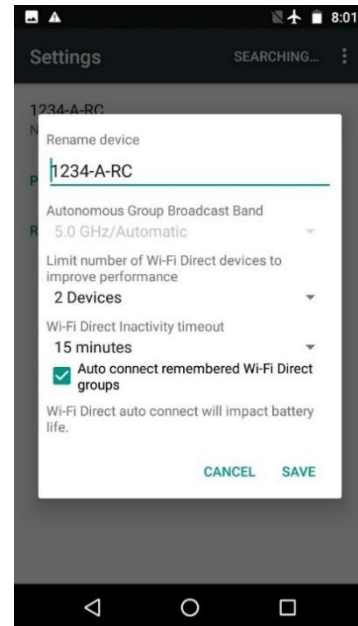


Figure 19 - Rename Device

The maximum number of connections can be specified, ranging from 2 to 8. A smaller number is safer and more efficient, while a larger number could allow (for example) multiple Blocks or OnBot Java programmers to access a single shared RC device. Users must be very careful to avoid conflicts in sharing and editing.

The other two items allow selection of a Wi-Fi Direct inactivity timeout, and whether or not to automatically connect to a remembered Wi-Fi Direct group when discovered. You can make selections here, but in general, the FTC apps will manage connections, as needed.

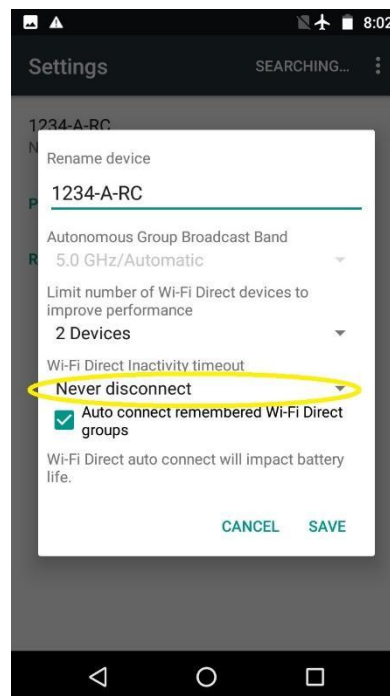


Figure 20 - Optional Settings

Click Save when done and return to the home screen. Make new connections only from the FTC Driver Station app, as described in the following section.

Troubleshooting Wi-Fi Direct Connections

Ideally, the teams should be able to use the **Pair with Robot Controller** activity of the FTC Driver Station App to pair to the target FTC Robot Controller. Once the devices have been paired through the FTC Driver Station app, they should automatically reconnect to each other when both devices are turned on *and* both devices have their respective FTC apps running.

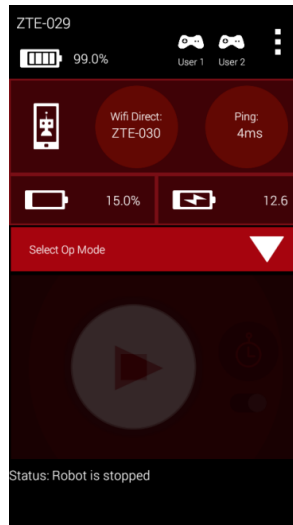


Figure 21 - When the Driver Station is connected, it displays useful status info.

When your Driver Station is able to connect to the Robot Controller successfully, it will display useful status information (see Figure 21) on its screen including the name of the device that it is connected to, the average ping times between the Driver Station and Robot Controller and voltage info for the Robot Controller smartphone (if used) and the main robot battery.

Is the Robot Controller On?

For problems connecting to the Robot Controller, check the following basic items:

- Is the Robot Controller device turned on?
- Is the Robot Controller smartphone in Airplane mode with Wi-Fi enabled?
 - Smartphone specific:
 - Is the Robot Controller device running the FTC Robot Controller app?
 - Is the FTC Robot Controller app in the foreground (and NOT minimized)?
 - The Robot Controller device must be powered on and have the FTC Robot Controller app running before the Driver Station can connect to it.

Are Both **FIRST** Tech Challenge Apps Installed?

If you are having problems pairing the Android devices, please make sure that you do not have the FTC Driver Station app and the FTC Robot Controller app installed at the same time on a single Android device. The apps have the potential to cause Wi-Fi Direct conflicts if they are both installed. Make sure neither device has both apps installed at the same time.

Do Both FTC Apps Have the Same Version Numbers?

If you have verified that the Robot Controller and the Driver Station are turned on and have their respective FTC apps running, verify that the apps have compatible version numbers. If you select the **About** menu item for each app, a new screen should appear on the Android device with version

information about the app. It is most important that the “Robot Wi-Fi Protocol Version” numbers of the FTC Robot Controller app and the FTC Driver Station app match. For example, if the Robot Controller has a “Robot Wi-Fi Protocol Version” number of v4 but the Driver Station only has a “Robot Wi-Fi Protocol Version” number of v3.5, then the two apps might be unable to connect and communicate with each other, due to the difference in the versions.

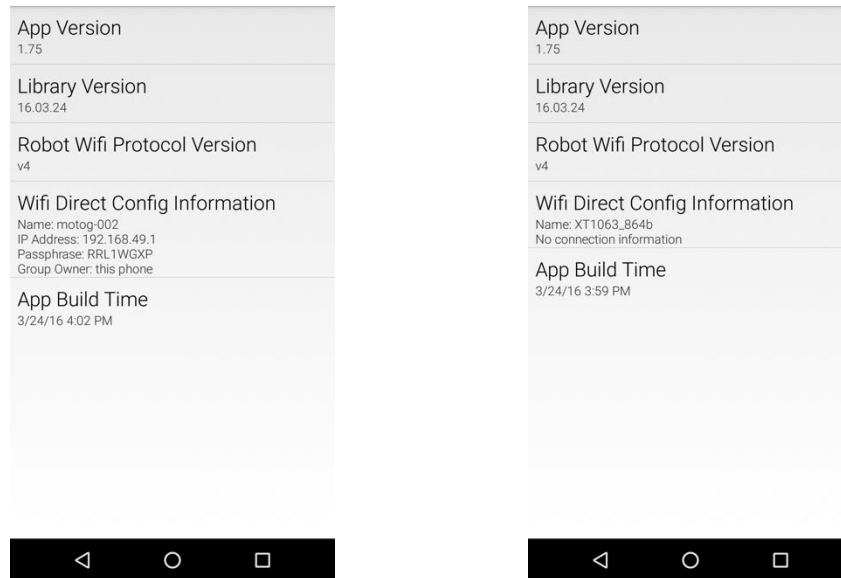


Figure 22 - The Robot Wi-Fi Protocol Versions should match for the Robot Controller (left) and the Driver Station (right)

If the “Robot Wi-Fi Protocol Version” numbers do not match, then one of the apps should be downgraded or upgraded so that the numbers will match. Often it is advisable to upgrade, however, in some instances, a team might feel more comfortable downgrading to a previous, stable version of the app. Minimum required levels of FTC app versions are specified in the [Competition Manual](#) and its updates. The [Competition Manual](#) specifies a minimum required app version level and the major and minor version numbers for the DS and RC apps must be the same.

Is Either Device Also Connected to Another Network?

For the *FIRST* Tech Challenge competitions, we recommend that the Driver Station and Robot Controller devices are not connected to any other networks other than each other. It is possible and often desirable to connect your Android to an alternate wireless network:

- Teams like to use the *wireless ADB* mechanism¹ to debug their apps.
- Teams might need to connect to a wireless network to download something to their phone from the Internet.
- Teams might have used the Android device to check their e-mail or look up something on the Internet (we do not recommend doing this).

We also recommend that the teams *forget* any other Wi-Fi or Wi-Fi Direct network, except for primary Wi-Fi Direct connection between the Driver Station and Robot Controller (this applies only to the use of smartphone as the Robot Controller).

If a team’s Driver Station is having trouble connecting to the Robot Controller, check the following,

¹ See <http://developer.android.com/tools/help/adb.html#wireless> for details on wireless use of the Android debug bridge (ADB).

- Check to see if either Android device is connected to another Wi-Fi or Wi-Fi Direct device.
- If either Android device is connected to another wireless network, disconnect the device from the other network, forget the other network, and restart the Driver Station and Robot Controller apps.

Are there Lots of Devices Trying to Pair Simultaneously?

Before an Android device can connect to another device, it will *scan* the wireless spectrum to determine what Wi-Fi Direct enabled devices are available in the vicinity. This *discovery* process can be negatively affected if there is a high concentration of Wi-Fi Direct devices in the vicinity that are also scanning the spectrum for available devices. For instance, if there are many devices in the vicinity, the target device that you are trying to connect to ("12345-A-RC" for example) might not be visible in your list of available Wi-Fi Direct devices on your Android phone.

If you are at an event and the Android devices are consistently unable to find each other, or if the devices have trouble establishing a connection, it could be due to the presence of so many other Wi-Fi Direct enabled devices. If this is the case, one option would be to remove the pair of devices that you are trying to connect away from the crowd, and pair the two devices further away so that the other devices do not interfere with the discovery and pairing process.

Another option is to turn off the Android devices in the vicinity, and then have the teams turn on and pair their devices in successive small groups of no more than four teams or eight devices at a time. A wait time of a few minutes between each small group is recommended.

Once the devices are connected, they can withstand a reasonable amount of wireless traffic and noise and still operate reliably. This means that once a team has been able to pair/connect its Android devices, the team should be able to use the devices, even if there are a relatively high number of other devices operating in the vicinity.

Monitoring and Troubleshooting the Wireless Environment

The *FIRST* Tech Challenge Control System uses Wi-Fi Direct and/or Wireless Access Point technology to connect the Driver Station device to the Robot Controller. Wi-Fi Direct networks can be managed like normal Wi-Fi networks. The techniques and tools that you might use to monitor and troubleshoot a corporate Wi-Fi network can be applied to the Wi-Fi Direct networks used by the FTC Control System. This chapter provides some basic information to help the FTA/CSA/WTB keep the wireless environment clean and operational at an event.

The Wireless Spectrum

Wi-Fi enabled devices use wireless radios to send digital information back and forth to each other. These devices operate at specific frequencies within legally allocated portions of the wireless spectrum. All FTC-approved Android devices can operate at the 2.4GHz and 5GHz frequencies.

2.4GHz Portion of the Spectrum

In the U.S., there is a band of 11 channels (1 through 11 in Figure 23 shown below) in the 2.4GHz region that a Wi-Fi enabled device can use to communicate. Other regions outside of the U.S. often allow Wi-Fi devices to operate on a few additional channels.

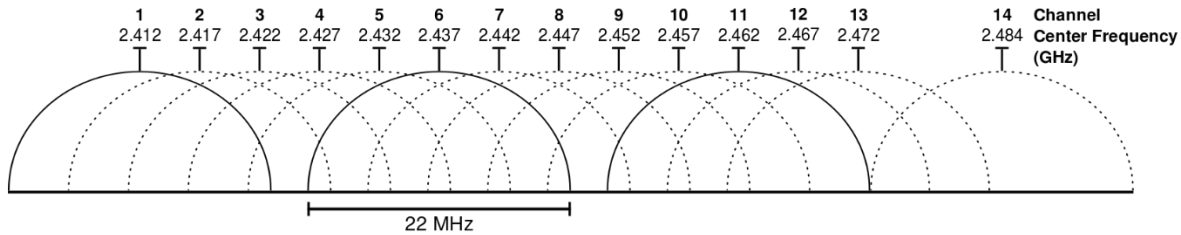


Figure 23 - In the U.S., there are 11 Wi-Fi channels in the 2.4GHz band. Other countries allow a few additional channels.²

For the 2.4GHz portion of the spectrum, although there are 11 (or more) operating channels that are available to use, you can see in Figure 23 that the adjacent channels overlap each other. This means that if you have two devices operating on wireless channels that are not separated by 5 channel “widths”, then the radios from each device will interfere with each other.

Ideally, if you want to avoid interference between two channels, you should make sure there is at least a 5-channel width separation in-between the two channels. For example, channels 1 and 6 have enough spacing in between so they will not interfere with each other. However, channels 1 and 5 overlap slightly and there will be some interference between the two overlapping channels.

In practice, a little bit of overlap between the channels might be OK. If one or more channels in the spectrum is very noisy and unusable at an event, then you might have to consider moving your devices to alternate, possibly overlapping channels.

Each wireless channel can only support a limited number of devices operating on the same channel. As the number of devices that are operating on a channel increases, the amount of noise and interference on that channel increases.

The FTC Driver Station and FTC Robot Controller apps can tolerate a fair amount of noise and interference. This means that it is usually possible to support a relatively large number (25 to 35 or even more) Driver Station-Robot Controller pairs on a single 2.4 GHz Wi-Fi channel. However, if there are other sources of traffic on a Wi-Fi channel (including non-Wi-Fi enabled devices) then the wireless connectivity of the Driver Station-Robot Controller pairs can suffer.

5GHz Portion of the Spectrum

FTC-approved smartphones and the REV Robotics Control Hub and Driver Hub support both the 2.4 and 5 GHz bands. The 5GHz band channels have the advantage that they do not overlap each other. Also, 5GHz channels offer greater bandwidth and have more limited range (which can be useful in a crowded competition venue with lots of robots) than 2.4GHz channels. **Note:** that the FIRST Robotics Competition (FRC) robots now exclusively compete on the 6GHz portion of the spectrum. This makes it easier to manage and troubleshoot the spectrum at FRC/FTC Hybrid events.

Monitoring the Wireless Spectrum

There are some tools that are available to an FTA, CSA, or WTA that can be helpful in monitoring activity on the wireless spectrum.

² This diagram was copied from Wikipedia ([https://en.wikipedia.org/wiki/List_of_WLAN_channels#/media/File:2.4_GHz_Wi-Fi_channels_\(802.11b.g_WLAN\).svg](https://en.wikipedia.org/wiki/List_of_WLAN_channels#/media/File:2.4_GHz_Wi-Fi_channels_(802.11b.g_WLAN).svg)) on 9/20/15.

Wi-Fi Analyzer

Wi-Fi Analyzer is a free app that is available on the Google Play store that you can install onto your Android device. <https://play.google.com/store/apps/details?id=com.farproc.Wi-Fi.analyzer&hl=en>

Wi-Fi Analyzer lets you see what wireless networks are operating in your venue. The app has a very useful graphical display that shows the available networks and overlays the networks onto a graph that shows the operating channels for each network. The app also displays the relative strength of each network.

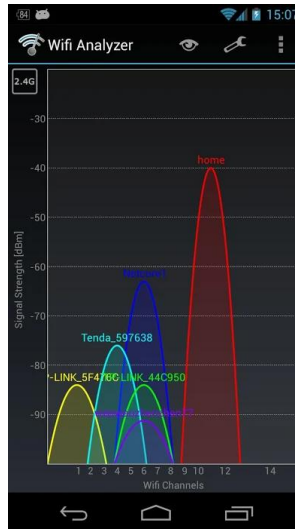


Figure 24 - Wi-Fi Analyzer screen shot.³

An FTA, CSA, WTA, or Tech Director can use the Wi-Fi Analyzer tool to see which networks are present at a venue. If there are some unauthorized wireless networks on a channel, the FTA, CSA, or WTA might be able to identify them using Wi-Fi Analyzer. Wi-Fi Analyzer can also be used to determine which wireless channel a team's Driver Station-Robot Controller pair is using.

Note: that if you run the Wi-Fi Analyzer app on an Android device that has a dual-band (2.4GHz and 5GHz) radio, then you can monitor channels on the 2.4GHz and 5GHz bands.

An FTA, CSA, WTA, or Tech Director can also use the app to help plan which channels the teams should use for their robot communication. In general, the robots should operate on the Wi-Fi channel with the least number of other wireless networks on that channel (assuming there are not any other sources of interference on that channel).

While the Wi-Fi Analyzer app is a helpful tool, it does have some limitations:

- Wi-Fi Analyzer will not display any activity from non-Wi-Fi signals operating on the same frequency. For example, if someone is operating a wireless microphone system at the venue, the microphone might be transmitting on the same frequency (2.4GHz) as the FIRST Tech Challenge devices. Wi-Fi Analyzer does not have the ability to detect and display non-Wi-Fi activity so it would not be able to tell if there was interference from something like a wireless microphone.
- Wi-Fi Analyzer only lists wireless networks and the relative signal strength of each network. It does not provide any information on how much activity is occurring on a network. An Event Host can use the app to see what Wi-Fi networks are operating on a channel, but the Host

³ Image taken from the Google Play listing for Wi-Fi Analyzer.

cannot determine if any of the networks are very busy and use up a lot of the available capacity on a channel.

- Wi-Fi Analyzer will not display any hidden Wi-Fi networks that might be operating on a channel. Typically, when someone sets up a wireless network, they have the option of hiding the network. The Wi-Fi Analyzer app will not list a hidden Wi-Fi network.

