# 2015-2016 *FIRST*® Tech Challenge
# Control System Troubleshooting Guide





## Volunteer Thank You

Thank you for taking the time to volunteer for a *FIRST* Tech Challenge Event. *FIRST* and *FIRST* Tech Challenge rely heavily on Volunteers to ensure Events run smoothly and are a fun experience for Teams and their families, which could not happen without people like you. With over 4,500 Teams competing annually, your dedication and commitment are paramount to the success of each Event and the *FIRST* Tech Challenge program. Thank you for your time and effort in supporting the mission of *FIRST*!


## Sponsor Thank You

Thank you to our generous sponsors for your continued support of the *FIRST* Tech Challenge!

***FIRST*® Tech Challenge Official Program Sponsor**

Rockwell Collins

***FIRST*® Tech Challenge Official IoT, CAD and Collaboration Software Sponsor**

PTC®

***FIRST*® Tech Challenge Official Control System Sponsor**

QUALCOMM®

| Revision History | | |
|---|---|---|
| **Revision** | **Date** | **Description** |
| 1 | 10/10/2015 | Initial Release |
| 2 | 11/20/2015 | Updated Sections on WiFi Direct troubleshooting. Also added better display filters to be able to use Wireshark more effectively to look for malformed packets. |
| 3 | 11/21/2015 | Links updated to www.firstinspires.org and branding updates |

# Contents

# Introduction

## *What is the FIRST Tech Challenge?*

*FIRST* Tech Challenge is a student-centered activity that focuses on giving students a unique and stimulating experience. Each year, Teams participate in a new Game that requires them to design, build, test, and program autonomous and driver-operated Robots that must perform a series of tasks.

The Playing Field for the Game consists of the *FIRST* Tech Challenge Game Pieces set up on a foam-mat surface, surrounded by a metal and Lexan Field frame. Each Tournament features Alliances, which are comprised of two Teams, competing against one another on the Playing Field. Teams work to overcome obstacles and meet challenges, while learning from and interacting with their peers and adult Mentors. Students develop a greater appreciation of science and technology and how they might use that knowledge to impact the world around them in a positive manner. They also cultivate life skills such as:

***FIRST* Tech Challenge *is More Than Robots!* While competing, students develop personal and professional skills they will be able to rely on throughout their**

- Planning, brainstorming, and creative problem-solving.
- Research and technical skills.
- Collaboration and Teamwork.
- Appreciation of differences and respect for the ideas and contributions of others.

To learn more about *FIRST* Tech Challenge and other *FIRST* Robotics Competitions, visit www.firstinspires.org.

## *FIRST Tech Challenge Core Values*

*FIRST* asks everyone who participates in *FIRST* Tech Challenge to uphold the following values:

- We display Gracious Professionalism with everyone we engage with and in everything we do.
- We act with integrity.
- We have fun.
- We are a welcoming community of students, Mentors, and volunteers.
- What we learn is more important than what we win.
- We respect each other and celebrate our diversity.
- Students and adults work together to find solutions to challenges.
- We honor the spirit of friendly Competition.
- We behave with courtesy and compassion for others at all times.
- We act as ambassadors for *FIRST* and the *FIRST* Tech Challenge.
- We inspire others to adopt these values.

## Gracious Professionalism™

*FIRST* uses this term to describe the program's intent. This is one of the most important concepts that can be taught to a young person who is learning to get along in the work world. At *FIRST*, Team members help other Team members, but they also help other Teams.

Gracious Professionalism is not clearly defined for a reason. It can and should mean different things to everyone.

Some possible meanings of Gracious Professionalism include:

- Gracious attitudes and behaviors are win-win.
- Gracious folks respect others and let that respect show in their actions.
- Professionals possess special knowledge and are trusted by society to use that knowledge responsibly.
- Gracious Professionals make a valued contribution in a manner pleasing to others and to themselves.

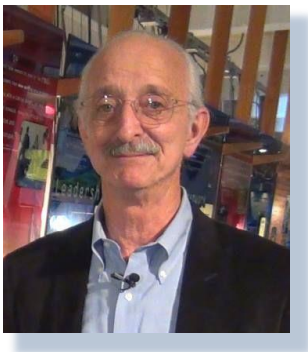
**An example of Gracious Professionalism is patiently listening to a Team's question and providing support despite having several pressing things to do on the day of the Event.**

In the context of *FIRST*, this means that all Teams and participants should:

- Learn to be strong competitors, but also treat one another with respect and kindness in the process.
- Avoid leaving anyone feeling as if they are excluded or unappreciated.
- Knowledge, pride and empathy should be comfortably and genuinely blended.

In the end, Gracious Professionalism is part of pursuing a meaningful life. When professionals use knowledge in a gracious manner and individuals act with integrity and sensitivity, everyone wins, and society benefits.

Watch Dr. Woodie Flowers explain Gracious Professionalism in this short video.



*"The FIRST spirit encourages doing high-quality, well-informed work in a manner that leaves everyone feeling valued. Gracious Professionalism seems to be a good descriptor for part of the ethos of FIRST. It is part of what makes FIRST different and wonderful."*

- Dr. Woodie Flowers, National Advisor for ***FIRST***

## Youth Protection Program

The purpose of the *FIRST* Youth Protection Program (*FIRST* YPP) is to provide coaches, mentors, volunteers, employees, others working in *FIRST* programs, Team members, parents, and guardians of Team members with information, guidelines, and procedures to create safe environments for everyone participating in *FIRST* programs.

The *FIRST* YPP sets minimum standards recommended for all *FIRST* activities. Adults working in *FIRST* programs must be knowledgeable of the standards set by the *FIRST* YPP, as well as those set by the school or organization hosting their Team.

### *Youth Protection Expectations and Guidelines*

Coaches and Mentors are expected to read and follow elements in the *FIRST* Youth Protection Program guide that are labeled as required. These are mandatory in the United States and Canada, and may not be waived without the approval of the *FIRST* Youth Protection Department.

*FIRST* recommends that the standards set forth in the *FIRST* Youth Protection Program guide be applied outside of the United States and Canada to the extent possible. At a minimum, local regulations regarding youth protection must be complied with.

**Everyone working with *FIRST* Teams should be familiar with the *FIRST* YPP policies.**

Forms are available here: http://www.firstinspires.org/resource-library/youth-protection-policy

Information on the US Screening process is available here: http://www.firstinspires.org/sites/default/files/uploads/about/US-Screening-Screen-Shots.pdf

Information on the Canadian Screening process is available here: http://vimeo.com/30137373

You can find FAQ and additional information about the *FIRST* Youth Protection Program on the *FIRST* website at: http://www.firstinspires.org/resource-library/youth-protection-policy

### *NOTICE OF NON-DISCRIMINATION*

United States Foundation for Inspiration and Recognition of Science and Technology (*FIRST*®) does not discriminate on the basis of race, color, national origin, sex, disability, or age in its programs and activities. The following person has been designated to handle inquiries regarding the non-discrimination policies: Lee Doucette, Youth Protection Program Manager, 200 Bedford Street, Manchester, NH 03101, 603-666-3906, Ext. 250.

# 1.0 Introduction

## *1.1 Android-based Platform*

For the 2015-2016 season, the *FIRST* Tech Challenge adopted an Android-based Control System for its Robot Competition. This document provides tips and recommended procedures for avoiding potential problems with the new Android-based Control System. It also provides information to help troubleshoot and resolve common problems with the system.

## *1.2 Point-to-Point Control*

The new 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.**

This point-to-point control model is different from the previous Samantha Field Control System (FCS). The older Samantha FCS uses a centralized control model. With the older Samantha FCS model, each Robot would connect to a single wireless router on the Field. Driver input and feedback from the Robot were sent through this wireless router.

With the Samantha FCS, it was the Event Host who was responsible for setting up, maintaining and troubleshooting the FCS laptop, gamepads, USB hubs, and wireless router. Teams were responsible for bringing their Robots and their Samantha radio modules and they would "pair" or connect their Robots to the FCS's wireless network at the Event.

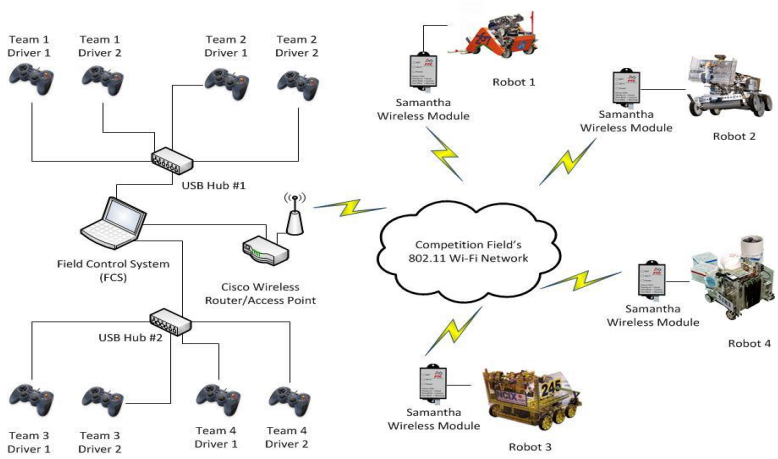



**Figure 2 – The Samantha FCS uses a centralized control model.**

With the new 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.

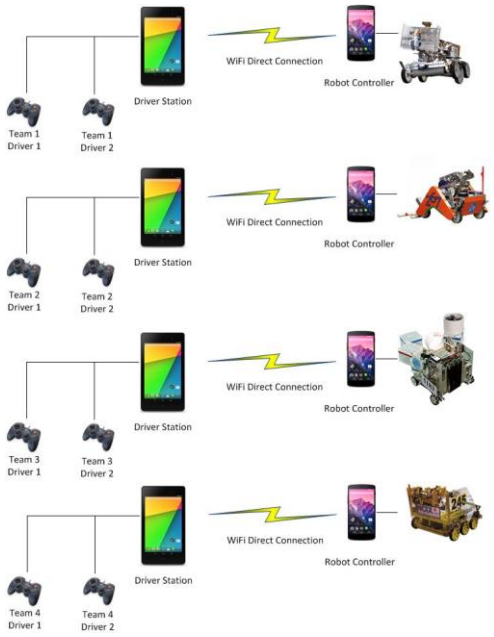



**Figure 3 - Each Team will have its own DS, RC and Wi-Fi Direct network connection.**

### *1.3 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 Field 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 new Control System.

This document provides information on steps that can be performed before and during an *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 was not intended to teach users how to operate the new *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 new Control System, please visit the *FIRST* Tech Challenge Robot Building Resources web page and click on the Intelitek Training Resources link listed under the Android-Based Technology category of the page.

# 2.0 Wi-Fi Direct 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. Wi-Fi Direct is a wireless specification that allows Wi-Fi Direct enabled devices to connect to each other without the need for a wireless access point.[1] The Android operating system supports Wi-Fi Direct with its Wi-Fi Peer-to-Peer (P2P) technology. The *FIRST* Tech Challenge Robot Controller and Driver Station apps use Android's Wi-Fi P2P technology to establish a secure and persistent wireless connection.