Mac OS Airport Utility

If you have access to a Mac OS computer, you can use the *airport* utility function to scan for locally available wireless networks. To use the airport utility, you must have the airport executable file included in your Mac OS shell's search path. You can also place a symbolic link to the airport utility in the `/usr/local/bin` folder of your Mac's file system. From a Mac OS command terminal, you can use the following command to create the symbolic link (note you will need to use super user status and provide your account's password to create the link): `sudo ln -s`

`/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport /usr/local/bin/airport`

From a Mac OS command terminal, if you type in the following command

```
airport --scan
```

The computer will conduct a scan of the wireless environment and list available local networks that it detected.

```
Toms-MBP:~ tom$ airport --scan
      SSID BSSID           RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
belkin.f5c 08:86:3b:20:4f:5c -89  11    Y  TW WPA(PSK/AES/AES) WPA2(PSK/AES/AES)
xfinitywifi e6:89:2c:f3:d9:c0 -90  11    Y  US NONE
CA52349 c8:d7:19:f0:73:94 -84   6    Y  -- WPA(PSK/AES,TKIP/TKIP) WPA2(PSK/AES,TKIP/TKIP)
Caroline's Wi-Fi Network 90:72:40:18:25:96 -84   6    Y  US WPA2(PSK/AES/AES)
CE_NET 78:24:af:7d:1c:c8 -46   6    Y  -- WPA2(PSK/AES/AES)
CE_NET 78:24:af:7d:1c:cc -59  149   Y  -- WPA2(PSK/AES/AES)

Toms-MBP:~ tom$
```

Figure 25 - The command “airport –scan” will list available visible Wi-Fi networks.

The `airport --scan` utility has similar limitations to the Wi-Fi Analyzer app described in this document. Also, this command line argument does not run continuously. A user must repeatedly issue the command (or write a script to do so) to get a continuous listing of available Wi-Fi networks.

NetScout (formerly Fluke) AirCheck™ Wi-Fi Tester

A company called netAlly (formerly NetScout and Fluke) makes an expensive, but powerful wireless network monitoring tool. The netAlly *Aircheck™ Wi-Fi Tester* is a handheld device that can be used to monitor the wireless spectrum at an event. Details regarding the Aircheck G3 Pro device can be found on the netAlly website:

<https://www.netally.com/products/aircheck/>



Figure 26 - netAlly AirCheck G3 Pro and the older Fluke Aircheck™ Wi-Fi Tester.

The netAlly AirCheck G3 Pro and the older Fluke AirCheck devices are similar to, but even more powerful than the previous tools that we have listed. The Aircheck can display information about any wireless network in the vicinity. Unlike the Wi-Fi Analyzer app and the Apple airport utility, these devices also provide information about hidden wireless networks.

Unlike the Wi-Fi Analyzer app, the AirCheck monitor can also tell the user how much wireless activity (both Wi-Fi and non-Wi-Fi) is occurring on a specific wireless channel. The AirCheck monitor can estimate how much of the channel's capacity is being consumed. This can help an FTA/CSA/WTAs determine if a wireless channel is “clean” or “noisy.”

It is important to note that the AirCheck monitor measures both Wi-Fi and non-Wi-Fi activity on a wireless channel. This feature can be useful for determining if other non-Wi-Fi devices (such as a wireless audio-visual system or a Bluetooth device) are affecting the Wi-Fi connections on a specific channel.

The Aircheck can be equipped with an external directional antenna. The external antenna can be helpful in locating the source of a wireless signal. An FTA/CSA/WTAs can use the antenna to monitor the strength of a wireless signal. The signal strength will increase as the antenna is pointed at the source of the signal.



Figure 27 - The Aircheck™ can be equipped with an external antenna.

MetaGeek inSSIDer

A company called MetaGeek makes software and hardware devices for Windows computers and smartphones that provide similar capabilities to the Fluke monitor at a slightly lower price. MetaGeek's *inSSIDer* software with their *Wi-Spy Mini* or *DBx* hardware combined with a Windows computer can be used to monitor wireless channels and see Wi-Fi and non-Wi-Fi traffic on the wireless spectrum. The MetaGeek *Wi-Spy Air* hardware connected to a smartphone provides similar Wi-Fi scanning capability in a more portable configuration.

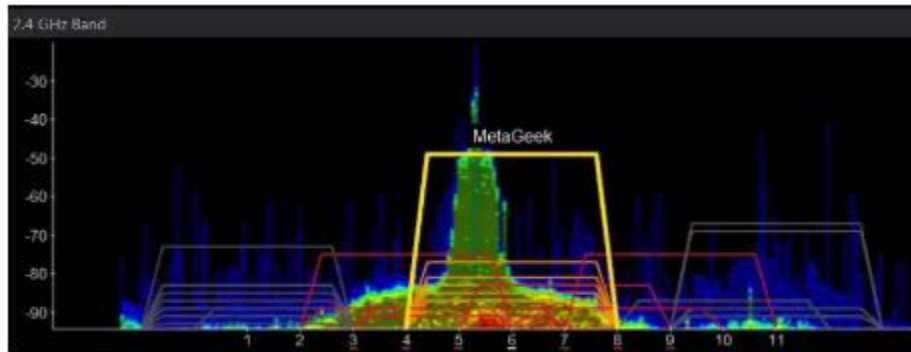


Figure 28 - MetaGeek's inSSIDer software with the Wi-Spy Mini or DBx hardware shows Wi-Fi & non-Wi-Fi activity.

Details regarding Wi-Spy Air, inSSIDer and the Wi-Spy Mini and Wi-Spy DBx hardware can be found on the MetaGeek website: <http://metageek.com>

The MetaGeek software and Wi-Spy Mini or DBx hardware require a laptop running Windows 10 to operate.

Wireshark

There is a free software application called Wireshark which can be used to help monitor and diagnose wireless issues. Wireshark is a powerful tool that requires specialized knowledge to operate.

Details on how to install and operate Wireshark can be found at the following website:
<https://www.wireshark.org/>

Explaining how to use Wireshark is beyond the scope of this troubleshooting guide. This part of the document provides instructions on how to use Wireshark to look for some specific problems with your wireless network. However, for detailed instructions on how to use Wireshark, consult the Wireshark website.

Wireshark is a tool that lets you capture and analyze the wireless *packets* that are being sent through the airwaves. To be able to use Wireshark to capture these wireless packets, you need to have a specially equipped computer. To be able to capture the wireless packets in your venue, your computer must support *monitor mode* operation for your wireless adapter.

Normally, when your computer's wireless card receives a wireless packet, it looks at the destination address of the packet. If the packet is not addressed to the computer's wireless adapter, then it will ignore the packet. If your computer is set up so that it can operate in monitor mode, then the wireless adapter will capture *all* the wireless packets that it receives, regardless of the destination address of the packets.

Wireshark can be installed on Mac, Linux, and Windows computers. Not every computer, however, supports monitor mode for their wireless adapters. Most Windows PCs do NOT allow for monitor mode. You can purchase an external wireless adapter (that connects through the USB port of the computer) to use Wireshark with a Windows PC (consult the Wireshark website for details).

Apple Mac computers support monitor mode operation of their wireless adapters. If you have a Mac computer you can install and run Wireshark in monitor mode. You do not need any special hardware to support monitor mode on a Mac OS machine.

For the Linux operating system, some, but not all, wireless adapters have drivers that support monitor mode operation. If you are a Linux user, you might need to consult the Wireshark documentation, as well as the Linux documentation to determine if your setup supports monitor mode.

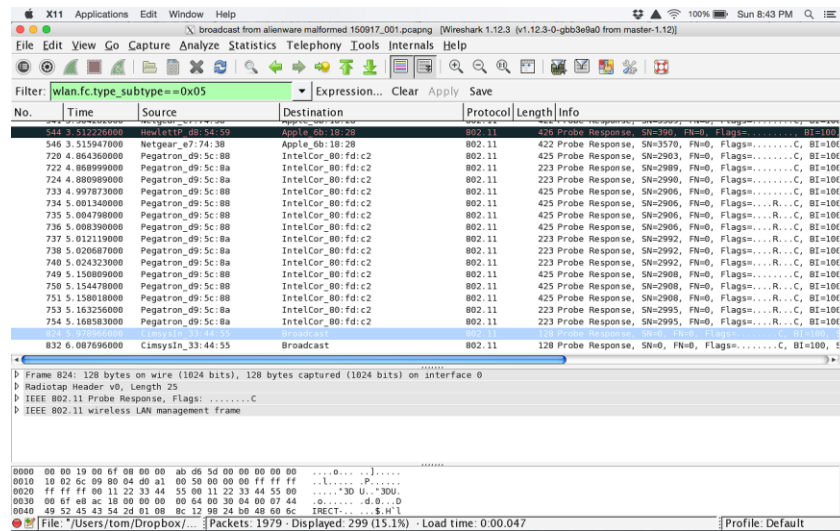


Figure 29 - Wireshark is a powerful tool, but it requires special knowledge and a Wi-Fi adapter that supports monitor mode.

If you do have a computer that supports monitor mode operation, then you can use Wireshark to capture and analyze samples of wireless packets at your venue:

1. With Wireshark you can examine the *retry rate* for a wireless network. Every Wi-Fi packet that has a specific destination address is supposed to be acknowledged by its target recipient. If the packet is not acknowledged, then the sender will attempt to retransmit the packet to the recipient. The retry rate is a ratio of retry packets to the total number of packets. The retry rate for a Wi-Fi network is an indicator of connection quality. As the retry rate increases, the Wi-Fi connection quality tends to decrease. Under ideal conditions (only one pair of wireless devices, no external interference, devices are stationary and relatively close to each other, the devices are equipped with quality radios and antennas) the retry rate should be around or under 5%. However, in practical conditions, the observed retry rates typically will be much higher (10% to 40%). In general, a lower measured retry rate corresponds to a clean wireless environment. If your observed retry rates are constantly hovering around or above 35% then your wireless channel might have a lot of interference and/or excessive traffic.
2. You can use Wireshark to examine the wireless data to look for evidence of problems such as a DEAUTHENTICATION attack or a malformed Wi-Fi Direct probe response.
3. You can use Wireshark to see what Wi-Fi devices are transmitting in or near your venue (although other tools like the Fluke Aircheck meter or the MetaGeek inSSIDer software might be better suited for this task).

Troubleshooting the Wireless Environment at an Event

If you are at an event and you suspect that there might be wireless interference that is causing problems then there are some things that you can look at to try and diagnose the problem.

Ping Times

If you are at a *FIRST* Tech Challenge event, you can use the ping time feature of the FTC Driver Station app as an indicator of network quality. When a Driver Station is connected to a Robot Controller, it will periodically send a *heartbeat* packet to the Robot Controller. The Robot Controller is supposed to respond to each ping and send an *acknowledgement* packet (aka “ACK”) back to the Driver Station.

The Driver Station constantly measures the amount of time that it takes to send a heartbeat packet to the Robot Controller and to receive an acknowledgement packet back from the Robot Controller. This amount of time is known as the *ping time*.



Figure 30 - Ping time represents the time it takes for a packet to be sent to and acknowledged by the Robot Controller.

Whenever a Driver Station is connected to a Robot Controller, the average ping time is displayed in the upper right-hand corner of the FTC Driver Station app (see Figure 30).

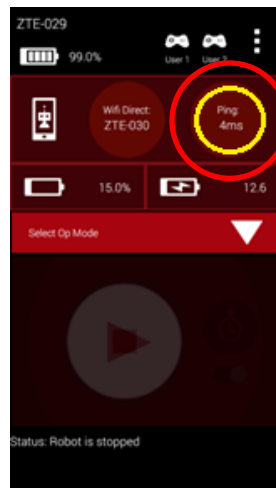


Figure 31- The average ping time is displayed in the upper right-hand corner (highlighted in yellow in this image).

The average ping time can be used as an indicator of connection quality for a Driver Station-Robot Controller pair. If the wireless connection between the Driver Station and the Robot Controller does not have a lot of noise, traffic, or interference, then the average ping time is generally smaller. If the noise,

traffic, or interference increases, then the Wi-Fi devices on a channel tend to resend packets more frequently, which causes the average ping time to increase.

At an event, an FTC/CSA/WTa should have access to a pair of Android devices (preferably an Android device that supports Wi-Fi channel changing through the FTC Robot Controller app) that they can use to monitor the wireless connection quality on a Wi-Fi channel at the venue. If the ping time is low (on the order of 5 msec or less) then the wireless connection quality is very good (exceptional). If the observed ping time hovers at a high value (such as 250 msec or more) then the wireless connection quality is poor and the FTA/CSA/WTa should try and identify the cause of the poor connectivity.

Note: that the average ping time only provides a measure of quality for the operating Wi-Fi channel. It does not indicate quality for the entire set of channels. For example, if you have a pair of devices that are operating on channel 1, the ping times observed for this pair of devices is primarily useful for monitoring the wireless quality of channel 1. If you wanted to measure the wireless quality for channel 6 or 11, then you would have to change the operating channel for your devices, reconnect them, then look at the ping times for the newly selected channel.

If a team is encountering issues with communicating with their robot, look at the ping times on their Driver Station to determine if the Robot Controller has a responsive connection (ping times less than 50 msec, preferably on the order of 5 msec).

Using the average ping time is a convenient way to determine if the wireless channel is clear and relatively noise-free. If the ping times are low, then the channel is most likely free of other Wi-Fi and non-Wi-Fi traffic.

Important Note: the observed ping time is also affected by the availability of the Robot Controller to respond to the heartbeat messages from the Driver Station. If the Robot Controller is busy (for example, because it is blocking in a portion of an improperly written OpMode) and it is unable to respond in a timely manner to the Driver Station, the observed ping times will be higher, even if the wireless connection is strong.

Is the Wi-Fi Channel Too Busy?

In addition to ping times, an FTA/CSA/WTa can use other tools, such as the Aircheck meter or the MetaGeek inSSIDer application, to get a more detailed view of the wireless activity on a Wi-Fi Channel. If you are at an event and you have access to a device like the Aircheck meter, then you can examine the activity level for each wireless channel and determine if a channel is being saturated.

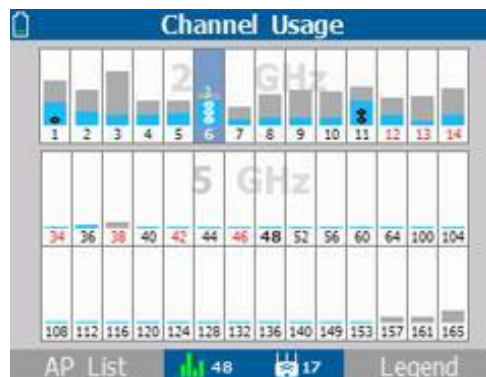


Figure 32 - The Aircheck meter shows Wi-Fi (light blue) & non-Wi-Fi (gray) activity on each *channel*.⁴

⁴ Image from the Fluke website (<http://www.flukenetworks.com/enterprise-network/network-testing/AirCheck-Wi-Fi-Tester>) downloaded on 9/21/15.

In Figure 32, you can see the Wi-Fi (shaded in light blue) and non-Wi-Fi (shaded in gray) activity on each Wi-Fi channel. You can see that channel 3 in the example has a lot of non-Wi-Fi (gray) activity and that the channel is very busy. You can also see that channel 7 has less activity and is not very busy.

Also note that adjacent Wi-Fi channels for the 2.4GHz band overlap, so activity on one channel might have a negative effect on activity of a nearby channel.

If you notice high activity levels, then you can try to find and disable the devices that are causing the interference. You can also try to move the Driver Station-Robot Controller devices to a different, less busy channel.

Potential Sources of Wi-Fi Interference

Potential sources of Wi-Fi interference include the following:

- Wireless access points that belong to the venue (such as an access point used to provide wireless access within a school).
- Unauthorized team or spectator access points.
- Mobile hotspots.
- Wi-Fi enabled cameras or other devices (such as Game Boy consoles, etc.).

Potential Sources of Non-Wi-Fi Interference

Potential sources of non-Wi-Fi interference include the following,

- Bluetooth devices (which also operate in the 2.4GHz band of the spectrum).
- Wireless audio/visual systems (including wireless microphones and cameras).
- Cordless telephones and headsets.
- Remote control cars, helicopters, drones, and planes.
- Microwave ovens.

Are There Too Many Robots Operating on the Same Channel?

Related to a channel being too busy, if there are too many robots operating on a channel, then the average wireless connection quality might suffer. *FIRST* has done stress testing where we had a high number of Driver Station-Robot Controller devices operating reliably on a single Wi-Fi Channel. We could operate close to 50 pairs on a single 2.4GHz Wi-Fi channel. However, in practice, the number of robots that can operate on a single channel will vary with a variety of factors. If there is a lot of external wireless interference on a channel, then the number of robots that can operate on a channel will decrease.

If you are at an event and you suspect that there are too many robots operating on a single channel, you can try to distribute the robots evenly across available, less busy channels. Ideally, the 2.4GHz channels should be spaced at least 5 channel-widths apart, to avoid any overlap. However, if necessary, you can move robots to overlapping channels.

Is there a Wi-Fi Suppressor Operating in the Vicinity?

Many IT organizations use Wi-Fi suppressors to suppress any unauthorized Wi-Fi access points operating in a venue. These suppressors have a list of authorized wireless networks that can operate within the venue. If the suppressors detect an unauthorized wireless network, it will send out packets to disrupt the operation of the unauthorized network. Many of these suppressor functions are built-in to modern wireless access points.

Each Driver Station-Robot Controller pair establishes its own Wi-Fi network. If there is a Wi-Fi suppressor operating in the vicinity, then the suppressor disrupts the operation of any Driver Station-Robot Controller in the area. If you suspect that there is a Wi-Fi suppressor operating at a venue, then you need to work with the venue's IT staff before the day of the event to disable the suppressor for any scheduled *FIRST* Tech Challenge events.

Note that even though Wi-Fi suppressor technology is gaining popularity, according to the FCC,⁵ federal law "prohibits the operation, marketing, or sale of any type of jamming equipment, including devices that interfere with cellular and Personal Communication Services (PCS), police radar, Global Positioning Systems (GPS), and wireless networking services (Wi-Fi). There is an FCC Enforcement Advisory that warns that Wi-Fi blocking is prohibited."⁶

Are the Wireless Radio Signals Being Blocked by Metal?

If you are at an event and you suspect that one or more robots are having wireless issues (higher ping times, less responsive robots, etc.), then you should make sure that radio signals from the Driver Station and the Robot Controller are not being blocked or screened by large sheets or pieces of metal.



Figure 33 - Metal music stands like this one can block, reflect or attenuate the signals to/from the Driver Station.

For example, if the Robot Controller Android device is mounted deep within the frame of the robot and if there are pieces of sheet metal or aluminum channel blocking or obscuring the Android device, then the radio signal from the Robot Controller might get blocked and/or reflected. This can attenuate/reflect signals to and from the Robot Controller. Also, if the Android device is mounted directly onto a metal plate on the robot, the signal can also be blocked, reflected, or attenuated (remember, the antenna on many Android devices is located near the back pane of the device).

Similarly, if the Driver Station Android device is placed on something like a sheet metal plate, or if the Driver Station device is enclosed in metal housing, then the signals to and from the Driver Station might be blocked, reflected or attenuated.

As an example, *FIRST* conducted some experiments using a metallic music stand as a Driver Station stand for a ZTE phone. *FIRST* used Wireshark to monitor the activity with and without the music stand in place. *FIRST* observed that the wireless retry rate for the Android device sitting on the music stand was about *twice* as high as the wireless retry rate for the same device when it was sitting on a wooden table. Even though the human driver during the test did not perceive any difference in responsiveness of the robot when the music stand was in place, the Wireshark data indicated that the quality of the

⁵ See <https://www.fcc.gov/encyclopedia/jammer-enforcement> (accessed on 08/07/22).

⁶ See https://apps.fcc.gov/edocs_public/attachmatch/DA-15-113A1.pdf (accessed on 08/07/22).

wireless connection was worse whenever the music stand was in place. *FIRST* attributes the increase in retry rate to the metal music stand attenuating/reflecting the radio signals to/from the Driver Station.

Ideally, the Robot Controller and Driver Station devices should be mounted in a way that protects the devices, but does not block the radio signals traveling to/from the devices. In most cases, the radios will work fine, even if they are partially (or almost fully) obscured by metal. However, whenever the radios are obscured, the signals are attenuated, therefore if the attenuation is high enough, the devices might start to experience wireless connection problems.

Is There Malicious Activity Occurring?

Unfortunately, it is possible for a motivated individual to disrupt Wi-Fi networks using tools and techniques that are described on the Internet. This vulnerability is true for most Wi-Fi networks, including the Wi-Fi networks that are established by the FTC Driver Station-Robot Controller pairs.

There is an amendment to the 802.11 standard (802.11w) that makes it more difficult to conduct some of these types of attacks. The 802.11w standard is the default setting for the REV Driver Hub and the REV Control Hub, but unfortunately this wireless standard is not yet available on Android smartphones. For now, the Android smartphone devices used at *FIRST* Tech Challenge events are vulnerable to certain wireless attacks.

There are tools that can help detect when certain wireless attacks have occurred. For example, Section 13 of this document describes how to use Wireshark to look for clues that indicate that certain wireless issues are present. However, these tools are not always available at many *FIRST* Tech Challenge events.

If you are at an event where you suspect that some malicious activity is occurring, you can try to use any available tool to identify the party that is conducting the malicious activity. You can also rely on good, old-fashioned “detective work” to look for suspicious activity in and around the Competition Fields. Also, if you believe malicious activity is occurring, you can remind spectators and participants that this type of behavior is ungracious and punishable by disqualification from the event and possibly the season.

Determining if Wi-Fi Interference Warrants a Match Replay

The most critical responsibility of a *FIRST* Technical Advisor (FTA), Control System Advisor (CSA) or Wireless Technical Advisor (WTA) is deciding if the wireless interference during a match is significant enough to warrant a replay of the Match. This is a difficult and subjective decision to make. The Competition Manual states that “Matches are replayed at the discretion of the Head Referee only for a failure of an Arena Element or verified Wi-Fi interference that was likely to have impacted which Alliance won the Match.”

To make a recommendation to the head referee for a match replay, the FTA (or CSA or WTA) must have sufficient proof of such Wi-Fi interference.

Scenario 1: High ping times for a robot

Consider the following scenario, a team during a match complains of an unresponsive or sluggish robot. The team states that the ping times (as displayed on the FTC Driver Station app) were high (> 200 msec) for most of the match. Does this situation warrant a Match replay?

In this instance, without additional information or evidence, an FTA would **not** have sufficient proof to recommend a Match replay. High ping times for a robot can be caused by numerous factors.

- For example, if the Robot Controller app is running an improperly written OpMode (that blocks, for example, in the main program thread and prevents other background tasks from periodically

running), the Robot Controller might become unresponsive, and the team will experience control issues.

- Or, as another example, if the Robot Controller is physically mounted in an area where its radio signals are blocked by metal, the Driver Station might have a hard time “hearing” the Robot Controller and the average ping time can increase noticeably.

For this scenario, the FTA needs additional proof before making the recommendation to replay a match:

- During the match, did the FTA have a Robot Controller – Driver Station pair that was used to monitor the same channel as the team’s robot?
 - If so, what were the observed ping times for the other Robot Controller – Driver Station set?
 - Did the other set also experience sustained, high ping times during the match?
- Did the FTA have a device such as an Aircheck monitor or the MetaGeek Wi-Spy device (with the inSSIDer software) monitoring the robot’s channel during the match?
 - If so, did the device indicate that the wireless channel was extremely busy?
 - Was there noticeable Wi-Fi and/or non-Wi-Fi interference on the channel?
- Did other teams who were operating on the same wireless channel during the match also experience sustained high ping times and poor control of their robots?
- Did the FTA have a laptop running Wireshark in monitor mode to capture packets during the match on the channel in question?
 - If so, what were the observed retry rates for the robot that experienced the unresponsiveness?
 - Also, what were the observed retry rates for the other robots on that channel?
 - Was it only the one robot that had a high retry rate on the channel or did all the robots have high retry rates?

Scenario 2: Robot Controller unexpectedly disconnects from the Driver Station

During a match, a team’s Robot Controller unexpectedly disconnects from the Driver Station and the team loses the ability to communicate with and control their robot. Does this event warrant a Match replay?

Unfortunately, for this scenario, unless the FTA has solid proof of wireless interference causing the disconnect, it is difficult for the FTA to recommend a Match replay.

There are several reasons why a Driver Station can lose wireless connectivity to its Robot Controller. These reasons include (but are not limited) to the following:

- Low battery on the Robot Controller or Driver Station Android device.
- Improperly configured Robot Controller or Driver Station Android device (for example, the Robot Controller is also connected to another device and/or another network during the match).
- Disruption due to an electrostatic discharge (ESD) event or a physical impact to the Robot.
- High current draw from motors or servos causing a “brown-out” which temporarily affects the Robot Controller’s Wi-Fi antenna.
- Loose or disconnected wire supplying power to the REV Control Hub.
- Wire with damaged insulation contacts the robot structure.

To recommend a Match replay, the FTA would have to have reliable evidence that demonstrates that Wi-Fi interference caused the observed disconnect:

- A Wireshark capture from the match that shows DEAUTH packets that look like they were sent from the Robot Controller (i.e., that have the same MAC address/BSSID as the Robot Controller).
- Or, if an FTA/CSA/WTa observed a very large spike in activity on the Robot's channel during the match using a tool like the MetaGeek WiSpy or the netAlly Aircheck G3 Pro.

Additional Thoughts on Recommending a Match Replay

To recommend that a Match be replayed, the FTA must have reliable evidence that would support this recommendation. For a high-profile event it is important that the FTA/CSA/WTa take steps before the event and have some tools available during the event to help monitor the wireless environment.

If resources are limited, then using a spare set of Robot Controller and Driver Station devices to keep track of the ping times is a relatively easy way to monitor the wireless environment. For larger and higher profile events, the event host and the technical volunteers should consider using some of the more sophisticated tools described in this document to monitor the wireless spectrum at their event. Unfortunately, these more sophisticated tools can be expensive and require an investment in time to learn how to use them effectively.

Accommodating a Large Number of Robots at an Event

Wi-Fi Event Planning Guide

The wireless Control System is a point-to-point system. This means that each Driver Station-robot pair will establish its own Wi-Fi network at an event (see Figure 2). If there are a large number of robots in a venue, then there will be a large number of wireless networks operating in the venue. If there are a large number of wireless networks operating in a small area, then there could be interference between the networks.

At smaller events with lower numbers (< 30 or 40) of robots, the likelihood of significant interference caused by the Robot-Driver Station activity is small. If there is not any other source of interference (such as Bluetooth devices operating on the Field or wireless audio/video systems broadcasting on the same frequency) the *FIRST* Tech Challenge Control System should be able to operate properly.

At larger events (>30 or 40 robots) some steps might need to be taken before the event and during the event to help keep things running smoothly. *FIRST* Tech Challenge has published a [Wi-Fi Event Planning Guide](#) that contains detailed steps that a technical volunteer can take to help keep the wireless environment operating smoothly.

Distributing Robots Across Multiple Channels

Wi-Fi Channel Overlap

If there is an event that will have a large number of robots (> 40) in a small area, you should consider distributing the Robots across multiple channels. Ideally, the lower the number of robots there are per channel, the less traffic and interference there will be per channel. Note that 2.4 GHz Wi-Fi channels that are less than 5 channel widths apart overlap (see Figure 23).

Ideally, for 2.4 GHz Wi-Fi connections, you should distribute your robots on channels that are at least 5 channel widths apart. For example, if you were to configure one group of robots to channel 1 and a

second group to channel 5, the two groups of robots would overlap slightly since the second channel is only 4 channel widths away from the first channel. If the second group of robots were moved from channel 5 to channel 6, then the two groups would no longer overlap since they are 5 channel widths apart.

Sometimes it might not be possible to space your robots 5 channel widths or greater apart. For example, one portion of the spectrum might be very noisy, and the robots are unable to operate on channels in or near that portion of the spectrum. In this case, it still might be beneficial to place the robots in groups on separate, overlapping channels. Even though the channels overlap slightly, placing the robots onto these channels might produce lower ping times and more responsive robots when compared to keeping all the robots on the same channel.

Factors to Consider when Selecting Wi-Fi Channels

If you would like to configure your robots to operate on more than one channel, here are some factors to consider when doing your planning:

1. **How many robots will be present?** The data rates for the control streams of the robot are low. If the wireless environment at your venue is clean, then a single channel should be able to support a pretty large number of robots. In our testing, we could run 46 pairs of Android devices in a very tight area (approximately 14' x 14') with good reliability and responsiveness.

If your event will have a modest number of robots (less than 40), and if your wireless environment is clean, then you probably do not need to worry about moving robots around to different channels.

If you do have a large group of robots, then you should consider dividing them up, so you have a maximum of 35 to 40 groups per channel if possible. For example, if you have 70 robots, you can divide them into two groups of 35. You can also break up a large number of robots into even smaller groups and then place them onto multiple, overlapping channels if needed.

2. **Before you select your channels, are the target channels clear?** Before you move your robots to a specific channel, you should do some tests on the channel to verify that it is clear.
 - a. **Use Wi-Fi Analyzer or a similar tool:** You can use a tool like Wi-Fi Analyzer to see how many access points are present on a channel. Remember, Wi-Fi Analyzer only shows you the visible (non-hidden) wireless network. Also, Wi-Fi Analyzer does not show you how busy a channel is, it only shows you what visible Wi-Fi networks are on a channel.
 - b. **Use a pair of Android devices to monitor ping times:** If a target channel looks relatively clean, you should use a pair of Android devices running the FTC Driver Station and FTC Robot Controller apps to monitor the ping times on the target channel. You will need a pair of Android devices that support channel changing (such as approved FTC phones). You should switch to the target channel and test to make sure you can select and run an OpMode (like the NullOp sample OpMode). If the average ping times for the test Android devices are low (< 5 msec) then the channel is clear. If the average ping times are high (>50 msec) then there might be some type of interference on the channel.
 - c. **If available, use a more sophisticated tool to monitor the target channels:** If you have access to a more sophisticated tool like the Aircheck meter, you can use it to sweep a target channel. You want to use the tool to check for visible and hidden Wi-Fi networks. You also want to check to see how much Wi-Fi and non-Wi-Fi traffic is present on the channel. If the activity level is low on a target channel, then it should be safe to place your robots on the channel.

d. **What type of Android devices will the Teams be using?** FTC-approved smartphones support channel changing using the FTC Robot Controller app. Here is the list as of August 2023:

- i. Motorola Moto G4 Play (4th Generation)/Motorola Moto G4 Play
- ii. Motorola Moto G5
- iii. Motorola Moto G5 Plus
- iv. Motorola Moto E4 (USA versions only, includes SKUs XT1765, XT1765PP, XT1766, and XT1767)
- v. Motorola Moto E5 (XT1920)
- vi. Motorola Moto E5 Play (XT1921)

For a list of allowed phones and other hardware and software, teams should refer to the FTC Competition Manual and its updates.

UnPairing Then Re-Pairing the Driver Station to the Robot Controller.

Note that after you have changed the channel on your Robot Controller Android device, you might have to unpair your Driver Station from the Robot Controller, and then re-pair the Driver Station back to the Robot Controller.

To unpair the Driver Station from the Robot Controller, launch the **Settings** menu from the Driver Station app and select the **Pair with Robot Controller** item.

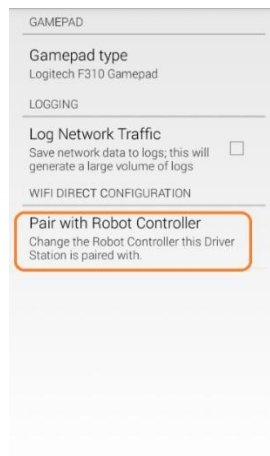


Figure 34 - Select Pair with Robot Controller.

From the Pair with Controller screen, select **None** to unpair your phone.

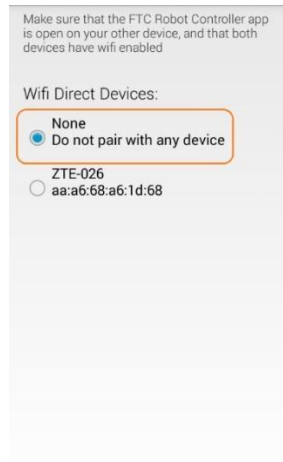


Figure 35 - Select "None" to unpair the device. Use the back arrow to return to the main screen.

Use the back arrow to return to the main Driver Station screen. The screen should now indicate that the Driver Station is not paired with any Wi-Fi Direct device.

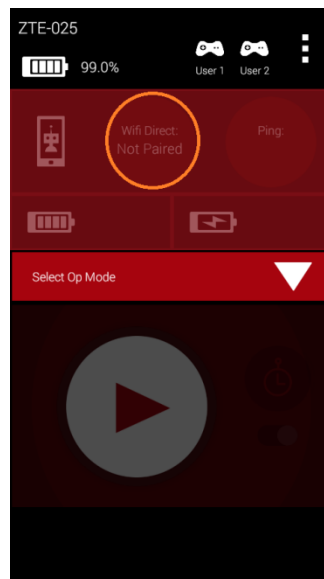


Figure 36 - The Driver Station should now be unpaired from the Robot Controller.

To re-pair the two devices, launch the **Settings** menu from the Driver Station app again and select **Pair with Robot Controller** again (see Figure 34). Find the listing for your Robot Controller Android device (in this example "ZTE-026") and select this item.

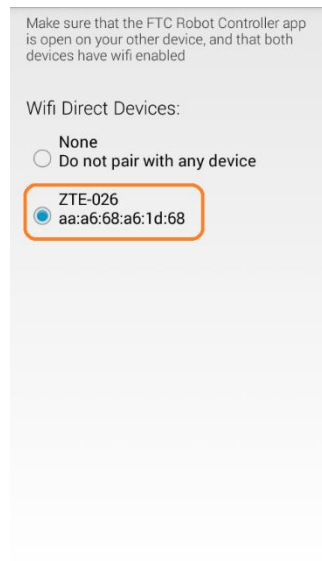


Figure 37 - Select your target device (in this example ZTE-026), then use the back arrow to return to the main screen.

Note: Your Robot Controller smartphone Android device might prompt you to make sure you approve the connection request. On the Robot Controller device, click on the Accept button to approve the connection request.