## *2.1 Wi-Fi Direct Group Owner*

For a Wi-Fi Direct P2P connection, one of the connecting devices acts like a Wi-Fi access point and is referred to as the *group owner*. The other connecting device acts as connecting client to the group owner. For the *FIRST* Tech Challenge application, the Robot Controller is the device that acts as the group owner for the P2P connection.

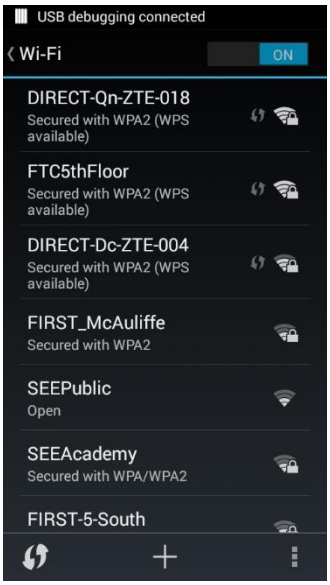
## *2.2 Examining Android's Wi-Fi Direct Configuration Activity*

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 *FIRST* Tech Challenge Driver Station app to pair/unpair your devices.

Even though Teams should pair/unpair their devices through the *FIRST* Tech Challenge Driver Station's activity, it is still 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.

### *2.2.1 Accessing the Wi-Fi Direct Configuration Activity*

To access this screen, on your Android device, launch the **Settings** activity then click on the **Wi-Fi** item to launch the Wi-Fi configuration activity:




[1] See http://www.thinktube.com/tech/android/Wi-Fi-direct (accessed on September 18, 2015).

**Figure 4 - Launch the Wi-Fi activity.**

To access the Wi-Fi Direct menu, touch the three vertical dots in the lower right hand corner of the screen to display a pop-up menu. Select **Wi-Fi Direct** from the pop-up menu to display the Wi-Fi Direct configuration activity. Note that the screenshots in this document were generated using a ZTE Speed phone running Android Kit Kat. The screen images and menu text might vary from device to device.

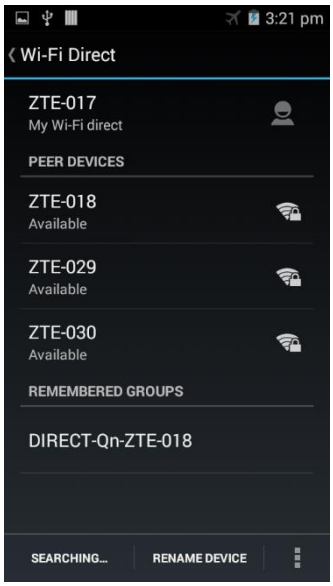



**Figure 5 – Wi-Fi Direct activity.**

Towards the top of the activity, the Wi-Fi name of the Android device (in this example “ZTE-017”) is displayed. Teams can change the name by clicking on the **RENAME DEVICE** button at the bottom of the screen. If this button is not visible at the bottom of the screen, a user can also rename the device by touching the three vertical dots in the lower right hand side and selecting the **Rename device** option from the pop-up menu.

Below the device’s name, there is a list of available Wi-Fi Direct devices listed (under the heading **PEER DEVICES**). In this example, we see three devices, “ZTE-018”, “ZTE-029”, and “ZTE-030” listed with status of available.

Below the list of available peer devices, is a list of **REMEMBERED GROUPS**. When you establish a Wi-Fi Direct connection with another Android device, a remembered group will appear for that connection. The next time your device tries to connect to its paired device, it will attempt to use the information in the remembered group to establish the connection.

In Figure 5 there is a single remembered group **DIRECT-Qn-ZTE-018** listed. This is the remembered group that Android device named “ZTE-017” uses to connect to Android device “ZTE-018”.

Sometimes you will see multiple remembered groups listed on this screen. This can occur if you have used your Android device to connect to more than one Wi-Fi direct device previously. For example, suppose the device named “ZTE-017” were to connect to a device called “LEE-RC”. In this case, multiple remembered groups might appear in the Wi-Fi Direct activity (see Figure 6).

Multiple groups might also appear if you paired your device to the same Android device, but on different wireless channels. For example, suppose you had “ZTE-017” paired to “ZTE-018” on channel 1 of the 2.4GHz

band. Then you later switched the operating channel for “ZTE-018” to 6. When you re-pair “ZTE-017” to “ZTE-018”, it will create a new remembered group.

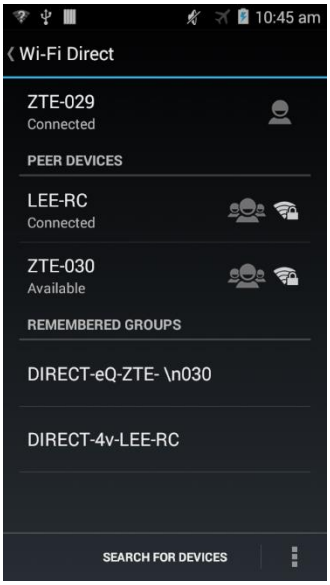



**Figure 6 - Multiple Remembered Groups can be listed.**

Over time, your Android device can accumulate a lot of remembered groups if you have used it to connect to a variety of different devices or under different circumstances (for example, switching channels frequently). It is OK to leave the remembered groups in the list. However, sometimes it is helpful to clean up old and unused remembered groups. To “forget” a remembered group, simply touch and hold the remembered group that you want to forget. The Android device will prompt you if it’s OK to forget the select group. Hit the **OK** button to forget the group.

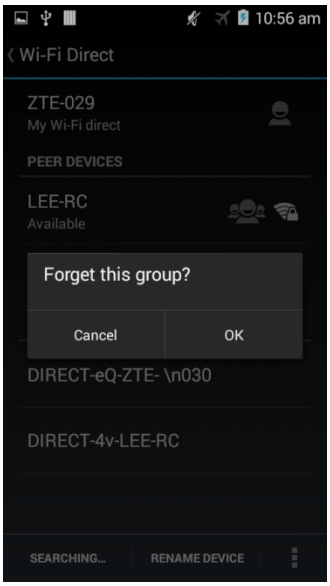



**Figure 7 - Hit OK to forget the group.**

## *2.3 Troubleshooting Wi-Fi Direct Connections*

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

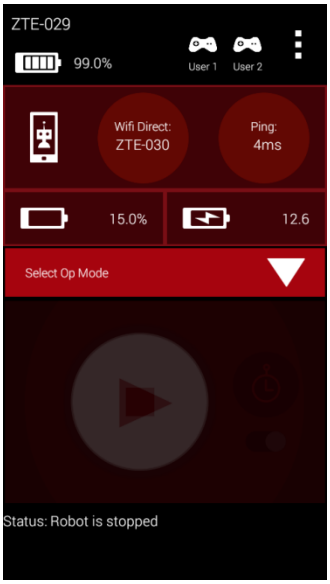



**Figure 8 - When the Driver Station is connected, it displays useful status info.**

When your Driver Station is able to connect to the Robot Controller successfully, it will display useful status information (see Figure 8) on its screen including the name of the device that it is connected to ("ZTE-030" in Figure 8), the average ping times between the Driver Station and Robot Controller and voltage info for the Robot Controller and the main Robot battery.

### *2.3.1 Is the Robot Controller On?*

It sounds fairly obvious, but if you are having problems connecting to your Robot Controller, check the following items,

- Is the Robot Controller device turned on?
- Is the Robot Controller device running the *FIRST* Tech Challenge Robot Controller app?
- Is the *FIRST* Tech Challenge Robot Controller app in the foreground (and NOT minimized)?
- Is the Robot Controller device in Airplane mode with Wi-Fi enabled?

The Robot Controller device must be powered on and have the *FIRST* Tech Challenge Robot Controller app running before the Driver Station can connect to it.

### *2.3.2 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 *FIRST* Tech Challenge Driver Station app and the *FIRST* Tech Challenge 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.

### *2.3.3 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* mechanism[2] to debug their apps.
- Teams might need to connect to a wireless network to download something to their phone from the Internet (like the *FIRST* Tech Challenge apps from the Google Play store).
- 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).

For Competition use, we recommend that the Teams make sure that their Android devices are not connected to any other Wi-Fi or Wi-Fi Direct network. We also recommend that the Teams *forget* any other Wi-Fi or Wi-Fi Direct network, with the exception of primary Wi-Fi Direct connection between the Driver Station and Robot Controller.

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

- 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.

### *2.3.4 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 a large number of devices in the vicinity, the target device that you are trying to connect to ("ZTE-018" 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 together 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.

Paradoxically, 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.

### *2.3.5 Are Android Devices Rebooting Upon FIRST* Tech Challenge *Driver Station Startup or Wi-Fi Direct Scanning?*

If you are at an Event where some or all of the Android devices reboot when the *FIRST* Tech Challenge Driver Station apps are first launched or whenever the devices try to do a Wi-Fi Direct scan, then there might be a

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

device that is generating *malformed probe response* packets in the vicinity. These wireless packets with *malformed* data can cause an Android device running the Kit Kat version of the operating system to reboot whenever the device tries to do a Wi-Fi Direct scan.

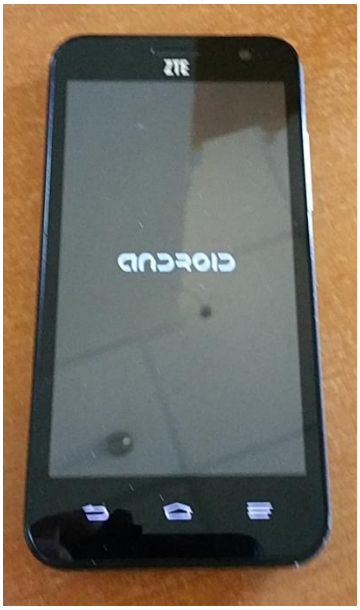



**Figure 9 - Does your Android device reboot when you launch the *FIRST* Tech Challenge app or if you try to do a Wi-Fi Direct Scan?**

There is a bug in Android Kit Kat in which a probe response that contains *malformed* device information can cause the Android device to reboot.

An organization known as Core Security first reported this problem. Details of this problem can be found at the following URL: http://www.coresecurity.com/advisories/android-Wi-Fi-direct-denial-service

What the security report demonstrates is that Kit Kat devices that have this vulnerability will reboot if they encounter a certain type of wireless packet that contains improperly formatted data. One of the most common causes of this problem that we have seen in the field is an Android device that is configured with certain non-alphanumeric characters contained within its device name. For example, if a team or a spectator has an Android device that has a newline character ('\n' or 0x0a in ASCII) in the Wi-Fi Direct name, this device can cause Android Kit Kat devices to reboot whenever these Kit Kat devices attempt to do a Wi-Fi Direct scan.

If you are at an Event and you see Kit Kat Android devices (such as the ZTE Speed) rebooting whenever you first launch the *FIRST* Tech Challenge apps or when you try to search for other Wi-Fi Direct devices, then there might be a device that is inadvertently or intentionally sending out probe responses with malformed data (for example, a newline embedded in the data).

This vulnerability is associated with Android Kit Kat and is supposed to be addressed in later versions of the Android operating system. Unfortunately, the ZTE Speed phones that are used by *FIRST* Tech Challenge are Kit Kat devices and susceptible to this vulnerability.