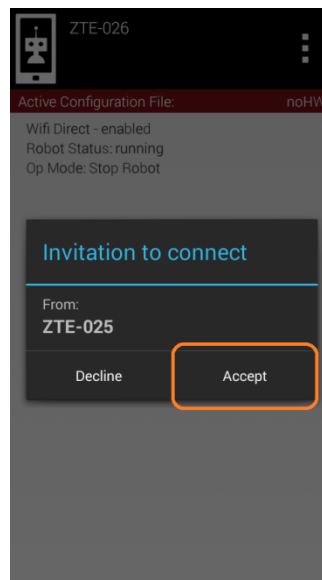


Figure 38 - Click Accept to approve of the connection request.

Once you have accepted the connection request, the Driver Station screen should display that it has successfully connected to the Robot Controller.

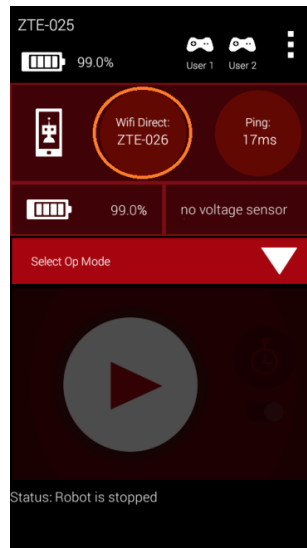


Figure 39 - Once the connection request is accepted, the Driver Station should connect to the Robot Controller.

Changing the Channel Using an Approved Motorola Smartphone

If you are using an approved Motorola smartphone as your Robot Controller, you can use the channel change function that is built into the FTC Robot Controller app to change the Wi-Fi Direct operating channel. From the Robot Controller app, launch the **Settings** menu and select the **Change Wi-Fi Channel** option.

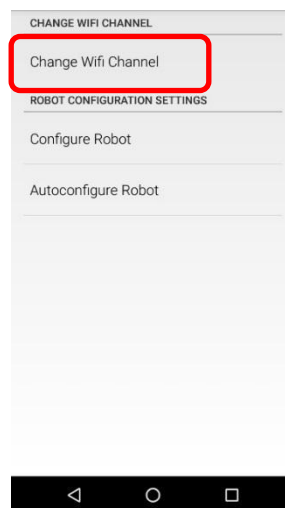


Figure 40 - Launch the Settings menu, then select Change Wi-Fi Channel (highlighted above in red).

In the Change Wi-Fi activity, select the target channel from the drop-down list of available channels.

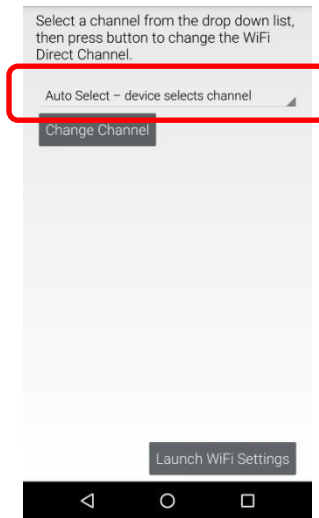


Figure 41 - Select your desired operating channel from the drop-down list (highlighted above in red).

Once you have selected the desired target channel, push the **Change Channel** button to change the operating channel. If the operation is successful, you should see a toast appear, indicating that the channel was successfully changed.

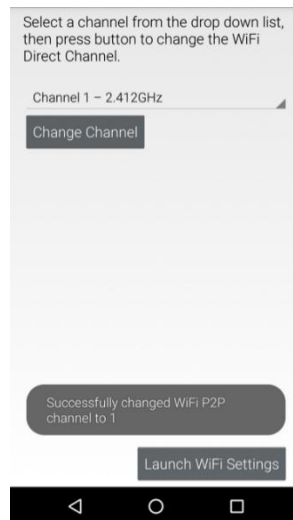


Figure 42 - A toast should appear indicating that the channel change was successful.

Once the channel change has completed, use the Android back arrow button to return to the main screen. The Driver Station should be able to reconnect to the Robot Controller using the new operating channel.

Mitigating Disruptions Due to Electrostatic Shocks

Electrostatic discharge (ESD) events have the potential to disrupt the normal operation of a competition robot. The [Managing Electrostatic Discharge Effects](#) article on [ftc-docs](#) provides a comprehensive discussion of this topic. Key takeaways include:

- ESD is bad for robots.
- To mitigate risks:
 - Treat tile floors with Anti-Static Spray or Water
 - Ferrite Chokes may be used to dampen ESD effects
 - Electrically isolate electronics from the metal frame of the robot
 - Ground electronics to metal frame using approved grounding cables

Troubleshooting Common Issues

FIRST Tech Challenge Driver Station

Gamepad is Not Recognized

If a gamepad is recognized by the *FIRST* Tech Challenge Driver Station app, then whenever there is activity with that gamepad, the appropriate gamepad icon in the upper right-hand corner of the *FIRST* Tech Challenge Driver Station main screen will be highlighted in green.

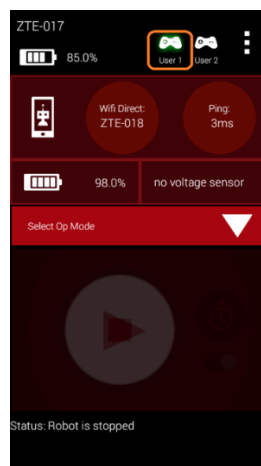


Figure 43 - The gamepad icon will be highlighted in green if gamepad is recognized and active.

If you encounter a team who is having problems with input from the gamepad, check the following items:

1. If a team is using Logitech F310 gamepads, then make sure the button on the bottom side of the gamepad is set to the "X" position (Xbox emulation mode). NOTE: In SDK 9.1 and newer, this is not necessary – both the "X" and "D" modes are supported.
2. Make sure the gamepad has been designated as either driver (user) #1 or driver (user) #2. To designate a gamepad as driver #1, press the START button and A (green colored) button on the F310 gamepad. To designate a gamepad as driver #2, press the START button and the B (red colored) button on the gamepad. **Important note:** the version 5.5 and greater Driver Station software automatically recognizes what type of FTC-approved gamepad a team is using. Teams no longer need to configure the gamepad type manually.

3. Check the wired connection. If the gamepad was temporarily disconnected from the Driver Station, then the driver will have to re-designate which driver they would like to be by pushing START and A (to be driver #1) or START and B (to be driver #2). Note that in some instances, the Driver Station can automatically recover a lost joystick, but if both gamepads are of the same type, and if both gamepads get disconnected at the same time, the user will have to re-designate the driver position for each gamepad.
4. Disable Advanced Gamepad Settings. If the application is crashing when the gamepad is plugged in, or the gamepad is not being detected properly, disabling the Advanced Gamepad Settings is one step to triage issues. Advanced Gamepad disabling has only been proven effective with troubleshooting first-generation DualShock 4 controllers. Advanced Gamepad settings are not required to use the gamepad.

Gamepad Joysticks Were Not in Neutral Position When Connected to Driver Station

The teams can connect up to two gamepads to the Driver Station Android device. Each gamepad has a pair of joysticks that the team drivers can use to control their robot. The gamepads are usually connected directly to a REV Driver Hub or through a non-powered USB hub to the USB Micro OTG port on an Android smartphone Driver Station. When the gamepads are first connected to the Android device, the Android device calibrates the zero or neutral position of the two analog joysticks on each gamepad.

If a user has deflected or moved the joysticks while they are being plugged into the Driver Station, the Driver Station might use a non-zero position of the joysticks as the calibrated reference point. This can cause unexpected behavior when an OpMode is run. For example, if the user starts the OpMode and the robot starts driving without the user touching the joysticks, one possible cause could be an improperly calibrated joystick.

To clear this problem, stop the OpMode, disconnect the gamepads from the Driver Station, and then reconnect the gamepads, making sure the joysticks are in their neutral position.

Mode Button on F310 Gamepad is Pressed

The Logitech F310 gamepads have a button on them labeled as “MODE”. When this button is pressed, the little green LED next to the button should toggle on. Teams usually do NOT want to have this button enabled. When the F310 gamepad has this MODE button enabled, the left-hand side controls change so that the outputs of the left joystick and the D-pad are swapped. This often confuses teams if the MODE button gets enabled during a match. If a team is experiencing weird behavior from their gamepads, verify that the MODE button is not enabled. If it is enabled and the green LED is lit, push the MODE button again to disable this function.



Figure 44 - There is a button marked “MODE” on the F310 gamepad controller.

Gamepad Disconnects While Joysticks are in a Non Neutral Position During an OpMode Run

If a team is running a driver-controlled OpMode, and one of the gamepads gets disconnected from the Driver Station while the joystick of the gamepad was in a non-neutral position, the control system might remember the last driver command. If the team stops the OpMode using the STOP button on the driver station, then reconnects the gamepad to the Driver Station and runs the OpMode again, the OpMode might receive the remembered, non-neutral gamepad input and cause the robot to move. To correct this situation, the user should move the joysticks on the gamepad around briefly and then let them move back to their neutral positions.

Driver Station Goes to Sleep While OpMode is Running

This problem will typically occur if the Sleep timer is set too low, and the Driver Station goes to sleep while an OpMode is running. To address this issue, go to the phone's Settings -> Display -> Sleep, and set it to sleep after 10 or 30 minutes of inactivity. **NOTE:** This is not necessary on the Driver Hub.

Driver Station Powers Off Unexpectedly

This problem will occur when either Android device has a low battery. If the device is unexpectedly shutting off, check that the device has an adequate battery charge state.

Unable to Find a Specific OpMode in the Driver Station's List of Available OpModes

If the team used Android Studio and the *FIRST* Tech Challenge SDK to create an OpMode but they are unable to find this OpMode on the *FIRST* Tech Challenge Driver Station's list of available OpModes, then you should ask the team if they remembered to register their OpMode in the `FtcOpModeRegister` class. If they created the OpMode, but did not register it, then it will not be visible on the Driver Station.

Gamepad Left Joystick is Not Working

If a gamepad's left joystick is not working and the right joystick is working, the probable cause is the "Mode" button adjacent to the left joystick was activated. When the green light next to the Mode button is illuminated, the gamepad swaps the functionality of the directional pad (D-pad) with the left joystick. Press and release the Mode button to turn the light off and restore the functionality of the left joystick.

Robot Controller

User Code Threw an Uncaught Exception: null

This error occurs when a method is called on an object that is null at the time the method was called. To address this issue, first look at the robot log file to find where to search for the problem. To access the log files, open the settings in the Robot Controller app, and click "view logs." Then, scroll up until a block of red text appears, and look for the line that says

```
"com.qualcomm.ftcRobotcontroller.opmodes.YourOpmodeName.loop(YourOpmodeName.java:XX)"
```

The XX will have a line number where the null error occurred in the code, and is a good starting point to addressing the issue. For example, if this line is accessing a method of an `ElapsedTimer` object, make sure that the object has been instantiated with `"objectname = new ElapsedTimer();"` somewhere in the code before this line.

User Code Threw an Uncaught Exception: number XXX is invalid;

This is an exception that is typically thrown when a motor or servo is set to a value that is less than -1 or greater than 1. To find which line of code the exception was thrown on, check the robot logs by opening the settings in the Robot Controller app, and clicking "View logs." Then scroll up to the first block of text that is red, and find the line that says:

`"com.qualcomm.ftcRobotcontroller.opmodes.YourOpModeName.loop(YourOpModeName.java:XX)"`

The XX will be the line number where the exception was thrown, and is a good starting point to addressing the issue. Remember: the `setPower` method for `DcMotors` can only be passed a value from -1 to 1, and the `setPosition` method for `Servos` can only be passed a value from 0 to 1. If this value is likely to be outside of the range, the `Range.clip()` method can be used to constrain the value to a specified range.

Unable to find a hardware device with the name "..."

A very commonly encountered error occurs when the user tries to run a specific OpMode, and the *FIRST* Tech Challenge Robot Controller app complains that the "User code threw an uncaught exception: Unable to find a hardware device with the name "..."" where "..." is the name of the hardware device that the OpMode is trying to access.

This error will occur if the OpMode specifies a name for a device that does not match the corresponding name for the device in the *FIRST* Tech Challenge SDK's hardware map. This error will occur for apps created using Android Studio.

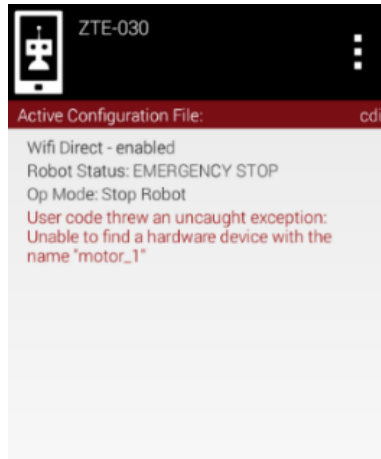


Figure 45 - The name used by the OpMode must match the name used in the configuration file for the device exactly.

If you are at an event and you encounter this error message, ask the team to verify that the name that they use in their OpMode to reference a hardware device matches the name specified for that device in the configuration file of their Robot Controller. The spelling is case sensitive so the names must match *exactly*.

Common Programming Errors

It can be difficult and frustrating trying to debug problems with a team's OpMode. However, there are some commonly encountered programming errors that can cause problems with a team's OpMode.

Neglecting to Insert `waitForStart()` Statement

For a `LinearOpMode` object, it is important that the programmer included a `waitForStart()` statement in the OpMode. Any statement in a `LinearOpMode` object that occurs after the `waitForStart()` statement will be executed after the user touches the START arrow on the Driver Station. If the programmer forgot to include a `waitForStart()` statement, the robot might behave unpredictably. For example, the robot might start moving instantly as soon as the OpMode is selected.

Uninterruptible Threads

Teams often incorporate loops in their robots' OpModes. Advanced programmers might also like to spawn threads within their OpModes to execute commands in parallel to the main OpMode process. If an OpMode contains a loop or spawns a separate thread, it is important that the loop or the thread is written such that the loop or thread can be interrupted by the Java application when necessary.

This can be illustrated by an example. Suppose a team writes the following OpMode:

```
// turn on motors.
motorL.setPower(0.2);
motorR.setPower(0.2);

// use while loop to run the cycle indefinitely.
while(true) {
    // stop if touch sensor is pressed..
    if(sensorTouch.isPressed()) {
        // stop OpMode.
        motorL.setPower(0);
        motorR.setPower(0);
        break;
    }
}
```

In this example, the motors are turned on, and then the OpMode loops indefinitely until the touch sensor is pressed. This OpMode is “uninterruptible.” If the user presses the STOP button on the driver station before the touch sensor is pressed, the while loop will continue to run and the OpMode will not be properly stopped. This can cause weird and unresponsive behavior of the robot. In this case, the robot will continue to run, and the Driver Station will continue to display the STOP button, even after the user has pressed the STOP button.

The FTC Robot Controller app can detect potential “runaway” OpModes. If the app detects what it thinks is a “runaway” or unresponsive OpMode, as a safety measure the app will record an error message in the log file and then it will crash itself. Unfortunately, this is the only way to stop an unresponsive thread. The user can check the robot controller to look for clues that their OpMode contains an infinite or uninterruptible loop.

```
06-23 19:58:27.022 E/RobotCore( 9265): user linear op
mode took too long to exit; emergency killing app.
06-23 19:58:27.022 E/RobotCore( 9265): possible
infinite loop in user code?
06-23 19:58:27.022 E/RobotCore( 9265):
```

Figure 46 - If the app detects a potential unresponsive OpMode, it will log an error then abort itself.

The correct way to prevent a “runaway” or unresponsive OpMode is to use an interruptible method within your loop or thread. The previous example could be made interruptible by using the `opModelsActive()` method as the loop condition, or by putting a sleep statement somewhere in the loop.

The following code would exit properly if the user touches the square STOP button on the Driver Station before the touch sensor was pressed.

```
// turn on motors.
motorL.setPower(0.2);
motorR.setPower(0.2);
```

```
// use while loop to run the cycle indefinitely.
while(opModelsActive()) {
    // stop if touch sensor is pressed..
    if(sensorTouch.isPressed()) {
        // stop OpMode.
        motorL.setPower(0);
        motorR.setPower(0);
        break;
    }
}
```

REV Robotics Control and Expansion Hubs

The REV Expansion Hub is a compact hardware controller that has 4 DC motor ports, 6 servo ports and multiple digital, I2C and analog ports. The REV Control Hub is a REV Expansion Hub with an integrated Android device.

Detailed information and specifications on the Control and Expansion Hubs and their use can be found in the *REV Robotics Control Hub Getting Started Guide* and *REV Robotics Expansion Hub Getting Started Guide*. The *Getting Started Guides* are available on the REV Robotics website (see <https://docs.revrobotics.com/duo-control/>).

This section contains important tips when troubleshooting a robot that is using a REV Robotics Control and/or Expansion Hub.

Resetting a REV Control Hub WiFi Password

A common issue for teams is resetting the password for the REV Control Hub WiFi password to a new value which is required as part of field inspection.

On the Driver Station app, go to the three dots (...) and select "Program and Manage"
Wait as it may take some time to load. (~30 seconds).