Here are some things you can do to try and address the problem:

***Try to Find Devices with Non-Alphanumeric Characters in Their Wi-Fi Direct Names***

Robot rule <RS02> specifies that teams must name their Robot Controller with their official *FIRST* Tech Challenge Team number and a –RC (e.g., "1234-RC") and teams must name their Driver Station with their official *FIRST* Tech Challenge Team number and –DS (e.g., "1234-DS"). Spare Android devices should be named with the Team number followed by a hyphen then a letter designation beginning with "B" (e.g., "1234-B-RC", "1234-C-RC").

If a team inadvertently or intentionally adds additional, non-alphanumeric characters to their Wi-Fi Direct Name this device can cause Driver Stations (running Android Kit Kat) to crash and reboot when the device attempts to scan for the presence of other devices. One of the most common causes that we have seen is an embedded newline character in the Wi-Fi Direct name. Since the touch screens on the Android devices are relatively small, users often accidentally hit the RETURN key when typing in the Wi-Fi Direct network name of the device. If an Android device has a newline in its network name, then it could inadvertently cause problems due to this vulnerability.

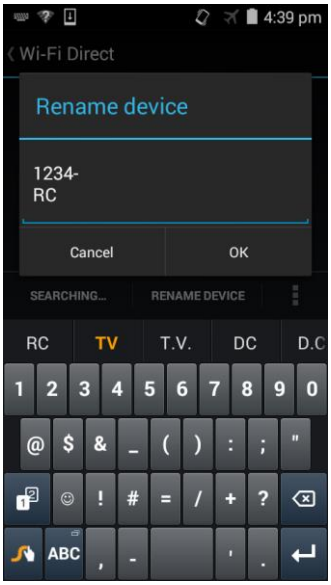



**Figure 10 - A newline character can cause devices with this vulnerability to reboot on Wi-Fi Direct scan.**

If your Android Kit Kat devices are rebooting whenever teams launch the *FIRST* Tech Challenge Driver Station or any other time a user attempts to do a Wi-Fi Direct scan, please check with the names of the Android devices at your Event and see if you can find one with a newline character or other non-alphanumeric character embedded in it.

If you have an Android device that is not vulnerable to this reboot bug (such as an updated device with a patch for this vulnerability) you can do a Wi-Fi Direct scan and look for any Wi-Fi Direct names that are displayed with an underscore character ('_') or any other unusual (non alphanumeric) character in it (see Figure 11).

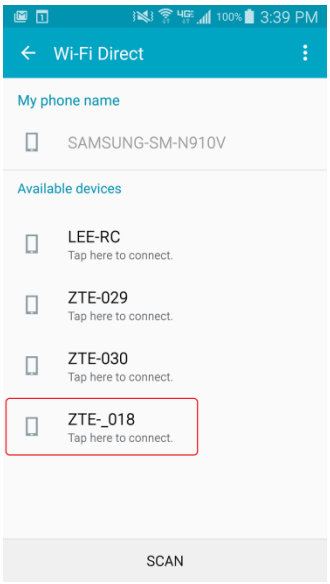



**Figure 11 - An embedded newline character would look like an underscore character '_'.**

It is possible that there is a device that is in the vicinity that contains malformed data in a different Field, which might make it difficult to identify the device. There are techniques available to detect the presence of these problematic packets (see the section entitled 7.2 Using a Display Filter to Look for Malformed Probe Responses of this document). These techniques, however, require access to some sophisticated tools (such as Wireshark running on a computer that has a wireless adapter that can run in monitor mode) which might not be available at every *FIRST* Tech Challenge Event.

***Pair Devices in Isolation***

If you are at an Event and you are unable to find the offending device that is causing problems with the Wi-Fi Direct discovery, you have the option of pairing the Team's devices in isolation. Similar to the problem described in section entitled 7.2 Using a Display Filter to Look for Malformed Probe Responses of this manual, for this scenario you can remove Android devices from the main group and pair them in isolation. Once the devices are paired, they will not do a Wi-Fi Direct scan and are no longer susceptible to the vulnerability. This vulnerability should not affect a Driver Station-Robot Controller pair once they have established a secure connection unless you power cycle the phones or if you swipe close or kill the *FIRST* Tech Challenge Driver Station app, in which case one of the phones will scan when you first launch the *FIRST* Tech Challenge Driver Station app.

FTAs have reported that at some early events where there was a phone broadcasting bad data causing this problem, teams left the venue with their phones (and went down the hallway or outside the building) and paired their phones away from the other devices and then went back in to the venue afterwards.

***Pair the Devices in Small Groups***

If you think the problem is caused by a misconfigured Android phone, then you can request that Teams turn off all of their devices, and then turn on and pair the phones in small groups. Once a set of phones has been paired, they should no longer be vulnerable to this bug (unless you power cycle the phones or if you swipe close or kill the *FIRST* Tech Challenge Driver Station app, then one of the phones will scan when you first launch the *FIRST* Tech Challenge Driver Station app). Pairing in groups can help you identify which phone is the one causing problems.

# 3.0 Monitoring and Troubleshooting the Wireless Environment

The new *FIRST* Tech Challenge Control System uses Wi-Fi Direct technology to connect the Driver Station devices to the Robot Controllers. Wi-Fi Direct networks can be managed like normal Wi-Fi networks. The techniques and tools that you might use to monitor and troubleshoot the Samantha Field Control System or a corporate Wi-Fi network can be applied to the Wi-Fi Direct networks used by the new Control System. This chapter provides some basic information to help the FTA/CSA/WTA keep the wireless environment clean and operational at an Event.

## *3.1 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. Currently, most Wi-Fi enabled devices have the ability to operate at a base frequency of 2.4GHz. Newer (and often more expensive) devices also often have the ability to operate at a base frequency of 5GHz.

The current *FIRST* Tech Challenge-approved Android devices for this season only operate at the 2.4GHz frequency. We hope to offer Android devices that have built in 5GHz radios, but for now, in an effort to minimize the cost of the devices, the approved Android devices only have 2.4GHz enabled radios.

In the U.S., there is a band of 11 channels (1 through 11 in Figure 12 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.

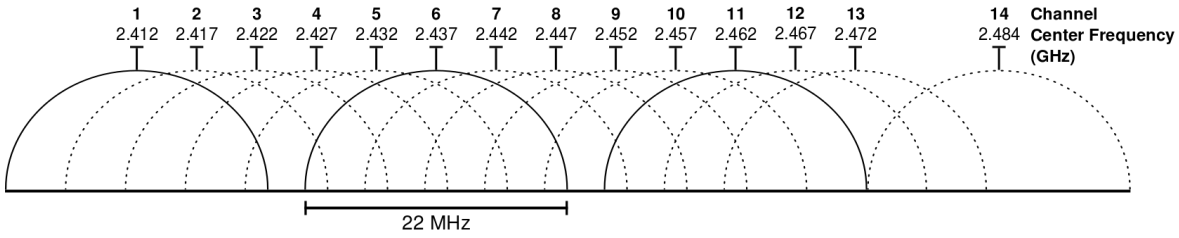



**Figure 12 – In the U.S., there are 11 Wi-Fi channels in the 2.4GHz band. Other countries allow a few additional channels.**[3]

Although there are 11 (or more) operating channels that are available to use, you can see in Figure 12 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 won't 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.

[3] 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) on 9/20/15.

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 *FIRST* Tech Challenge Driver Station and *FIRST* Tech Challenge 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 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.

## *3.2 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.

### *3.2.1 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

Unfortunately, this app is not available on Apple's iOS platform. It will only run on Android. We have not yet identified a similar app that is available for iOS devices.

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 13 - Wi-Fi Analyzer screen shot.[4]**

An FTA, CSA, WTA or Event Host 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 Event Host 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

[4] Image taken from the Google Play listing for Wi-Fi Analyzer.

of other wireless networks on that channel (assuming there aren't 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 an Event Host 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 *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 setups 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.

### *3.2.2 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. In order 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/sbin 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'll need to use super user status and provide your account's password in order to create the link): *sudo ln -s /System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport /usr/sbin/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 14 – 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 the section entitled "Wi-Fi" section 0 of this document. Also this command line argument does not run continuously. A user has to repeatedly issue the command (or write a script to do so) in order to get a continuous listing of available Wi-Fi networks.

### *3.2.3 Fluke AirCheck™ Wi-Fi Tester*

A company called Fluke makes an expensive, but very powerful wireless network monitoring tool. The Fluke *Aircheck™ Wi-Fi Tester* is a handheld device that can be used to monitor the wireless spectrum at an Event. Details regarding the Aircheck device can be found on the Fluke website: http://www.flukenetworks.com/enterprise-network/network-testing/AirCheck-Wi-Fi-Tester

The Fluke Aircheck is similar, but even more powerful than the previous tools that we've listed. The Aircheck has the ability to display information about any wireless network in the vicinity. Unlike the Wi-Fi Analyzer app and the Apple airport utility, the Fluke meter can also provide information about hidden wireless networks.

**Figure 15 - Fluke Aircheck™ Wi-Fi Tester.**

Unlike the Wi-Fi Analyzer app, the Fluke 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 Fluke monitor can estimate how much of channel's capacity is being consumed. This can help an FTA, or CSA, or WTA determine if a wireless channel is "clean" or "noisy".

It is important to note that the Fluke 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 Fluke Aircheck can be equipped with an external directional antenna. The external antenna can be very helpful in locating the source of a wireless signal. An FTA/CSA/WTA 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 16 – The Aircheck™ can be equipped with an external antenna.**

### *3.2.4 MetaGeek inSSIDer*

A company called MetaGeek makes software and hardware devices 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 can be used to monitor wireless channels and see Wi-Fi and non-Wi-Fi traffic on the wireless spectrum.

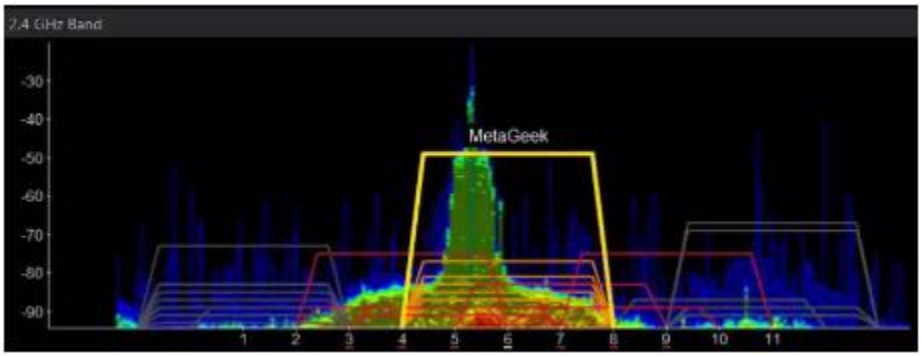



**Figure 17 - MetaGeek's inSSIDer software with the Wi-Spy Mini or DBx hardware shows Wi-Fi & non-Wi-Fi activity.**

Details regarding 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 8, 7 or Vista to operate.

### *3.2.5 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 training manual. Section 7

7.0 Wireshark of this 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. In order to be able to use Wireshark to capture these wireless packets, you need to have a specially equipped computer. In order 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* of 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.

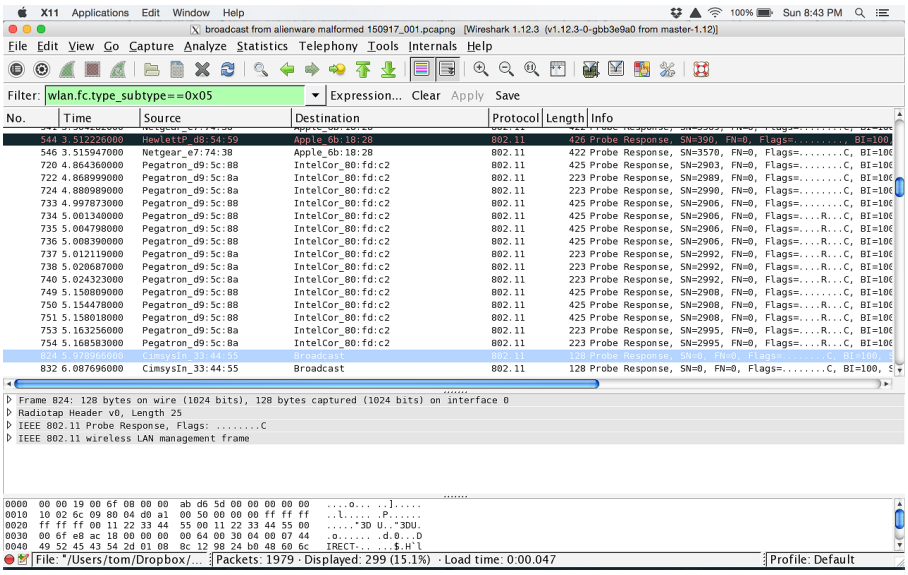



**Figure 18 – Wireshark is a powerful tool, but it requires special knowledge and a Wi-Fi adapter that supports monitor mode.**

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 estimate 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 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).

### *3.3 Troubleshooting the Wireless Environment at an Event*

If you are at an Event and you suspect that there might be wireless causing problems at the Event, there are some things that you can look at to try and diagnose the problem and see if it really is a wireless issue.

#### *3.3.1 Ping Times*

If you are at an *FIRST* Tech Challenge Event, you can use the ping time feature of the *FIRST* Tech Challenge 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 19 - 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 *FIRST* Tech Challenge Driver Station app (see Figure 20).

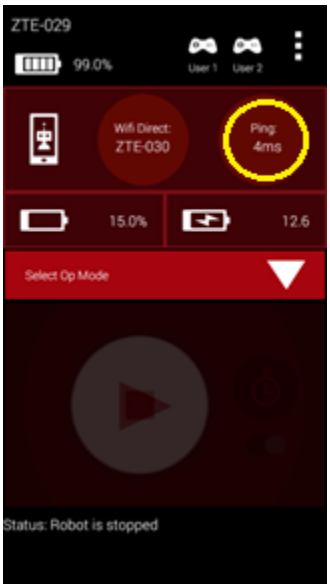


**Figure 20 - The average ping time is displayed in the upper right hand corner (highlighted in yellow in this image).**

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 FTA, CSA, or WTA should have access to a pair of Android devices (preferably the ZTE Speeds, because these devices allow you to change the channel manually) that he/she 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, or 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 determining if a wireless channel is clear and relatively noise-free. If the ping times are low, then the channel is mostly likely free of other Wi-Fi and non-Wi-Fi traffic.

### *3.3.2 Is the Wi-Fi Channel Too Busy?*

In addition to ping times, an FTA, CSA, or WTA can use other tools, such as the Fluke 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 Fluke Aircheck meter, then you can examine the activity level for each wireless channel and determine if a channel is being saturated.

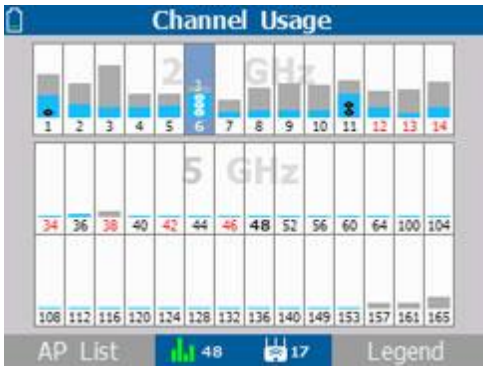



**Figure 21 - The Fluke Aircheck meter shows Wi-Fi (light blue) & non-Wi-Fi (gray) activity on each channel.[5]**

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

In Figure 21 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 Gameboys, 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.

### *3.3.3 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 were able to 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 channels should be spaced at least 5 channel-widths apart, to avoid any overlap. However, if necessary, you can move Robots to overlapping channels.

### *3.3.4 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 are allowed to 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,[6] 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."[7]

### *3.3.5 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.

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 are located near the back pane of the device).

**Figure 22 - Metal music stands like this one can block, reflect or attenuate the signals to/from the Driver Station.**

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 some kind of metal housing, then the signals to and from the Driver Station might be blocked, reflected or attenuated.

As an example, we conducted some experiments using a metallic music stand as a Driver Station stand for a ZTE phone. We used Wireshark to monitor the activity with and without the music stand in place. We 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 wireless connection was worse whenever the music

[6] See https://www.fcc.gov/encyclopedia/jammer-enforcement (accessed on 9/25/15).
[7] See https://apps.fcc.gov/edocs_public/attachmatch/DA-15-113A1.pdf (accessed on 9/25/15).

stand was in place. We attribute 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 doesn't block the radio signals travelling 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.

### *3.3.6 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 *FIRST* Tech Challenge Driver Station-Robot Controller pairs.

There is an amendment to the 802.11 standard (802.11w) that makes it more difficult to conduct many of these types of attacks, but unfortunately the amended standard is not yet available on the Android platform. For now, the Android 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. The section entitled 7.0 Wireshark 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.

# 4.0 Accommodating a Large Number of Robots

## *4.1 Wi-Fi Event Checklist*

The new 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 3). 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 very 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 relatively small. As long as there isn't any other source of interference (such as Bluetooth devices operating on the Field or wireless audio/video systems broadcasting on the same frequency) the new *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 Checklist that contains detailed steps that an Event Host or technical volunteer can take to help keep the wireless environment operating smoothly.

## *4.2 Distributing Robots Across Multiple Channels*

### *4.2.1 Wi-Fi Channel Overlap*

If there is an Event that will have a large number of Robots (> 40 bots) 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 Wi-Fi channels that are less than 5 channel widths apart overlap (see Figure 12).

Ideally, 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 4 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 of the bots on the same channel.

### *4.2.2 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 relatively 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 were able to 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 relatively 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 small 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 *FIRST* Tech Challenge Driver Station and *FIRST* Tech Challenge Robot Controller apps to monitor the ping times on the target channel. You'll need a pair of Android devices that support channel changing (like the ZTE Speed). You should switch to the target channel and test to make sure you can select and run an op mode (like the NullOp sample op mode). If the average ping times for the test Android devices are low (< 5msec) 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 Fluke 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.
3. ***What type of Android devices will the Teams be using?*** Not every Android device permits Wi-Fi Direct channel changing. Currently (as of September 2015) the ZTE Speed phone allows channel changing. If you download a specific app from the Google Play store (the app is free), then you should be able to switch the operating Wi-Fi Direct channel for your ZTE phone. The Motorola Moto G phones do not yet have the ability to switch the Wi-Fi Direct channels using the *FIRST* Tech Challenge SDK. However, the Motorola Moto G phones have their own internal algorithm for selecting a Wi-Fi Direct channel and according to the manufacturer takes into account the wireless activity before selecting a channel.
    Technically speaking, a rooted Android device should also allow the *FIRST* Tech Challenge Robot Controller app to do a channel change. The *FIRST* Tech Challenge Robot Controller needs root or super user status so it can modify a Wi-Fi Direct supplicant file (i.e., configuration file) to force the channel change. Currently (as of September 2015) only the ZTE Speed and Motorola Moto G phones are allowed for the *FIRST* Tech Challenge Competitions. *FIRST* does not recommend rooting your Android devices, since there is a chance that you will damage (i.e., "brick") the devices during the rooting process.

### *4.3 Using the ZTE Speed Channel Changing App*

Currently (September 2015) the only *FIRST* Tech Challenge approved Android device that supports Wi-Fi Direct channel changing is the ZTE Speed. This section contains information on how to use an app to change the Wi-Fi Direct channel on a ZTE Speed phone. The material was taken from the *FIRST* Tech Challenge Training Manual entitled *ZTE Channel Changing App* (first released on 8/3/2015).

### *4.3.1 Downloading the App from Google Play*

The channel changing app is available on Google Play for free for download. Note that this app should be installed on the Android device that is acting as your Robot Controller (i.e., the device that is acting as the Wi-Fi Direct Group Owner). You do not need to install this app on your Driver Station Android device.

IMPORTANT NOTE: This Wi-Fi Direct channel changing app will only work on the ZTE Speed phone. It will not work on other Android devices. It can only change the Wi-Fi Direct channel for a ZTE Speed phone.

The ZTE Speed channel changing app is located on Google Play. To install the app, you first need to connect the ZTE phone to an available wireless network that has access to the Internet. Launch the **Settings** activity on your phone, and select the **Wi-Fi** item to display the Wi-Fi activity.

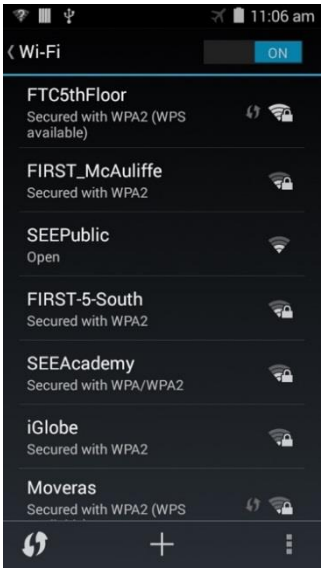



**Figure 23 - Select your wireless network from the Wi-Fi activity.**

Select your desired wireless network from the Wi-Fi activity and provide the password information required to access the wireless network.

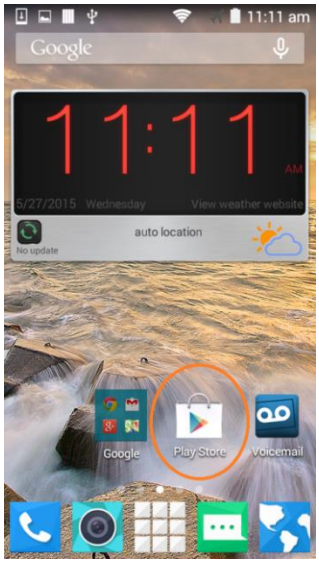



**Figure 24 - Launch the Google Play Store app.**

Once you have connected successfully to your wireless network, launch the Google **Play Store** app from your phone. The Play Store app might prompt you to either login to an existing Google account or create a new one. Follow the onscreen instructions to either create a new (free) account or login to your existing account.