Follow steps from [Updating a Control Hub - REV Hardware Client \(revrobotics.com\)](https://docs.revrobotics.com/duo-control/) to update:

- Select three lines in upper right-hand corner.
- Select "Manage" from the that pops up.
- Scroll-down to Wi-Fi Settings and enter new password twice in the input fields
- Make sure to write down the password somewhere for safekeeping.
- "Select "Apply Wi-Fi Settings"
- Unpair and re-pair the RC and DS.

Power Cycle Time

FIRST recommends that the teams power down the device for a minimum of 5 seconds before powering their Control or Expansion Hub back on again.

Logic Level Converters

The REV Robotics Control and Expansion Hubs operate using 3.3V digital logic levels. The older Modern Robotics-compatible sensors and encoders operate using 5V digital logic levels. If a team would like to use a 5V device that was compatible with the Modern Robotics hardware controllers, then the team will need to use a *Logic Level Converter* (available from REV Robotics) to connect the 5V device to the 3.3V Control or Expansion Hub ports.

5V Modern Robotics-Compatible Encoders

If a team has a 12V DC motor with a 5V encoder that is designed to connect to a Modern Robotics DC motor controller, then the team will need one REV Robotics Logic Level Converter per 5V encoder to connect the encoder to the Control or Expansion Hub.

5V Modern Robotics-Compatible I2C Sensors

If a team has a 5V, Modern Robotics-compatible I2C sensor, then the team will need a REV Robotics Logic Level Converter plus a REV Robotics Sensor Cable Adapter to properly connect the 5V device to a Control or Expansion Hub.

LED Blink Codes

The REV Control and Expansion Hub have the following blink codes.

Firmware Version 1.07.00 or Higher LED Codes







LED Status	LED Description	When	Hub Status
	Solid Blue	At Boot	Control Hub has power; Battery is >7V and is waiting to initialize communications.
	Solid Blue	Anytime	Hub is waiting for communication with the Driver Station Host. Control Hub has power; Battery is >7V.
	Solid Green with one or more blue blinks every ~5 Seconds	Anytime	Hub has power and active communication with the Android Platform. The number of blue blinks is the same as the Hub's address. The factory default address is 2 ().
	Blinking Blue	Anytime	Keep alive has timed out. Fault will clear when communication resumes.
	Blinking Orange	Anytime	Battery Voltage is lower than 7V. Either the 12V battery needs to be charged, or the Expansion Hub is running on USB power only. This fault will clear when battery voltage is raised above 7V. This will not be overwritten by the keep alive timeout pattern.

Figure 47 - REV Hub - Blink Codes

Note that the REV Robotics website maintains an up-to-date blink code chart (see <https://docs.revrobotics.com/duo-control/troubleshooting-the-control-system/led-blink-codes>).

Troubleshooting Dual Expansion Hubs

Teams can daisy-chain two REV Robotics Expansion Hubs together to provide additional I/O ports for their robot. Details on how to configure and troubleshoot a dual Expansion Hub robot are included in the *REV Robotics Expansion Hub Getting Started Guide*. Details on how to configure and troubleshoot a dual Expansion Hub robot can also be found in the following online documentation:

https://ftc-docs.firstinspires.org/en/latest/hardware_and_software_configuration/configuring/configuring_dual_hubs/configuring-dual-hubs.html

If a team is having a problem with a dual Expansion Hub configuration, it is important that the FTA or CSA verify that each of the daisy-chained Expansion Hubs has a non-conflicting serial address. By default, all Expansion Hubs are assigned an address of 2 at the factory. If a team wants to connect two Hubs together, then the team must first change the serial address for one of the Hubs to prevent it from conflicting with the other Hub's address.

An FTA or CSA can connect each Expansion Hub individually (not daisy chained) to a Robot Controller and verify the blink pattern for each Hub to determine the Hub's address. Details on how to change the address of an Expansion Hub are included in the online documentation listed above.

Useful Tips and Tricks

Use a Pair of Android Devices to Monitor Wi-Fi Channel

It is useful to have a set of Android devices that you can use to monitor the activity on a wireless network. You can configure the *FIRST* Tech Challenge Robot Controller to have an empty configuration file (no devices attached). Connect to the *FIRST* Tech Challenge Robot Controller app with the *FIRST* Tech Challenge Driver Station app on the other Android device. Use the ping time feature as an indicator for the network quality on the operating channel of the devices.

Use the Log Files to Help Troubleshoot Problems

Both the *FIRST* Tech Challenge Robot Controller and *FIRST* Tech Challenge Driver Station apps log information that can be useful for diagnosing problems with the system. On the *FIRST* Tech Challenge Robot Controller app, if you touch the three vertical dots in the upper right-hand corner of the main activity, you can select the **View logs** option from the pop up menu that appears. This will display the log file information for the *FIRST* Tech Challenge Robot Controller app.

You can also use the Android Debug Bridge (ADB) to pull the files from the Android devices to your local computer. Details on how to use the ADB utility are available on the Android Developer website,

<https://developer.android.com/tools/adb>

On the Android device running the FTC Robot Controller app, the log file is stored at the device's top level (Main Storage) under this filename,

robotControllerLog.txt

On the Android device running the FTC Driver Station app, the log file is stored at the device's top level (Main Storage) under this filename,

driverStationLog.txt

You can use the ADB utility to copy the file to your local computer. For example, the following command line string will attempt to pull the Robot Controller log file from the phone to the current directory of the computer.

```
adb pull /robotControllerLog.txt .
```

For Windows users, you can also connect your Android device as a media device, and use the Windows File Explorer to browse and find the log file.

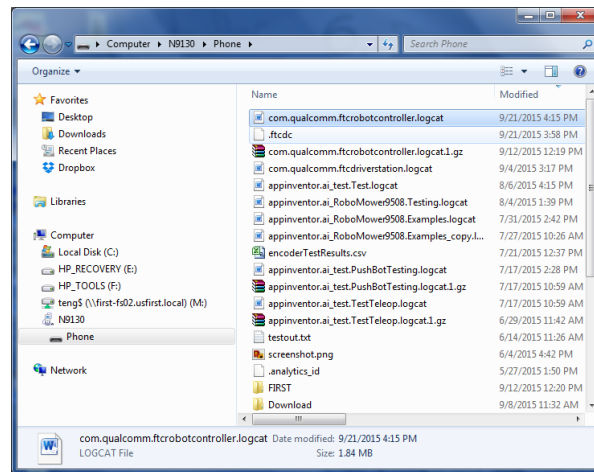


Figure 48 - Windows users can use the File Explorer to locate and copy the log file.

Wireshark

This section contains a couple of limited examples on how to use Wireshark to look for specific wireless issues. Detailed information on how to use Wireshark is beyond the scope of this document. For detailed Wireshark documentation, please visit the following web page: <https://www.wireshark.org/>

Creating a Capture Filter for DEAUTH Packets

In this section, we demonstrate how to create a Wireshark *capture filter* to capture deauthentication (DEAUTH) packets on a wireless channel. It is possible for a person to *spoof* (i.e., masquerade as) the MAC address of a device to disrupt the wireless communication with that device. If you have access to a machine with Wireshark, then you can use it to look for DEAUTH packets in your vicinity.

You can download the most current released copy of the software through the Wireshark website (<https://www.wireshark.org/>). Install the Wireshark software per the instructions (refer to the Wireshark website for documentation).

To use Wireshark to examine wireless data, you need to have a device that supports monitor mode operation of the wireless adapter. If you have a Mac computer running a recent version of macOS, you should be able to use the Mac's built in wireless adapter in monitor mode. For Linux devices, you need to consult the Wireshark and Linux documentation to figure out if your Linux computer will support monitor mode operation.

Once you have Wireshark properly installed and you have verified that you can run your wireless adapter in monitor mode, you should launch Wireshark. Note that some configurations require that you run Wireshark as a super user. If you have a configuration that requires super user status to run properly, it is possible to modify the permissions for your Wireshark installation so that you will no longer need to be a super user to run it. Please consult the official Wireshark documentation for details on how to do this.

The following screen shot shows the Wireshark user interface (version 1.10.2, Linux)

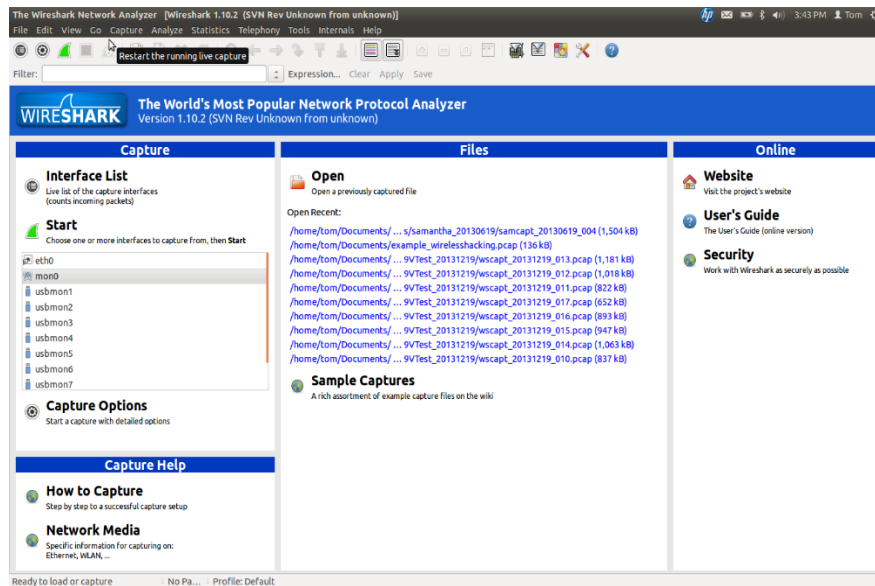


Figure 49 - Wireshark user interface.

Click on the **Capture** → **Options** menu (or press the button that looks like a small gear, which is second from the left on the button bar). The following image is a screen shot of the **Capture** → **Options** menu:

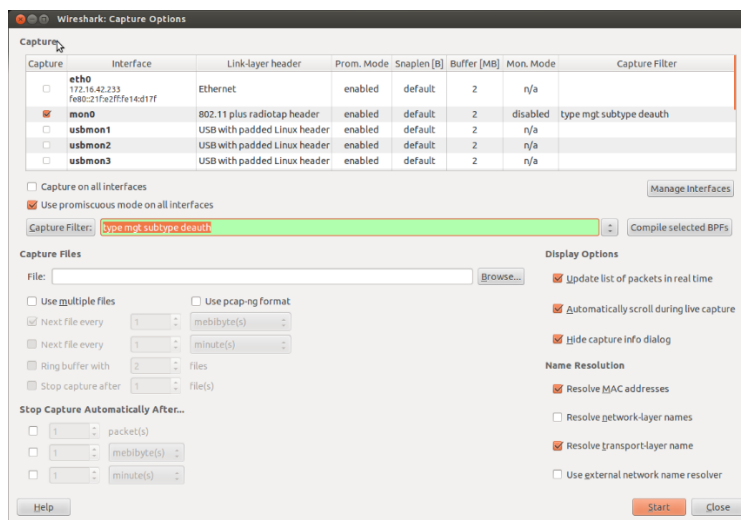


Figure 50 - Make sure you select the adapter that is running in monitor mode.

Make sure that the wireless adapter which is running in *monitor mode* is selected as the capture interface. In the screenshot above, the adapter "mon0" is the device that is running in monitor mode. Note for Mac users, you can double click on the wireless adapter in the list and check the **Capture packets in monitor mode** to place the selected adapter into monitor mode.

☐ Capture on all interfaces

☒ Use promiscuous mode on all interfaces

Capture Filter:

Capture Files Display Options

File:

☐ Use multiple files ☐ Use pcap-ng format ☒ Update list of files

[illegible]

43 of 60

Note that DEAUTH packets occur normally and a Wireshark capture might contain several normal (non-malicious) DEAUTH packets. If you are at an event and suspect that a DEAUTH attack might have occurred, you should note the approximate time of the attack as well as the team numbers of the robots on the field. You can use Wireshark to check if any DEAUTH packets were captured around the time of the suspected attack. You can also cross reference the source address of the DEAUTH packet to see if it matches any of the addresses of the robots that lost wireless connectivity on the field. During a DEAUTH attack, a hacker will *spoof* the MAC address of the target robot controller and pretend to be that robot controller and send DEAUTH packets to any devices (i.e., the Driver Station) that are connected to the robot controller's wireless network.

To stop the capture, you can press the red square icon to stop the capture. You can use the **File** → **Save** menu item to save the data to the hard drive.

Also, you can use the **Statistics** → **Summary** menu item to get some basic statistics about the capture data. Remember if you applied the DEAUTH capture filter, then only DEAUTH packets will have been captured by Wireshark. The following image shows the **Statistics** → **Summary** menu. In this example, 103 DEAUTH packets were captured during the session.

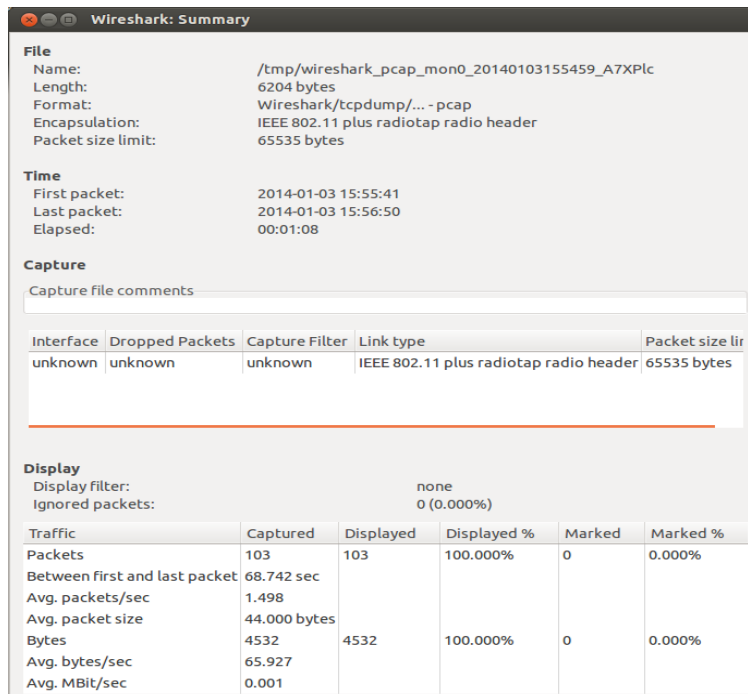


Figure 54 - 103 packets were captured in this example.

Viewing WLAN Traffic Statistics

You can use Wireshark to view Wi-Fi statistics for your captured data. If you disable any existing capture filters (to make sure that you capture all available data), and run Wireshark in monitor mode, you can capture and quickly review the data to get some basic information about your wireless environment. If you have captured wireless data in monitor mode, you can select **Wireless** → **WLAN Traffic** from Wireshark's menu to view some statistics about the captured data.

Figure 55 - The WLAN Traffic screen provides useful statistics about the captured data.

You can also look at the WLAN Traffic view to try and determine who is putting out the most data on the channel. This could be helpful if you are trying to identify sources of Wi-Fi traffic on a wireless channel.

Getting Additional Help

If you have questions about the *FIRST* Tech Challenge Control System, you can visit the *FIRST* Tech Challenge discourse forum and search for related posts or post your own questions:

- <https://ftc-community.firstinspires.org/>

There is also a [FIRST Tech Challenge Technology Slack Workspace](#) reserved for *FIRST* Tech Challenge technical volunteers (FTAs, CSAs, and WTAs) where these volunteers can ask questions and exchange information with other volunteers and with *FIRST* Tech Challenge staff. Prior to an event, *FIRST* Tech Challenge technical volunteers should visit this forum to get any last-minute information from other volunteers and from *FIRST* regarding event support and technical troubleshooting tips.

Tech Tips on Using Log Files

Introduction

One of the most useful features in the troubleshooting process is to have the ability to retrieve and access the log files on the Driver Station and Robot Controller devices. The system logs all types of info in these files and when an incident occurs, it is often helpful to review these files to see if we can notice any pattern or clues that can help diagnose the problem.

Helpful Tip: If you are on a Windows machine, the REV Hardware Client can be used to view and download log files from the Robot Controller Android device. See the REV Hardware Client Documentation here: <https://docs.revrobotics.com/rev-hardware-client/>

Verify the Date and Time

One important and often overlooked step that you can take to help with your troubleshooting is to verify the date and time on your Android devices. Ideally, you would like to verify that the dates and times on your devices match the local date and time. When the *FIRST* Tech Challenge apps record statements to the log file they include a timestamp that you can refer to when you are trying to troubleshoot a specific event.

When a problem with the robot occurs, you might not have the opportunity to view the log files and troubleshoot the problems right away. If you note the date and time of the incident, then at a later opportunity you can check the log files and read the timestamps to look for statements that occurred around the time of your incident.

The FIRST Tech Challenge Log Files

The logcat files are accessible on your phones. By default, the FTC Robot Controller and the FTC Driver Station apps store these files (as text files) in the top-level directory on your Android device. For the FTC Robot Controller app, if you are using Android Studio to write your app, the path on your phone to the log file is as follows,

```
/sdcard/ftcRobotControllerLog.txt
```

Note that this file path name is different than previous versions of the name (it used to be `/sdcard/com.qualcomm.ftcrobotcontroller.logcat`).