Once you have successfully logged in to Google Play, click on the search icon (a little magnifying glass) and search for the phrase “Wi-Fi Direct Channel Changing”

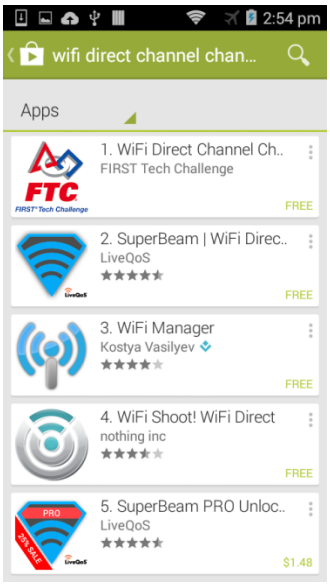



**Figure 25 - Search for the phrase "Wi-Fi direct channel changing"**

Once you have found the Wi-Fi Direct Channel Changing app, click on it and follow the onscreen instructions to install. Note that the application might prompt you to enter a credit card number or some other method of payment. The app is free and no payment method is required. You should be able to hit the “Skip” button to skip the process of providing a method of payment.

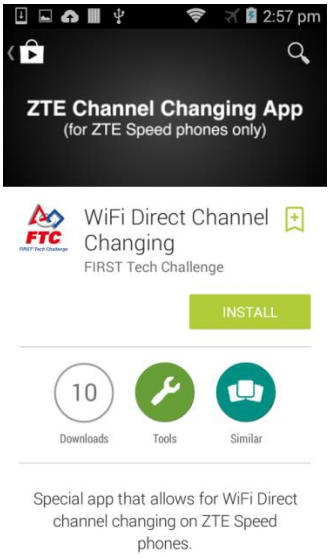



**Figure 26 - Follow the on-screen instructions to install the app.**

If you were able to install the app successfully, then there should be the Wi-Fi Channel Editor icon on your Android device.


**Figure 27 - After installing the app, you should see an icon on your Android's screen.**

IMPORTANT NOTE: After you have successfully installed the app, go to your Wi-Fi settings menu and “forget” the wireless network that you used to connect to Google Play. In general, you do not want to be connected to any wireless networks with the exception of your *FIRST* Tech Challenge Driver Station device.

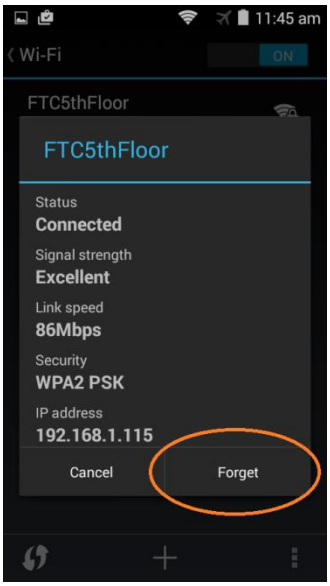



**Figure 28 - Don't forget to forget the wireless network after the installation is complete!**

### *4.3.2 Using the Wi-Fi Direct Channel Changing App*

#### *Changing the Channel*

After you have successfully installed the Wi-Fi Direct channel changing app onto your Robot Controller ZTE Speed phone, you have two ways to launch it:

1. Click on the Wi-Fi Channel Editor icon to launch the app (see Figure 27).
2. You can also launch the app from the *FIRST* Tech Challenge Robot Controller app. Go to the **Settings** menu of the *FIRST* Tech Challenge Robot Controller app and select the **Change Wi-Fi Channel** item to launch the channel changing app.

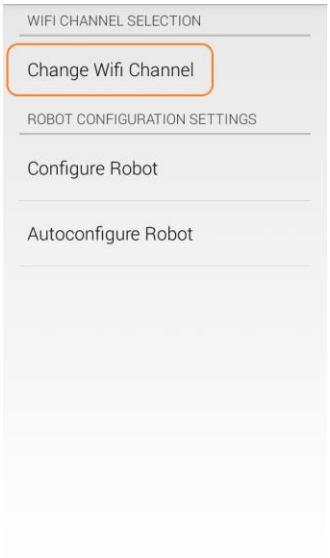



**Figure 29 - Click on Change Wi-Fi Channel from the Settings menu to launch the app.**

To change the channel, use the drop down spinner control to select your desired channel.

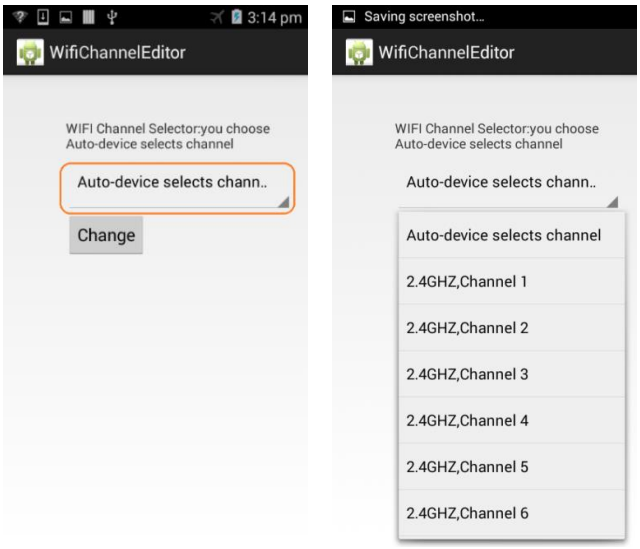



**Figure 30 - Use drop down spinner to select the desired channel.**

Once you have selected your desired channel, press the **Change** button to force the Robot Controller device to change its channel.

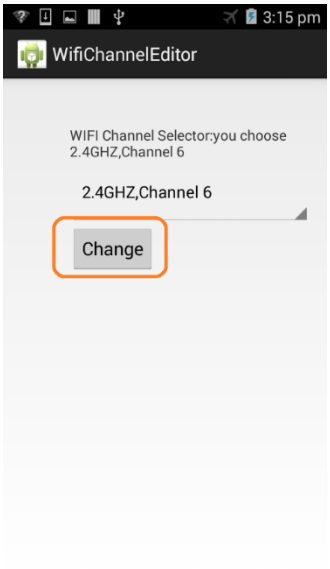



**Figure 31 - Press the Change button to force the channel change.**

The app should display a notification indicating that the listening and operating channel for the Wi-Fi group owner has been changed.

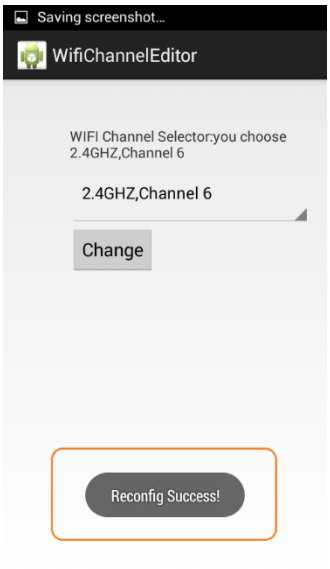



**Figure 32 - The app should display a notification ("Reconfig Success!") once it has completed the channel change.**

### *4.3.3 Un-Pairing 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 un-pair your Driver Station from the Robot Controller, and then re-pair the Driver Station back to the Robot Controller.

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

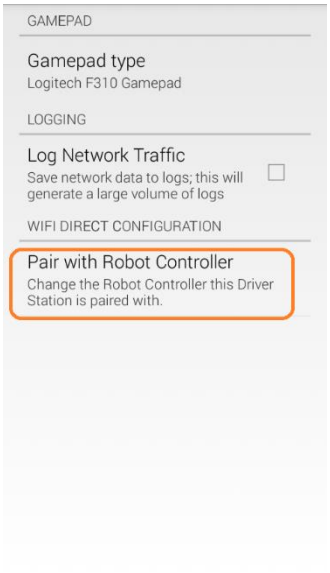



**Figure 33 - Select Pair with Robot Controller.**

From the Pair with Controller screen, select **None** to un-pair your phone.

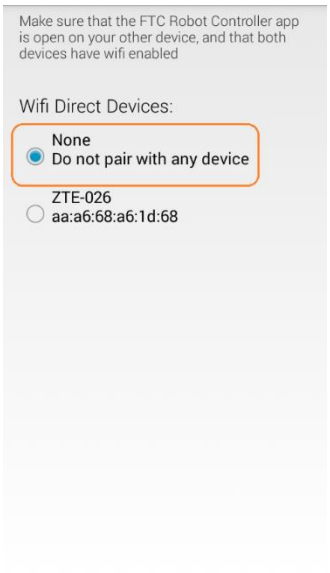



**Figure 34 - Select "None" to un-pair the device. Use the back arrow to return to the main screen.**

Use the back arrow to return back 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.

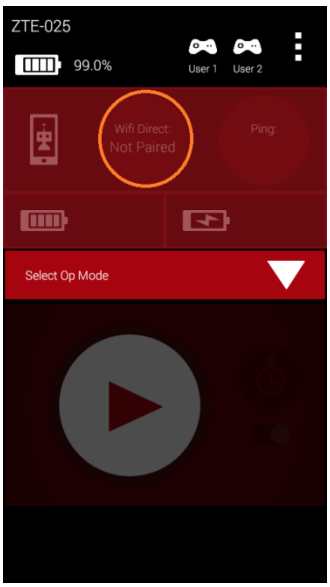



**Figure 35 - The Driver Station should now be un-paired from the Robot Controller.**

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 33). Find the listing for your Robot Controller Android device (in this example "ZTE-026") and select this item.

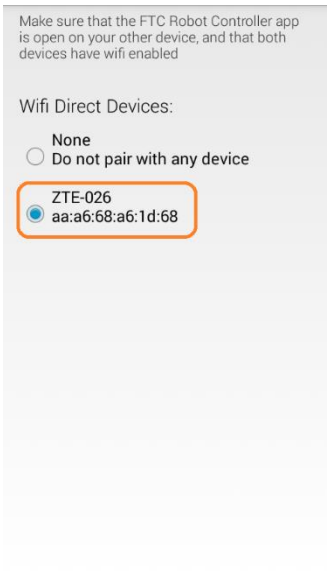



**Figure 36 - Select your target device (in this example ZTE-026), then use the back arrow to return to the main screen.**

Note that your Robot Controller 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.

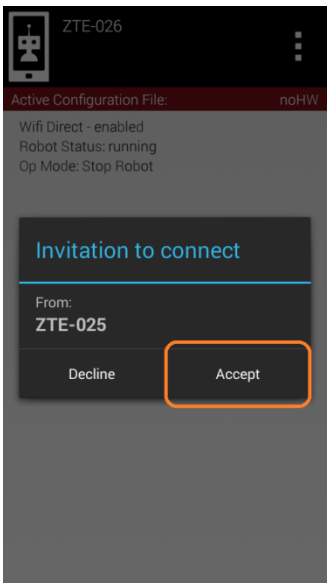



**Figure 37 - Click Accept to approve of the connection request.**

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

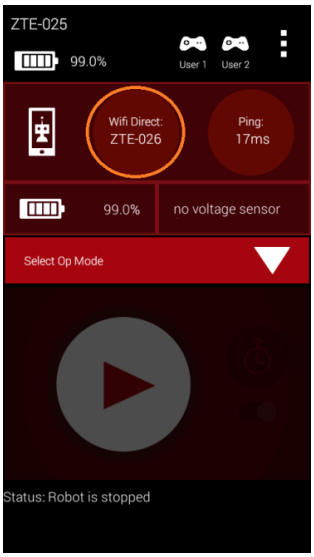



**Figure 38 - Once the connection request is accepted, the Driver Station should connect to the Robot Controller.**

# 5.0 Troubleshooting Common Issues

## *5.1 FIRST* Tech Challenge *Driver Station*

### *5.1.1 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.

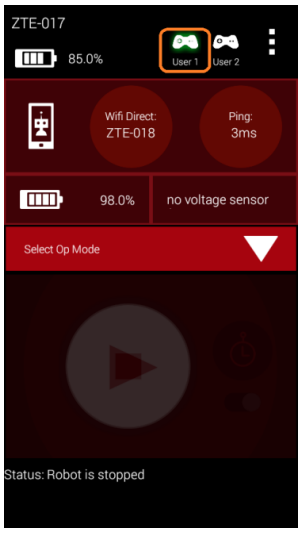



**Figure 39 - The gamepad icon will be highlighted in green if gamepad is recognized and active.**

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

1. For the current season, we are using Logitech F310 gamepads. Make sure the button on the bottom side of the gamepad is set to the “X” position (Xbox emulation mode).
2. In the **Settings** activity of the *FIRST* Tech Challenge Driver Station app, verify that the **Gamepad type** is set to “Logitech F310 Gamepad”.

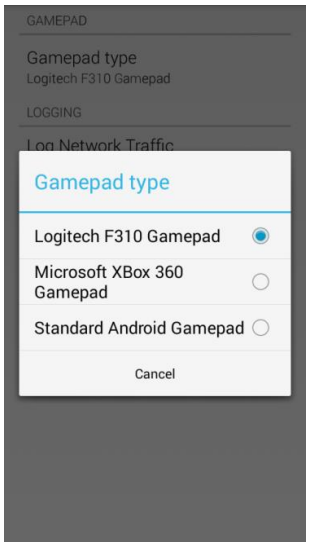



**Figure 40 - Make sure Gamepad type is properly set.**

3. 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 designated a gamepad as driver #2, press the START button and the B (red colored) button on the F310 gamepad.
4. Check the wired connection. If the gamepad was temporarily disconnected from the Driver Station, then the driver will have to re-designate which driver he/she would like to be by pushing START and A (to be driver #1) or START and B (to be driver #2).

### *5.1.2 Driver Station Goes to Sleep While Op Mode 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.

### *5.1.3 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 enough battery.

### *5.1.4 Unable to Find a Specific Op Mode in the Driver Station's List of Available Op Modes*

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

### *5.1.5 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 red 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.

## *5.2 Robot Controller*

### *5.2.1 Robot Controller is Unable to Find a USB Device*

Each USB device (Motor Controller, Servo Controller, Legacy Module and Device Interface Module) that is connected to the USB hub of the Core Power Distribution Module has a unique serial number. The *FIRST* Tech Challenge Robot Controller app will scan the USB bus to look for the devices that are listed in its active configuration file. If the *FIRST* Tech Challenge Robot Controller is unable to find any of the devices that are listed, it will display an error message and communicate this error to the Driver Station.

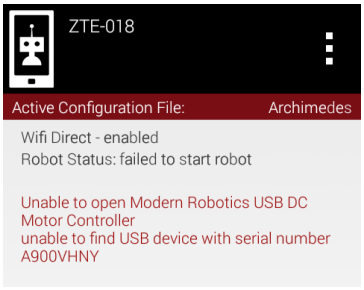



**Figure 41 - Controller is unable to find an attached USB module.**

If you are at an Event and a Team's *FIRST* Tech Challenge Robot Controller app keeps displaying an error that reads something like **"Unable to open XXXX"** or **"Unable to find USB device with serial number XXXX",** then you check the following items,

1. You want to first make sure that there is only one *FIRST* Tech Challenge Robot Controller app installed on the device. Some Teams, particularly those that use the App Inventor to create their op modes, might have more than one *FIRST* Tech Challenge Robot Controller style app installed on their phone. If more than one *FIRST* Tech Challenge Robot Controller type of app is installed on a single phone, then the apps can have problems detecting some of the hardware modules that are connected through the USB bus. This can occur if some of the USB modules are associated with one of the apps, while the rest of the USB modules are associated with a different app.
   If you encounter a Team that has more than one *FIRST* Tech Challenge Robot Controller app installed on their Android device, then you can suggest to the Team that they remove all but one app from their device, and that they make the remaining app the default app for all of their USB hardware modules.
2. If the Team is still experiencing problems finding a hardware module, then verify that the Robot hardware is powered on and connected properly to the Power Distribution module, and select "Restart Robot" from the Settings menu of the Robot Controller.
3. If the Team is still experiencing problems finding a hardware module, then check to make sure that the correct configuration file is activated and select "Restart Robot" from the Settings menu of the Robot Controller.
4. If the Robot Controller is still unable to find the hardware module, then disconnect the Robot Controller from the Power Distribution Module, turn off the Robot and wait a few seconds, turn on the Robot and wait a few seconds, connect the Robot Controller to the Power Distribution Module, and try "Restart Robot"
5. Finally, if there are still issues, disconnect the Robot Controller from the Power Distribution Module, turn off the Robot and the Robot Controller, wait a few seconds, turn them both back on, launch the Robot Controller app, connect the Robot Controller to the Power Distribution Module, and try "Restart Robot"

### *5.2.2 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.

### *5.2.3 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.

### *5.2.4 Unable to find a hardware device with the name "…"*

A very commonly encountered error occurs when the user tries to run a specific op mode, 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 op mode is trying to access.

This error will occur if the op mode 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 and for apps created using the App Inventor.

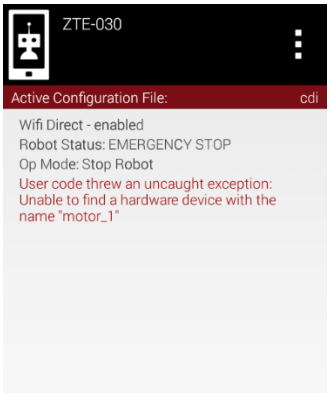



**Figure 42 - The name used by the op mode must match the name used in the configuration file for the device exactly.**

If you are at an Event an you encounter this error message, ask the Team to verify that the name that they use in their op mode 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*.

# 6.0 Useful Tips and Tricks

## *6.1 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. If you are using ZTE Speed phones, you can change the operating channel and monitor the wireless quality on different channels at a venue.

## *6.2 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,

http://developer.android.com/tools/help/adb.html

On the Android device running the *FIRST* Tech Challenge Robot Controller app, the log file is stored in the following directory,

/sdcard/com.qualcomm.ftcRobotcontroller.logcat

On the Android device running the *FIRST* Tech Challenge Driver Station app, the log file is stored in the following directory,