For the *FIRST* Tech Challenge Driver Station app, the path to the log file is as follows,

```
/sdcard/ftcDriverStationLog.txt
```


Note that this file path name is different than previous versions of the name (it used to be /sdcard/com.qualcomm.ftcdriverstation.logcat).

Viewing the FIRST Tech Challenge Robot Controller Log File

You can use the FTC Robot Controller app to browse log file information on the phone. From the main FTC Robot Controller screen, click on the three dots to bring up the main menu:

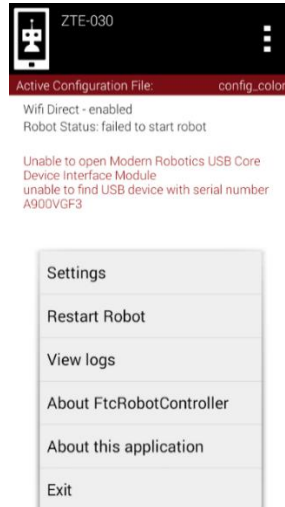


Figure 56 - Click on the three dots in the upper right hand portion of the screen then select View logs from the menu.

Select the **View logs** item from the menu to display the log file:

```
10-14 11:17:26.607 W/System.err( 4628): unable to
find USB device with serial number A900VGF3
10-14 11:17:26.607 W/RobotCore( 4628): Caught
exception during loop init:
com.qualcomm.robotcore.exception.RobotCoreExcepti
on: unable to find USB device with serial number
A900VGF3
10-14 11:17:26.607 E/RobotCore( 4628):
com.qualcomm.robotcore.exception.RobotCoreExcepti
on: unable to find USB device with serial number
A900VGF3
10-14 11:17:26.607 E/RobotCore( 4628):
com.qualcomm.modernrobotics.ModernRoboticsUsbU
tils(SourceFile:196)
10-14 11:17:26.607 E/RobotCore( 4628):
com.qualcomm.modernrobotics.ModernRoboticsUsbU
tils(SourceFile:112)
10-14 11:17:26.607 E/RobotCore( 4628):
com.qualcomm.modernrobotics.ModernRoboticsUsbU
til.openUsbDevice(SourceFile:90)
10-14 11:17:26.607 E/RobotCore( 4628):
com.qualcomm.hardware.HardwareDeviceManager.cr
eateDeviceInterfaceModule(SourceFile:190)
10-14 11:17:26.607 E/RobotCore( 4628):
com.qualcomm.hardware.HardwareFactory.create(Sourc
eFile:186)
10-14 11:17:26.607 E/RobotCore( 4628):
com.qualcomm.hardware.HardwareFactory.createHar
dwareMap(SourceFile:124)
10-14 11:17:26.607 E/RobotCore( 4628):
com.qualcomm.ftccommon.FtcEventLoopHandler.get
HardwareMap(SourceFile:93)
10-14 11:17:26.607 E/RobotCore( 4628):
com.qualcomm.ftccommon.FtcEventLoop.init(SourceF
ile:87)
10-14 11:17:26.607 E/RobotCore( 4628):
com.qualcomm.robotcore.eventloop.EventLoopManag
er(SourceFile:496)
```

Figure 57 - You can scroll up and down to view the statements in the log file.

Scroll up and down to view the log statements. The oldest statements appear at the top and the most recent statements appear at the bottom. Error messages are displayed in red.

Note that the **View log** feature only shows you an abbreviated version of the log file. While this is useful, it is sometimes more helpful if you can view the entire log file and search for older statements which might not be available through the **View log** feature. The subsequent sections of this manual contain instructions on how to grab the log file from the phone onto a computer.

File Navigation

Each FTC app stores log files at the top level of its Android device's storage system. Using the device's file manager app (if any), look inside Main Storage. The FTC Robot Controller app creates files named robotControllerLog.txt, sometimes with a digit appended. Similarly, the FTC Driver Station app creates files named driverStationLog.txt.

Third-party file manager apps can also be found at the Google Play store; many are free and perform well.

For some FTC Android phones, the only pre-installed file navigation tool is found inside a menu (not displayed on the Apps screen). Open the phone's Settings, then select Storage. Scroll to the bottom of the list, and touch Explore. This opens a basic file navigation screen, typically at the Main Storage level. Scroll down to see the FTC log files. Touch and hold a filename to activate a file management menu at the top of the screen. Or briefly touch a filename to open it with a selected text viewing app, if any.

Besides the standard full logs, the FTC Robot Controller app can also create shorter Match Logs. These are found in the *FIRST* folder, in the subfolder called matchlogs. Note: the *FIRST* folder contains robot configuration (XML) files and other useful subfolders, beyond the scope of this troubleshooting guide.

The Android phone may have a pre-installed or third-party app for viewing text files in various formats. But these FTC log files are large, each text line is long, and the phone's screen is small. It is usually easier to copy these files to your PC and use an application on the PC to browse the file.

Using Windows File Explorer to Locate the Log Files

If you are a Windows user you can use the Windows File Explorer to browse the contents of your Android phone and find the log files. The first thing you should do is connect the phone via a USB cable to your Windows machine.

Once the phone is connected, you want to make sure the phone is in media device mode rather than charging-only mode. For most FTC phones, swipe down from the top edge of the screen, and select "use USB to transfer files."

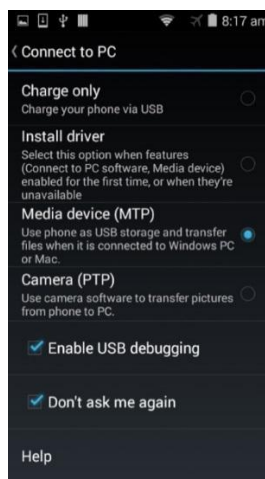


Figure 58 - Make sure your phone is in Media device mode.

If the phone is in Media device mode you can launch the Windows File Explorer to browse the contents of the phone. The phone should appear as a media device (e.g., "Moto E (4)") connected to your PC:

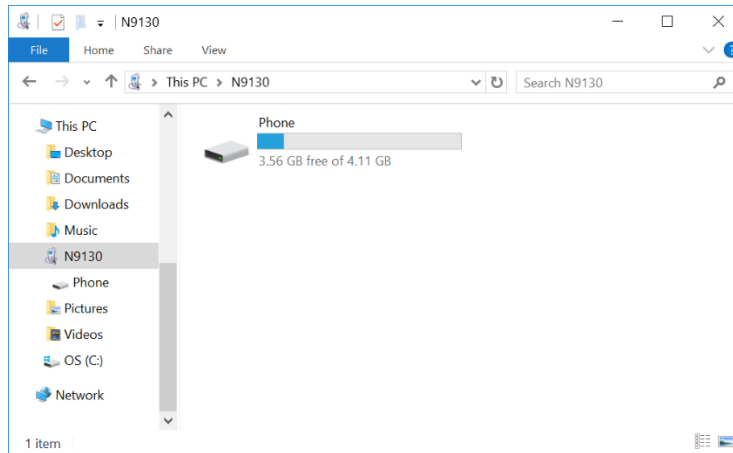


Figure 59 - The phone should appear as a media device connected to your PC.

You can double click on the phone's hard drive to open and browse the main directory of the phone.

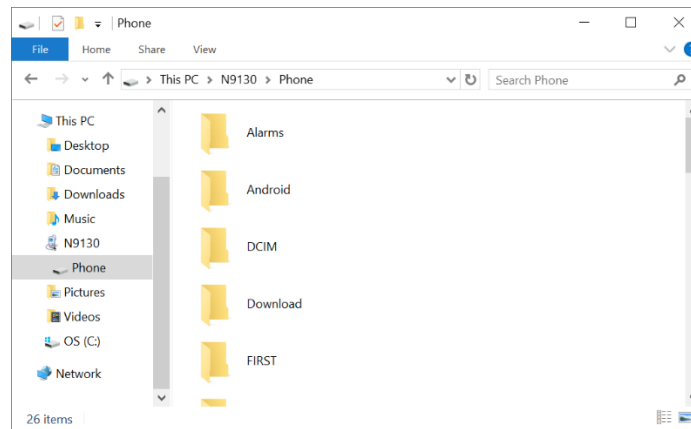


Figure 60 - You can browse the main directory of the phone.

When you open the phone's "hard drive" you are exploring the top level or Main Storage directory of your Android device. You can see that there is a *FIRST* subdirectory in this main directory. You can also scroll down to the bottom of the window to find the log files:

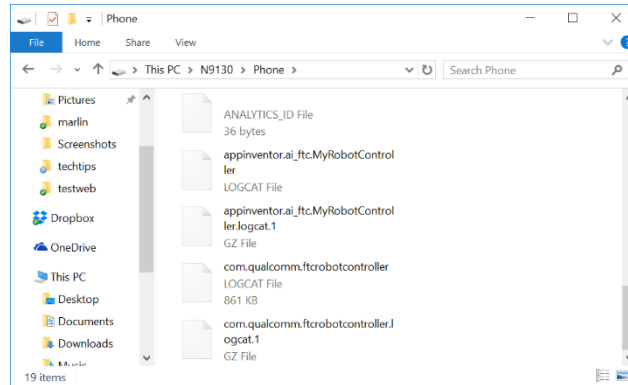


Figure 61 - The log files should be visible in this directory. With Windows File Explorer you can copy one or more log files and then paste the file onto your own PC's hard drive. This allows you to create a local copy of the log file on your computer that you can

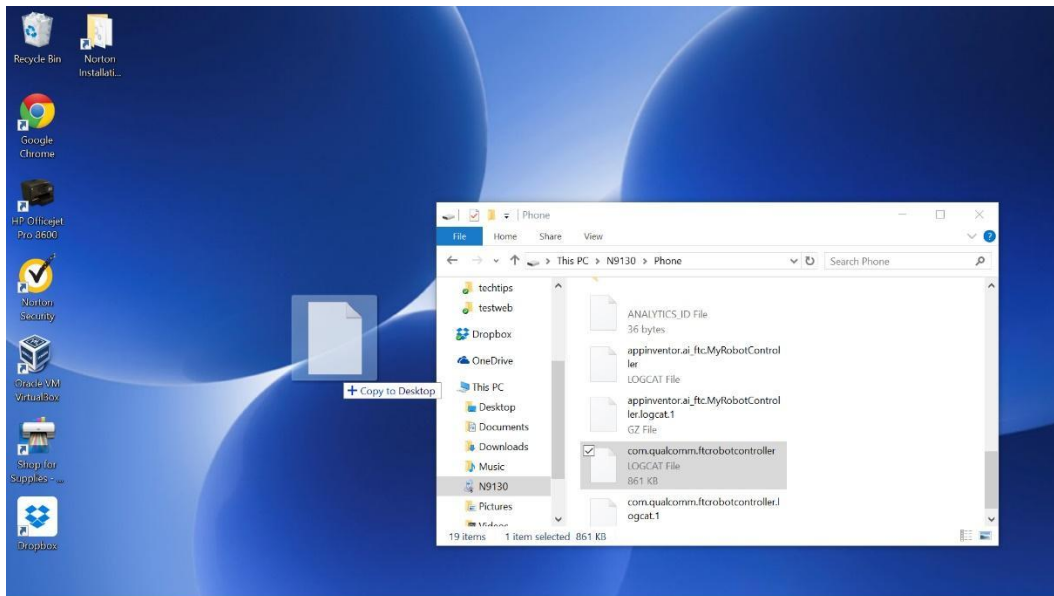


Figure 62 - You can make a copy of a log file by dragging and dropping the file to your personal computer.

Viewing the Contents of the Log File

Once you have successfully copied the log file to your local computer, you can use an application on your computer to open and read the file. The log file is simply a text file and you might be tempted to open the file using an application like Windows Notepad. If you do try and use Notepad, you might find that the formatting of the displayed text is not useful:

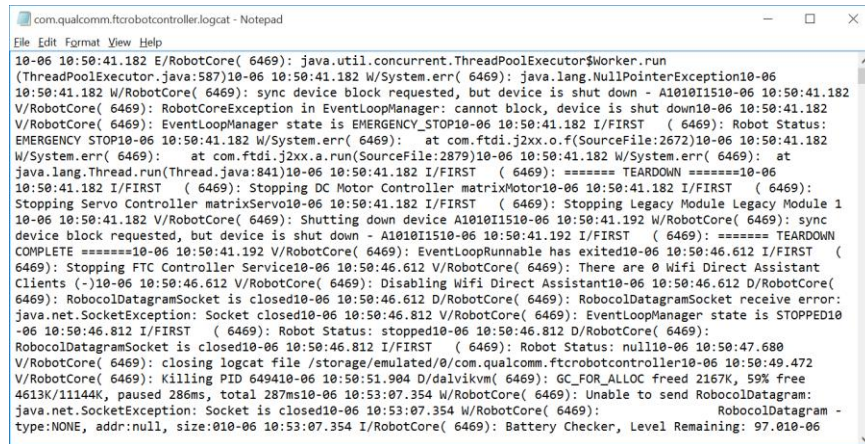


Figure 63 - Using Notepad might not be as useful to browse the log files.

If you are a Windows user, then you can use Microsoft Word to open and view the file. When you attempt to find the file, you need to make sure that the file type filter in the Open file dialog box is set to “All Files” when you try to find your log file.

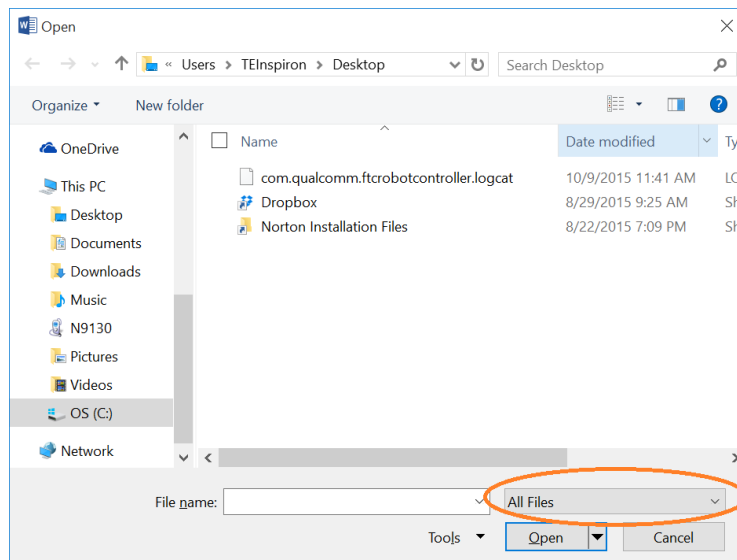


Figure 64 - Make sure the file filter is set to “All Files” when you browse to find your log file.

Using Microsoft Word to view the log file makes it easier to read and search for text in a log file. A useful alternative is Notepad++.

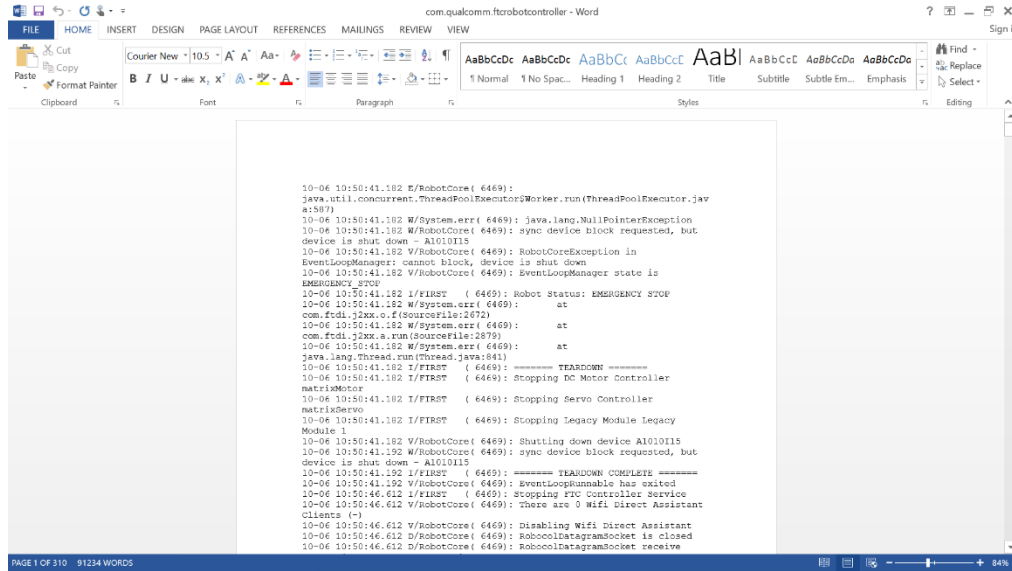


Figure 65 - Using Microsoft Word to view the log file makes it easier to read the contents and search for specific text strings.

Non-Windows Users

If you are a user who has a Mac or Linux computer, you do not have access to the File Explorer application to copy and paste files from the phone. Mac users have the option of use the Android File Transfer program to browse and copy files from the phone:

<https://www.android.com/filetransfer/>

Mac, Linux, and Windows users also have the option of using the Android Debug Bridge utility to transfer files from the phone to their local computer. We will examine the Android Debug Bridge program in the next section of this document.