/sdcard/com.qualcomm.ftcdriverstation.logcat

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 /sdcard/com.qualcomm.ftcRobotcontroller.logcat .
```

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. Note that if you used the App Inventor to generate the *FIRST* Tech Challenge Robot Controller apps for a device, the name of the log file might be different from the default com.qualcomm.ftcRobotcontroller.logcat filename.

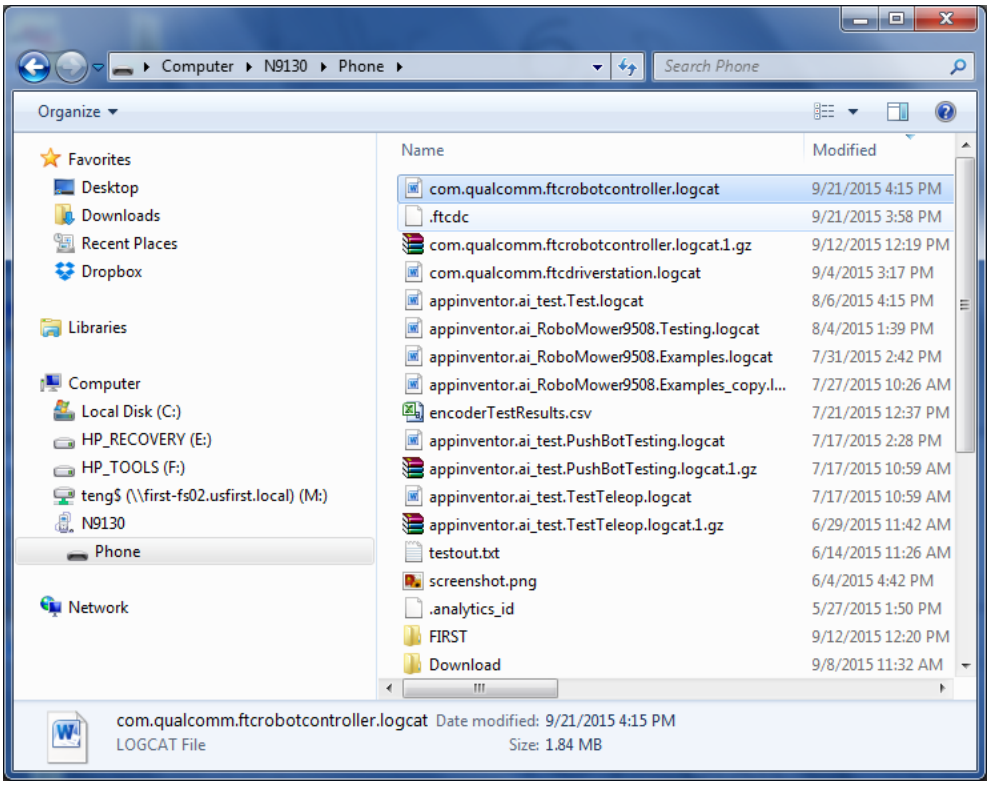



**Figure 43 - Windows users can use the File Explorer to locate and copy the log file.**

# 7.0 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/

## *7.1 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 (http://www.wireshark.org). Install the Wireshark software per the instructions (refer to the Wireshark website for documentation).

In order 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 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)

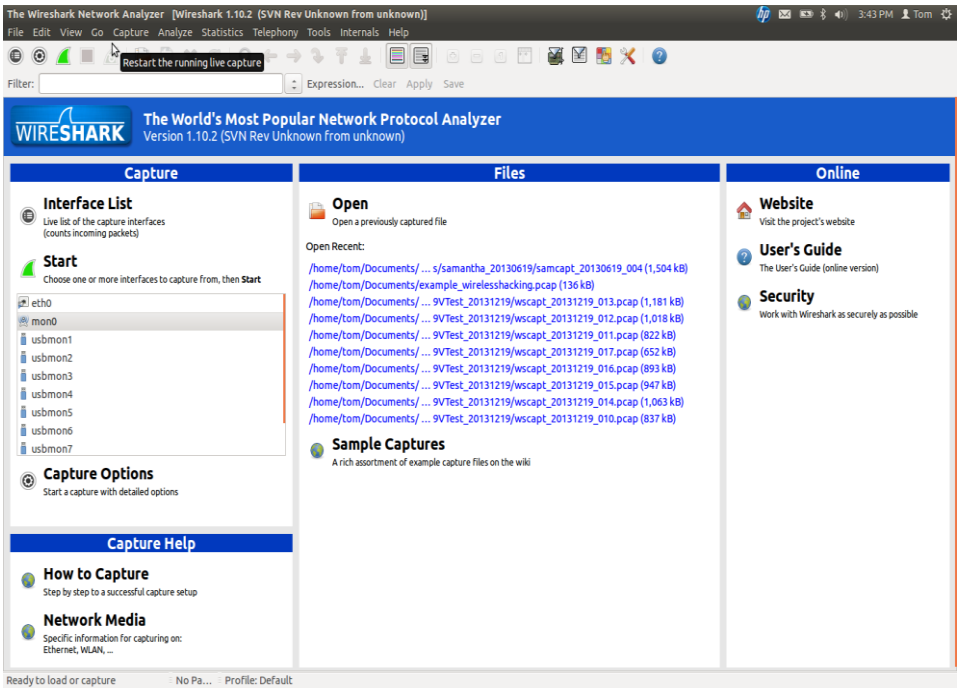



**Figure 44 –Wireshark user interface.**

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:

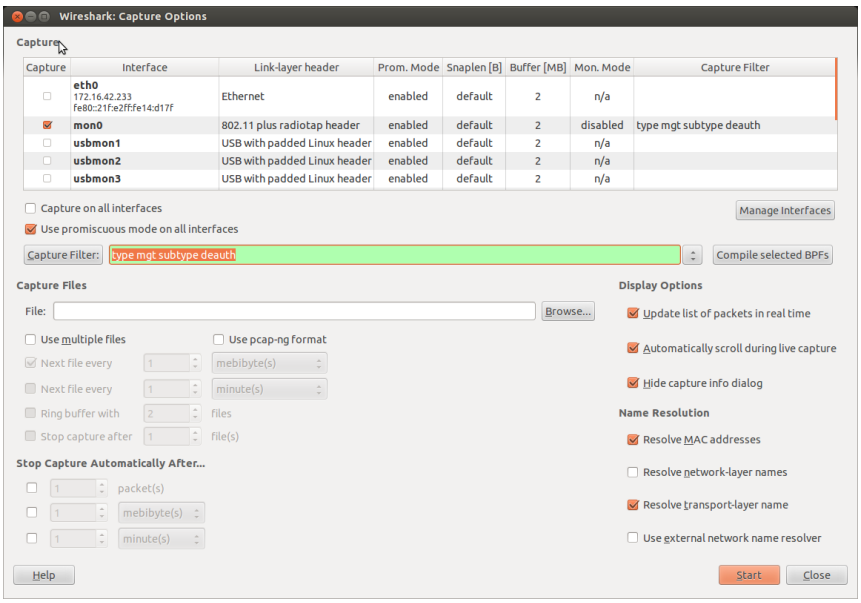



**Figure 45 - Make sure you select the adapter that is running in monitor mode.**

Make sure that the wireless adapter which is running in *monitor mode* is selected as the capture interface. In the screen shot 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.

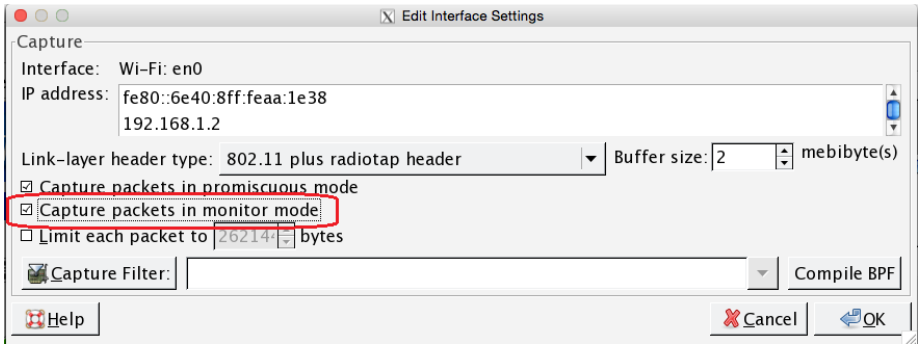



**Figure 46 - Mac users can select the Capture packets in monitor mode option.**

In the Wireshark: Capture Options window, there should be a text box next to a button called “Capture Filter”. You want to specify your capture filter in this text box. The capture filter will filter only a limited number of packets that match the filter criteria for your Wireshark capture session. In this case, you want to specify the filter as “type mgt subtype deauth” (you do not type in the quotation marks... just provide the text). If the syntax for your capture filter is correct, the background color of the text box should turn green.

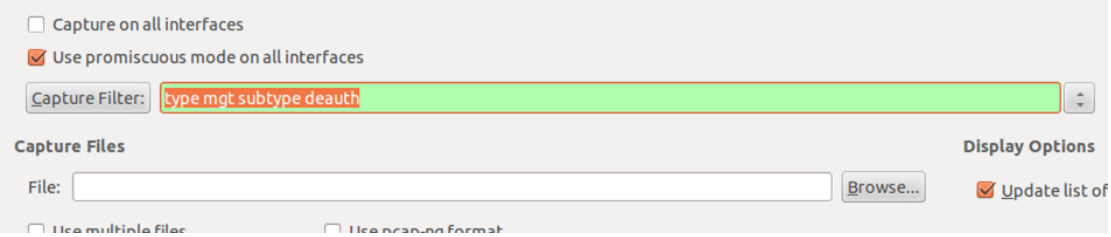



**Figure 47 - Specify the capture filter "type mgt subtype deauth" in the text box.**

When you are ready to start your capture, simply hit the Start button. If the capture filter is applied properly, the Wireshark panel will display any packet of type mgt and subtype deauth. The following image shows some DEAUTH packets, which were captured by Wireshark:

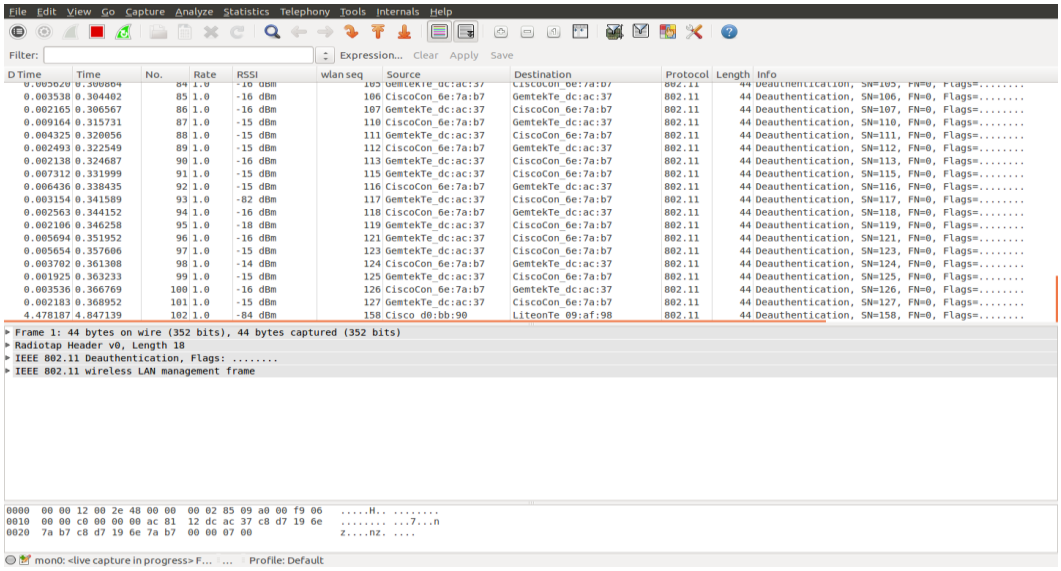



**Figure 48 - DEAUTH packets captured by Wireshark.**

You can run this capture (with the filter applied) during an *FIRST* Tech Challenge Event and it will only display the DEAUTH packets. It is normal for DEAUTH packets to occur, but if you see several DEAUTH packets in a row for a specific time during Event, it might indicate an intentional DEAUTH attack occurred.

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 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.

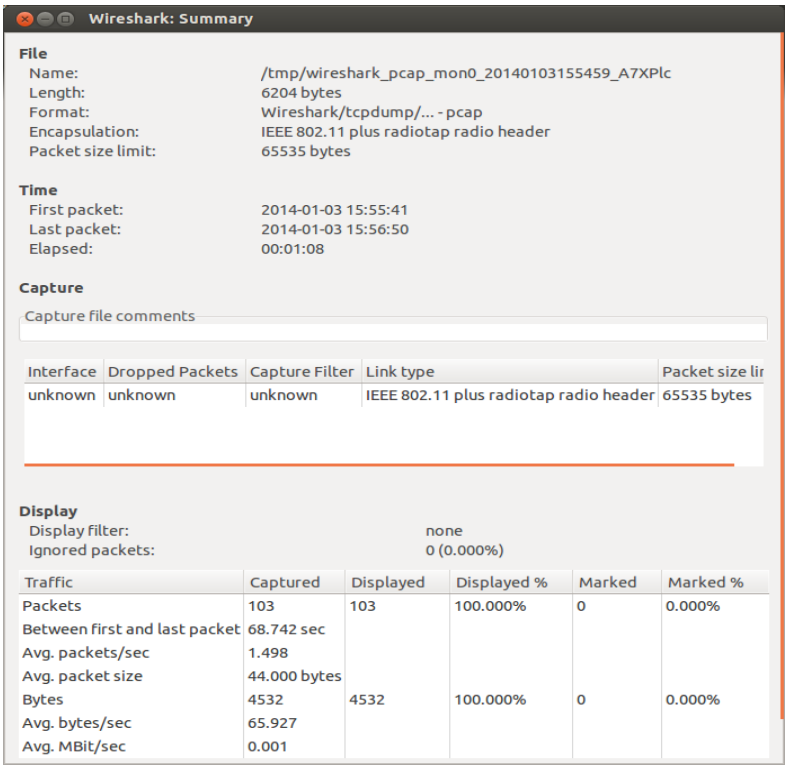



**Figure 49 - 103 packets were captured in this example.**

## *7.2 Using a Display Filter to Look for Malformed Probe Responses*

You can also use Wireshark to see if there are any malformed Wi-Fi Direct probe responses in the area which might cause problems when Teams try to pair their Robots (see the section entitled 2.3.5 Are Android Devices Rebooting Upon FIRST Tech Challenge Driver Station Startup or Wi-Fi Direct Scanning? of this document for details about this problem).

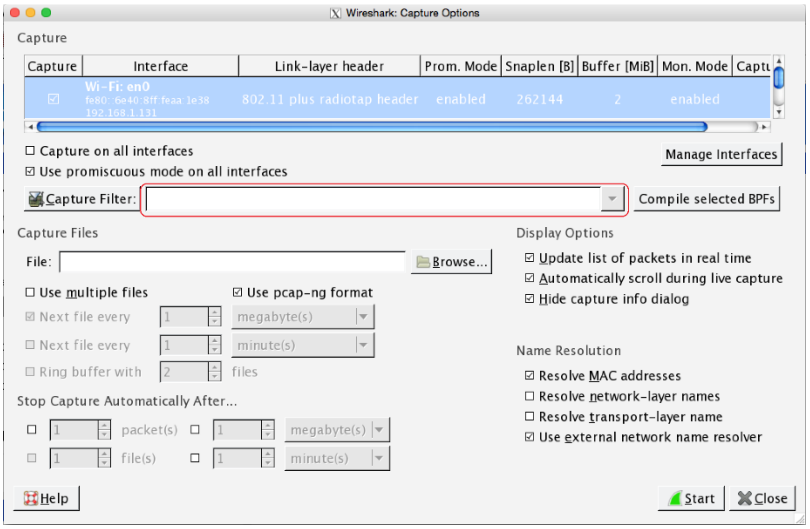



**Figure 50 - The Capture Filter text box should be empty.**

To try and detect this problem you can disable the capture filter on Wireshark. Open the Wireshark capture options window (the Mac OS version is shown in Figure 50) and make sure that **Capture Filter** text box is empty. When this Field is left empty, Wireshark will capture all types of packets.