Using the REV Hardware Client Windows App to View Log Files

A convenient and easy way to troubleshoot problems with the REV Control system is to view log files using the REV Hardware Client for Windows computers. The REV Hardware Client log viewer has filters, tags, and a search function that makes it easy to see what is happening on the Control Hub or Driver Hub during an OpMode run. Instructions for using the REV Hardware Client are available on the REV Robotics website:

<https://docs.revrobotics.com/rev-hardware-client/control-hub/using-the-log-viewer>

Using the Android Debug Bridge for Troubleshooting

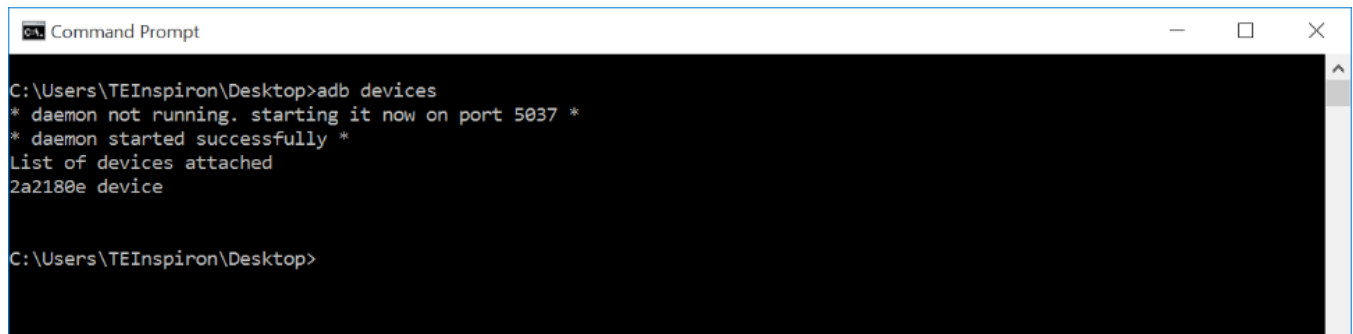
The Android Debug Bridge (ADB) is a utility program that is included with the Android Software Development Kit (SDK) platform tools. ADB is a program that you can invoke from a command line. It is a very helpful utility. To use the ADB utility you will need to have the Android SDK platform-tools installed (preferably a recent version of the Android SDK). Note that normally when you install Android Studio, you also install the Android SDK including the platform tools package.

The examples in this section were made with a Windows PC but the process is similar for Mac and Linux computers. If your computer does not recognize the command “adb” then you should check to make sure that the Android SDK platform-tools are installed in your machine. You should also check that the file path to the adb utility program is included in the command line search path (refer to the appropriate Windows, Mac, or Linux documentation for details on how to check this).

“Shelling” into an Android Device

You can use ADB to “shell” into an Android device. This means that you can use ADB to establish a terminal session with an Android device. The ADB shell provides a command line interface that you can use to type commands to interact with the phone.

To launch an ADB shell, you need to first make sure that USB debugging is enabled for your Android phone and that you have the appropriate driver installed on your computer. Connect the phone to the computer with a USB cable. Open a command line window or a terminal window on your computer and type in “adb devices” at the prompt. This command will list all available Android devices that are currently connected to your computer:

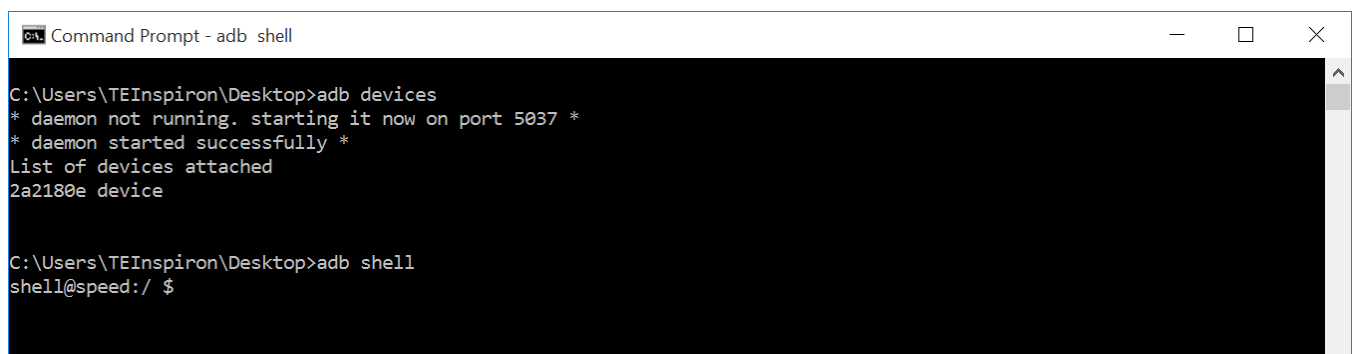
A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the command "adb devices" being entered at the prompt "C:\Users\TEInspiron\Desktop>". The output of the command is displayed below the prompt: "* daemon not running. starting it now on port 5037 *", "* daemon started successfully *", "List of devices attached", and "2a2180e device". The prompt then returns to "C:\Users\TEInspiron\Desktop>".

```
C:\Users\TEInspiron\Desktop>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
2a2180e device

C:\Users\TEInspiron\Desktop>
```

Figure 66 - From a terminal or command line interface type “adb devices” to see a listing of attached Android devices.

If you want to establish a terminal or shell session with your Android phone, simply type “adb shell” at the command prompt:

A screenshot of a Windows Command Prompt window titled "Command Prompt - adb shell". The window shows the command "adb devices" being entered at the prompt "C:\Users\TEInspiron\Desktop>". The output of the command is displayed below the prompt: "* daemon not running. starting it now on port 5037 *", "* daemon started successfully *", "List of devices attached", and "2a2180e device". The prompt then returns to "C:\Users\TEInspiron\Desktop>". Below this, the command "adb shell" is entered, and the prompt changes to "shell@speed:/ \$".

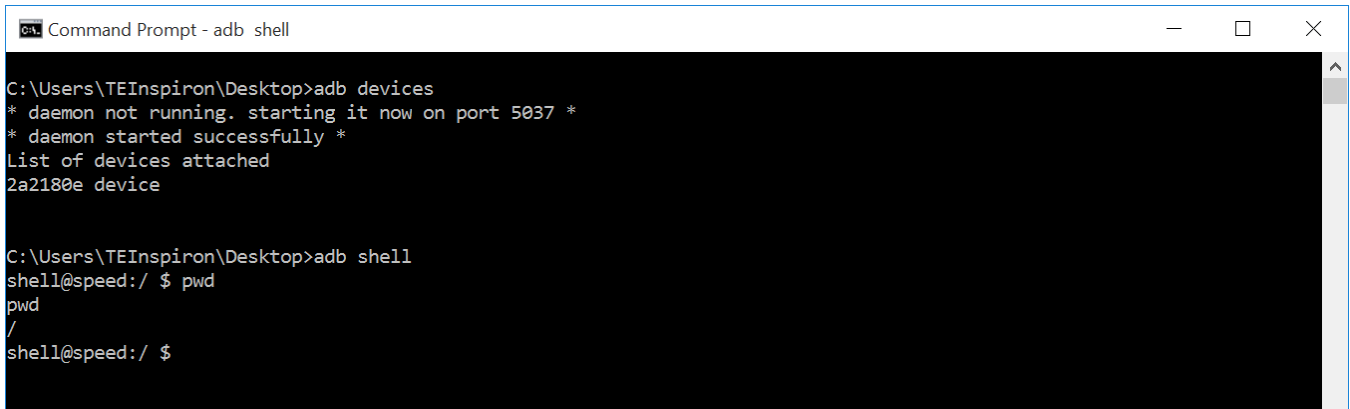
```
C:\Users\TEInspiron\Desktop>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
2a2180e device

C:\Users\TEInspiron\Desktop>adb shell
shell@speed:/ $
```

Figure 67 - Type in “adb shell” to create a terminal session with your Android device.

If you look at Figure 67 you see that the command line prompt changes to “shell@speed:/ \$” after the words “adb shell” were entered. This new command line prompt indicates that the user is now connected to the phone and any commands that are entered will be processed by the phone. Note that Linux commands are case sensitive.

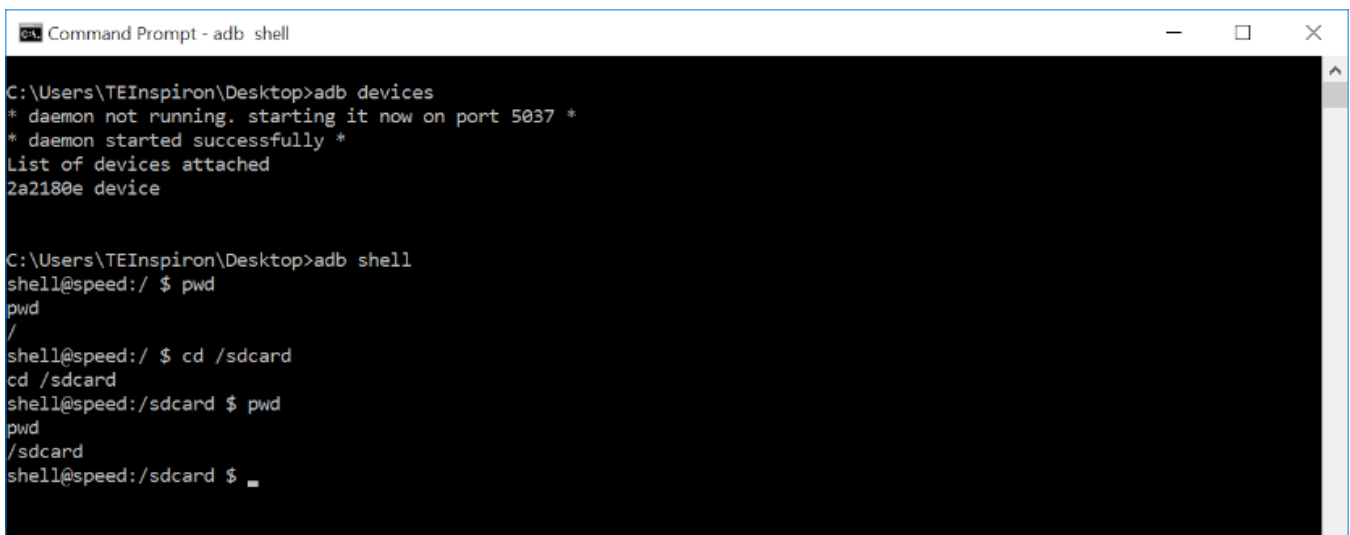
You can use standard Linux commands to navigate the environment. If you type “pwd” at the shell prompt, the phone will print the current directory on the screen:

A screenshot of a Windows Command Prompt window titled "Command Prompt - adb shell". The window shows the following text:
C:\Users\TEInspiron\Desktop>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
2a2180e device

C:\Users\TEInspiron\Desktop>adb shell
shell@speed:/ \$ pwd
pwd
/
shell@speed:/ \$

Figure 68 - The command “pwd” prints the current working directory.

If you type “cd /sdcard” the phone will change the current directory to the /sdcard subdirectory on its file system. If you then type in “pwd” (after you have changed directories) you will see your new location within the file system:

A screenshot of a Windows Command Prompt window titled "Command Prompt - adb shell". The window shows the following text:
C:\Users\TEInspiron\Desktop>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
2a2180e device

C:\Users\TEInspiron\Desktop>adb shell
shell@speed:/ \$ pwd
pwd
/
shell@speed:/ \$ cd /sdcard
cd /sdcard
shell@speed:/sdcard \$ pwd
pwd
/sdcard
shell@speed:/sdcard \$

Figure 69 - Entering in “cd /sdcard” will change the working directory. Entering in “pwd” will print the new working directory.

If you type “ls” at the command prompt the phone will list the contents of the current directory. If you look at the directories and files, you will see that they match the folders and files that you saw using the Android device’s File Manager app, or that you would see using the Windows File Explorer application:


```
shell@speed:/sdcard $ ls
ls
Alarms
Android
DCIM
Download
FIRST
Movies
Music
Notifications
Pictures
Podcasts
Ringtones
TouchPalv5
appinventor.ai_ftc.MyRobotController.logcat
appinventor.ai_ftc.MyRobotController.logcat.1.gz
com.qualcomm.ftcrobotcontroller.logcat
com.qualcomm.ftcrobotcontroller.logcat.1.gz
gallery3d
smvvm
shell@speed:/sdcard $
```

Figure 70 - The command “ls” will list the folders and files in the current directory.

To exit the ADB shell and return to your personal computer’s command prompt simply type in the command “exit”:

```
Notifications
Pictures
Podcasts
Ringtones
TouchPalv5
appinventor.ai_ftc.MyRobotController.logcat
appinventor.ai_ftc.MyRobotController.logcat.1.gz
com.qualcomm.ftcrobotcontroller.logcat
com.qualcomm.ftcrobotcontroller.logcat.1.gz
gallery3d
smvvm
shell@speed:/sdcard $ exit
exit
C:\Users\TEInspiron\Desktop>
```

Figure 71 - The command “exit” will exit you from the phone’s shell and return you to your computer’s command line.

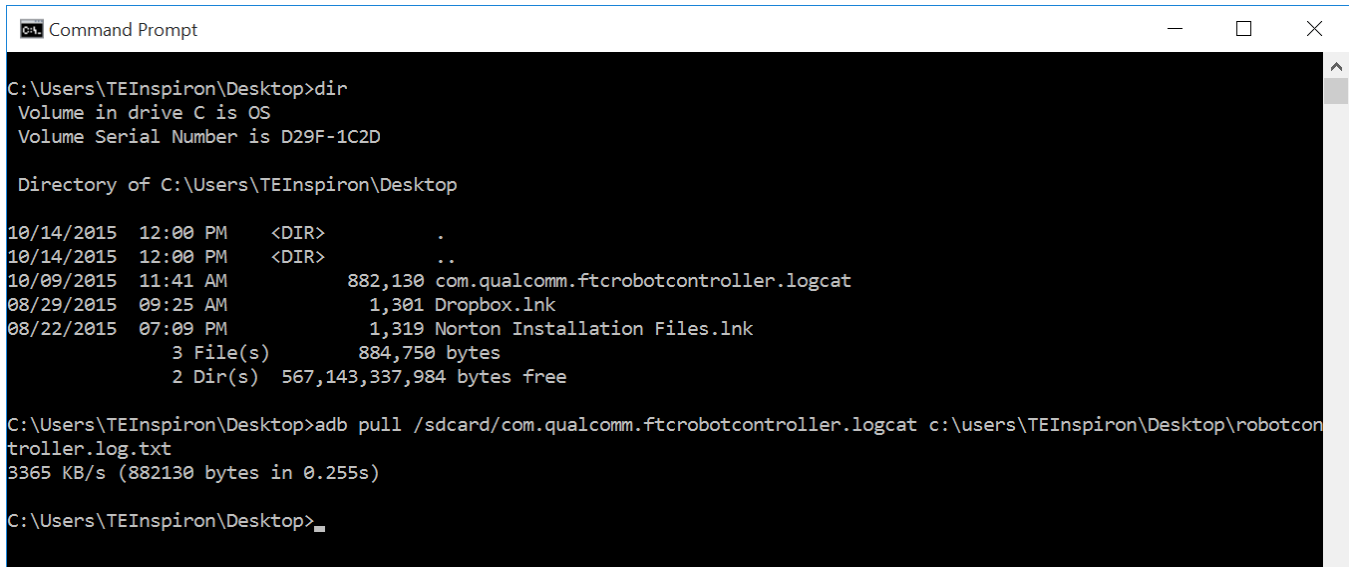
Pulling a File from the Android Device

You can also use the ADB utility program to *pull* a file from the phone to the local file system of your computer. The syntax is “adb pull <SOURCE PATH> <DESTINATION PATH>” where “<SOURCEPATH>” is where on the phone the original file is located and “<DESTINATION PATH>” is where on the computer you want to copy the file to.

For example, if you type in the following command at a user prompt,

```
adb pull /sdcard/com.qualcomm.ftcRobotcontroller.logcat c:\users\TEInspiron\Desktop\rc_log.txt
```

then the ADB utility will attempt to copy the log file from the Android device and to the local file system (c:\users\TEInspiron\Desktop\rc_log.txt).

A screenshot of a Windows Command Prompt window. The title bar says "Command Prompt". The command prompt shows the directory of C:\Users\TEInspiron\Desktop, listing files like com.qualcomm.ftcrobotcontroller.logcat, Dropbox.lnk, and Norton Installation Files.lnk. It then shows the execution of the command 'adb pull /sdcard/com.qualcomm.ftcrobotcontroller.logcat c:\users\TEInspiron\Desktop\robotcontroller.log.txt', which successfully copies the file at 3365 KB/s.

```
C:\Users\TEInspiron\Desktop>dir
Volume in drive C is OS
Volume Serial Number is D29F-1C2D

Directory of C:\Users\TEInspiron\Desktop

10/14/2015  12:00 PM    <DIR>          .
10/14/2015  12:00 PM    <DIR>          ..
10/09/2015  11:41 AM      882,130  com.qualcomm.ftcrobotcontroller.logcat
08/29/2015  09:25 AM       1,301  Dropbox.lnk
08/22/2015  07:09 PM       1,319  Norton Installation Files.lnk
               3 File(s)      884,750 bytes
               2 Dir(s)  567,143,337,984 bytes free

C:\Users\TEInspiron\Desktop>adb pull /sdcard/com.qualcomm.ftcrobotcontroller.logcat c:\users\TEInspiron\Desktop\robotcon
troller.log.txt
3365 KB/s (882130 bytes in 0.255s)

C:\Users\TEInspiron\Desktop>
```

Figure 72 - You can use the “adb pull” command to copy a file from the phone onto your local hard drive.

Once you have copied the file to your local hard drive, you can use an appropriate application to view the log file.

Using Android Studio to View Log Messages

You can also use Android studio to view log messages from your phone. If your phone is connected either through a USB cable or via wireless ADB,⁷ you can view the log messages using the Android Monitor window within Android Studio. Detailed instructions on how to access the Android Monitor feature are available on the Android Developer website:

- <https://developer.android.com/tools/debugging/debugging-studio.html>

Note that the amount of log statements that appear in the window can be overwhelming. It is possible to create filters to show only a subset of data in the logcat window of Android Studio (refer to the Android Developer website for details on how to do this).

⁷ Refer to <http://developer.android.com/tools/help/adb.html#wireless> to see information about how to use ADB wirelessly.

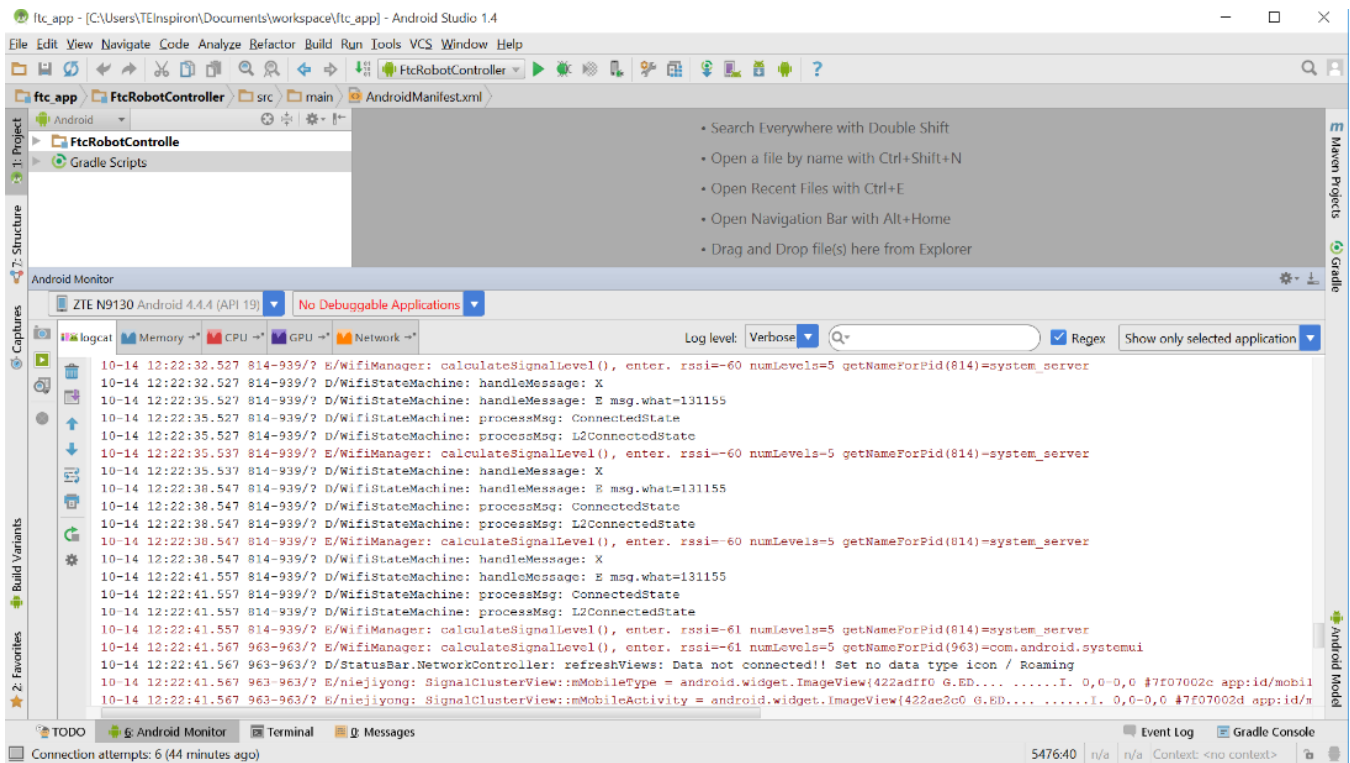


Figure 73 - You can view (and filter) logcat statements through Android Studio.

Creating Your Own Log Statements within an OpMode

It is possible (and often helpful) to insert your own log statements within an OpMode for debug purposes. The *FIRST* Tech Challenge SDK contains a class called `DbgLog` that has two static methods that can be used to log messages to the log file:

```
DbgLog.err(String message)
DbgLog.msg(String message)
```

These two messages can be used to create log statements in your log file. The `err` method will create an error message (which has a different level of severity and can be used to filter statements when viewing logcat output) and the `msg` method will create ordinary messages in the log file.

You can embed these methods within your OpMode and use them to debug the statements in real time using the Android Monitor. You can also look for your statements in the log file.

Example OpMode

The following text is an example OpMode that shows how to use the `RobotLog` class to embed log statements within your OpMode. Note that the `DbgLog` class was removed in version 3.3 of the FTC software (not deprecated, but removed). The `RobotLog.d()` method will print equivalent debug messages to the log file of the Robot Controller:

```
package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.eventloop.opmode.Autonomous;
import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
import com.qualcomm.robotcore.util.RobotLog;

/**
 * Created by tom on 10/3/17.
 */
```

```
@Autonomous
public class MyLogDemo extends LinearOpMode {

    @Override
    public void runOpMode() throws InterruptedException {
        RobotLog.d("TIE - entered runOpMode()");

        double dStart = getRuntime();
        double dCurrent, dElapsed = 0;

        RobotLog.d(String.format("TIE - dStart = %.03f", dStart));
        RobotLog.d("TIE - about to wait for start...");

        waitForStart();

        while (opModeIsActive()) {
            dCurrent = getRuntime();
            dElapsed = dCurrent - dStart;
            telemetry.addData("1. elapse", String.format("%.03f", dElapsed));
            RobotLog.d(String.format("TIE - dElapsed = %.03f", dElapsed));
            this.sleep(250);
        }
    }
}
```

This linear OpMode example shows how to use the *RobotLog.d* method to log information in the log file. You can use the Android Monitor window of the Android Studio IDE to view these log messages in real time. You can also create a filter so you only see a subset of log messages in the window.

Creating a logcat Filter in Android Studio

It is often desirable to filter out unwanted logcat statements when you are debugging. In the example OpMode listed in the section above, the log statements have the expression “TIE” in them (the author’s initials). You can create a filter that will display only log statements that contain this string.

In the right-hand side of the Android Monitor window use the drop-down selector to select **Edit Filter Configuration** to create a new filter (see Figure 74). The Create New Logcat Filter window should appear (see Figure 75). In the Create New Logcat window you can specify a new name for your filter (for example “TIE Filter”). You can also specify a regular expression.⁸ that is used to filter statements. In Figure 75, the expression “TIE” is used as a filter for the log message. This means that the Android Monitor window will only display log statements that include the expression “TIE” in the body of the message.

⁸ Visit https://en.wikipedia.org/wiki/Regular_expression for more information about regular expressions.

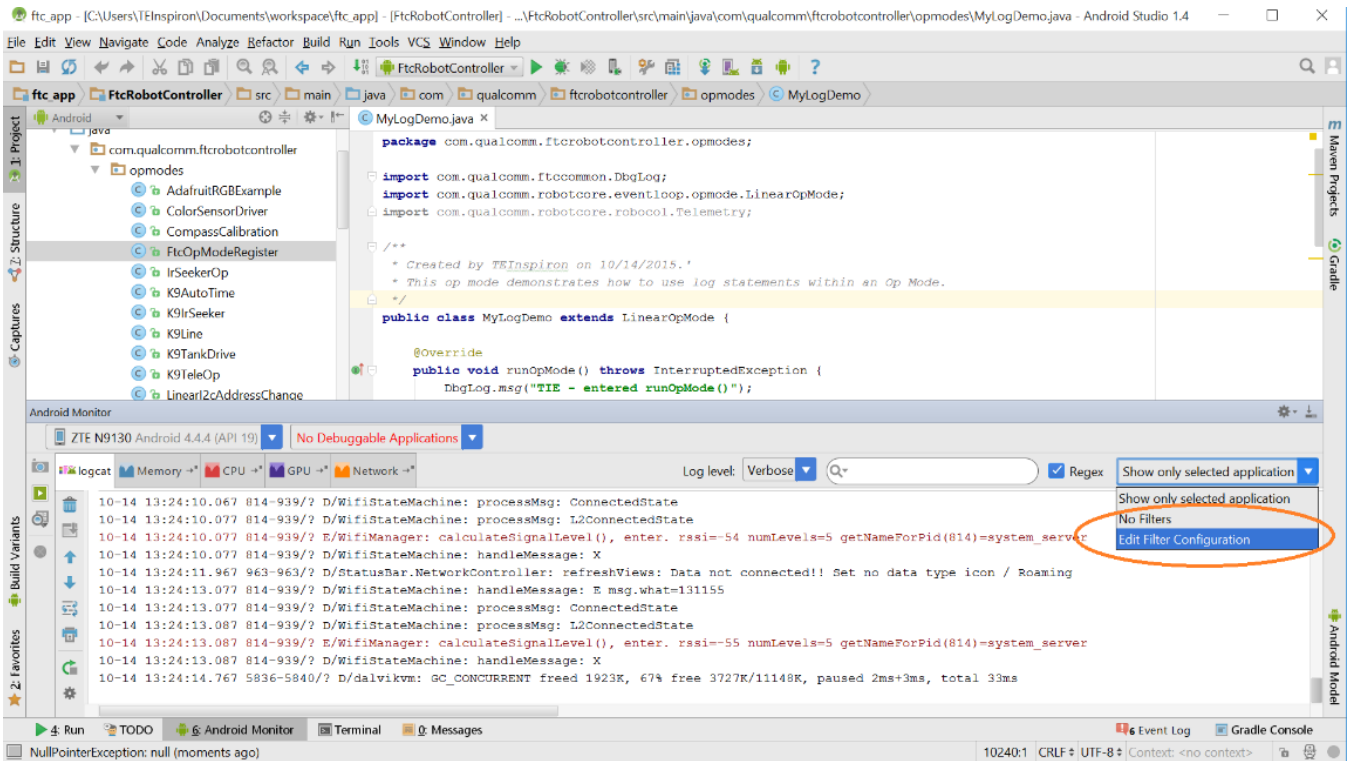


Figure 74 - Select Edit Filter Configuration to create a new filter

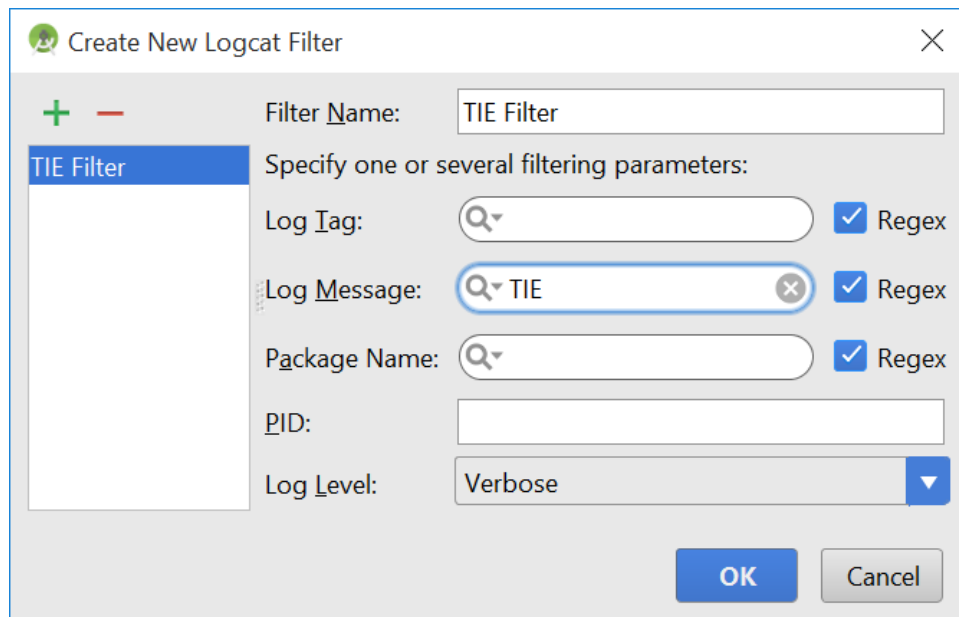


Figure 75 - Specify the Filter Name and a regular expression that you want to use for your filter (in this case "TIE").

Once you have created your filter, the Android Monitor window should automatically filter out messages that do not match the search criteria. You should see the filter statements in the window. If your OpMode is currently running, you can see them in real time.

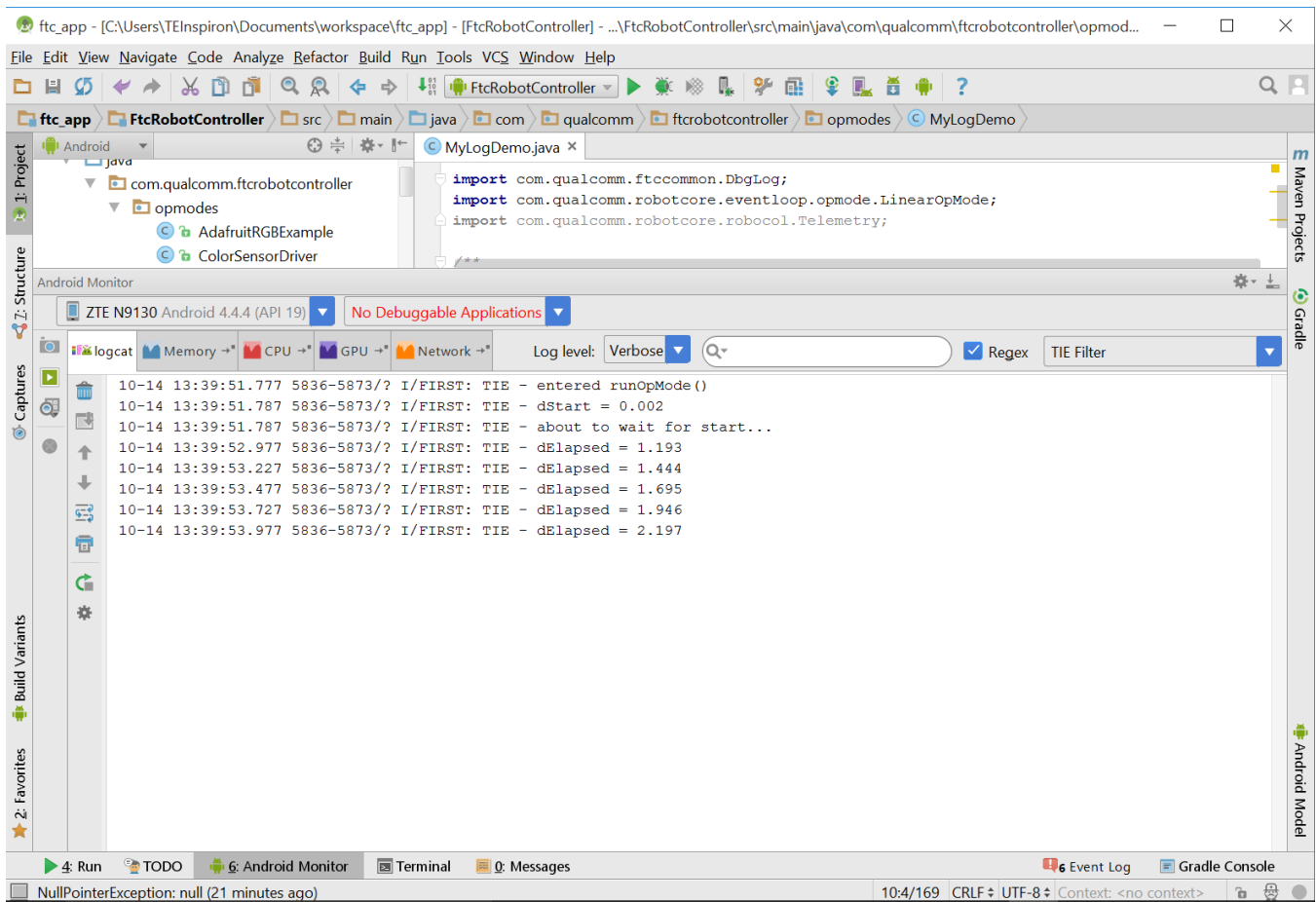


Figure 76 - If the OpMode is running you can see your log statements in real time with a USB or wireless ADB connection.