When you are ready to capture some sample data, push the green start button. Since you configured Wireshark to run without any capture filter in place, it will collect a large amount of data. You can run the capture for a moment (30 seconds should be sufficient) then hit the stop button to stop the capture.

Once the data has been captured, you can save the data to a local file. You can also apply a *display filter* to the data view.

The bug in Android Kit Kat that forces the Driver Station phone to reboot is caused by the presence of a malformed probe response Wi-Direct (P2P) packet. To look for possible malformed probe responses, you can to type the following text into the display filter text box of Wireshark:

(wlan.fc.type_subtype == 0x05|| wlan.fc.type_subtype == 0x08) && wlan_mgt.ssid matches "DIRECT-" && (wlan_mgt.ssid matches "\\W" || wifi_p2p.dev_info.dev_name matches "\\W")

The first condition of the filter (wlan.fc.type_subtype == 0x05 || wlan.fc.type_subtype == 0x08) will make Wireshark only display packets that are 802.11 probe response packets or 802.11 beacon packets.[8] The second condition (wlan_mgt.ssid matches "DIRECT-") will cause Wireshark to only display packets with an SSID that contains the expression "DIRECT-". Wi-Fi Direct networks incorporate this expression in their SSIDs. The final condition (wlan_mgt.ssid matches "\\W" || wifi_p2p.dev_info.dev_name matches "\\W") use *regular expressions* to find packets where there are "non word" characters embedded in the SSID or where there are non "word" characters embedded in the Wi-Fi Direct (P2P) device name. Note that the "W" in "\\W" is capitalized (to indicate it's a "non word" character).

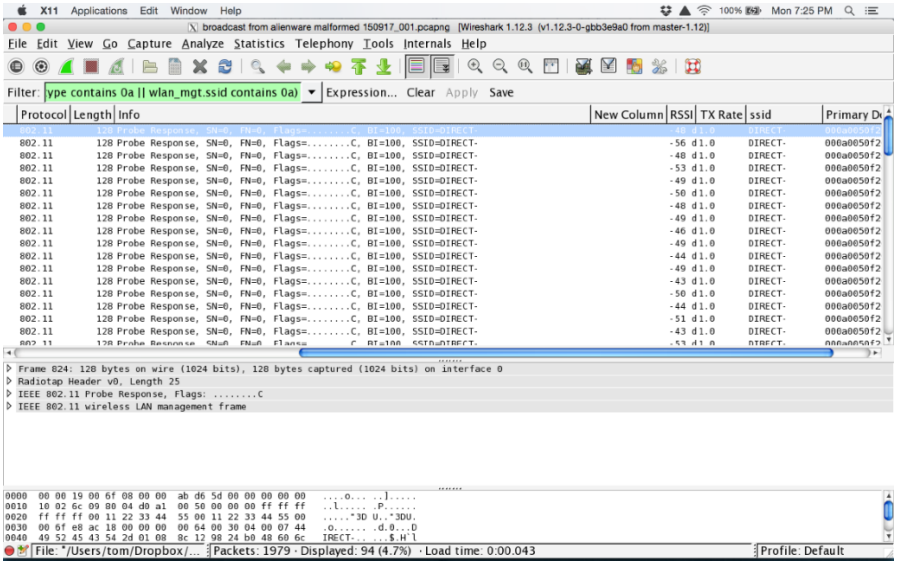



**Figure 51 – A packet that matches the display filter criteria is a potential malformed probe response packet.**

[8] A good reference for Wireshark display filters can be found at the following website http://www.lovemytool.com/blog/2010/07/wireshark-wireless-display-and-capture-filters-samples-part-2-by-joke-snelders.html

If you detect any packets that are displayed using this display filter, then you might have a malformed packet in your immediate area, which might cause Wi-Fi Direct discovery issues for Android Kit Kat devices. You will need to inspect the captured packets to see if you can determine if any are problematic.

Wireshark will sometimes flag malformed packets that can cause the reboot problem. In the following image, Wireshark has flagged a beacon frame as a malformed packet. If you look at the SSID for this packet, there is a newline (represented by "\n") character at the end of the SSID for this packet ("DIRECT-2V-4486-RC\n"). This malformed Wi-Fi Direct packet will cause a Driver Station to reboot if it is running Android Kit Kat and is attempting to do a Wi-Fi Direct scan.

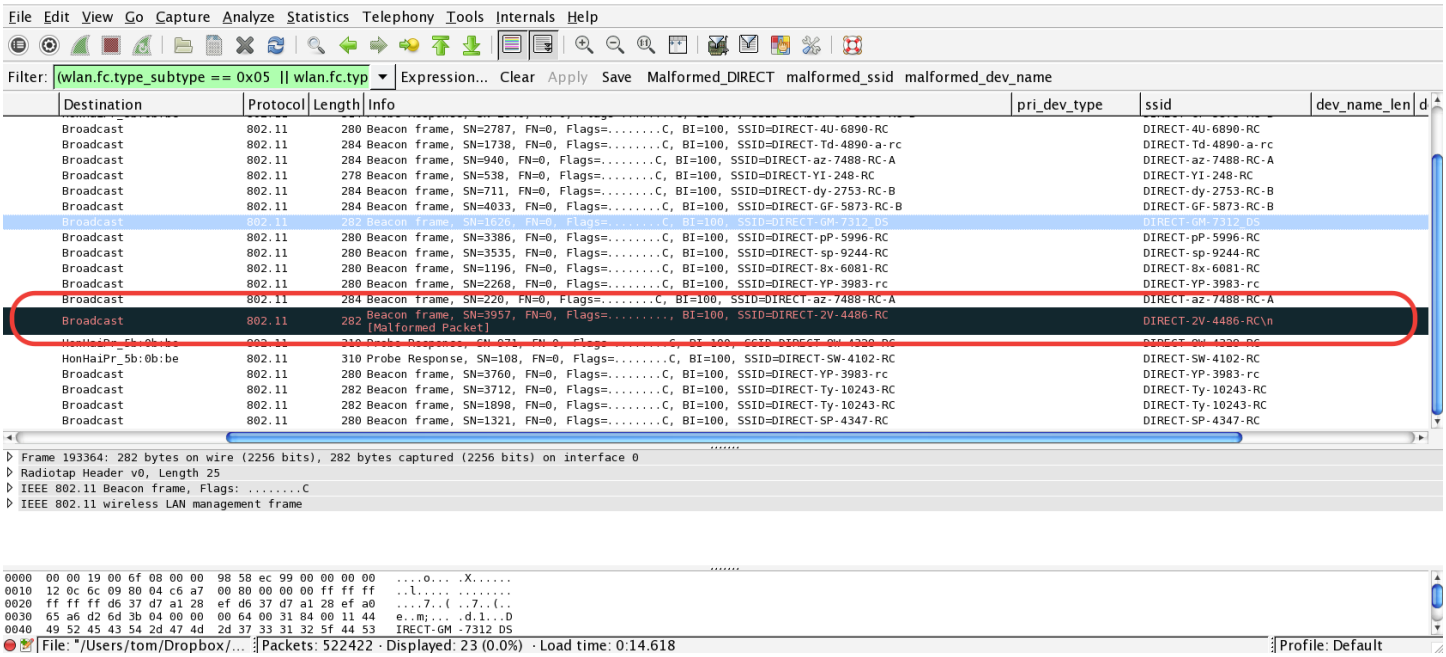
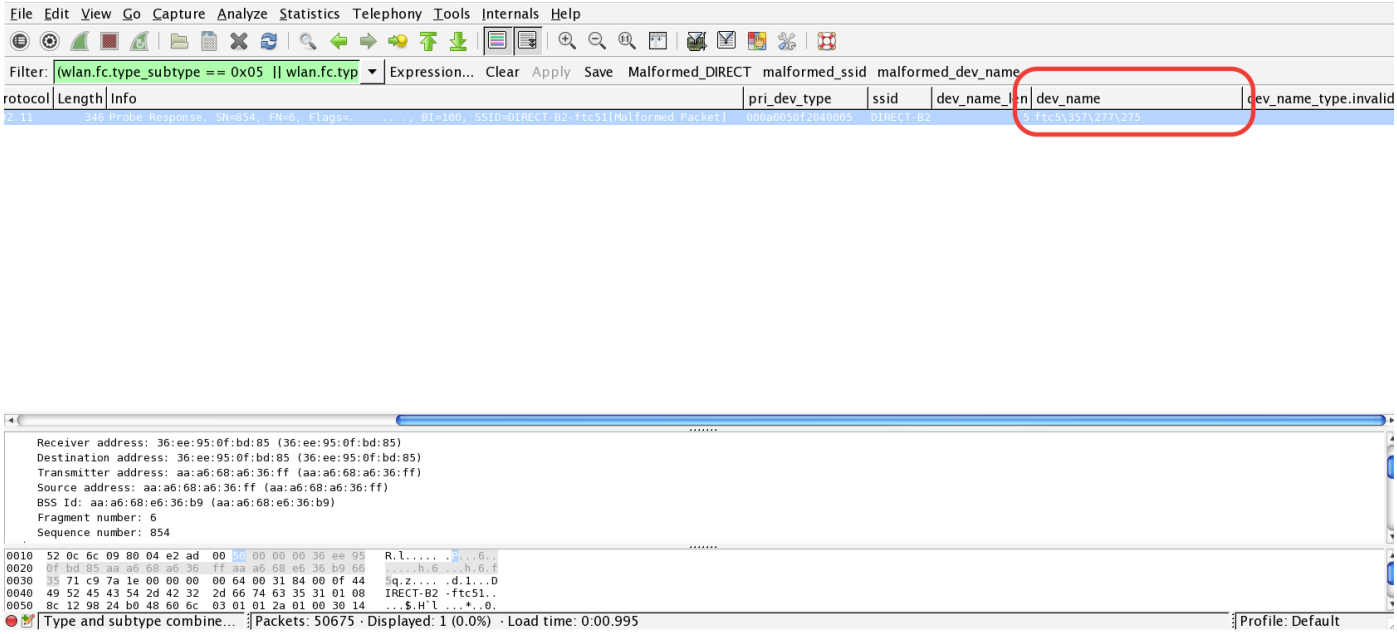



**Figure 52 - Wireshark will flag some malformed packets (note the "(Malformed Packet)" expression for the highlighted packet).**

The figure below shows another packet that was flagged by Wireshark as being "malformed". Note that in this case, the WiFi Direct device name did not have a newline character embedded in it, but it did have some non-word characters in the device name that caused the reboot problem for the Kit Kat phones.




**\Figure 53 - In this screenshot, Wireshark flags a packet that has "non word" characters in its device name as malformed.**

Unfortunately, Wireshark is not able to flag all malformed packets. There are some malformed packets will not be flagged by Wireshark, yet they will cause a Kit Kat device to reboot if it attempts to do a Wi-Fi Direct scan.

In the following image there are several packets that were not flagged as “malformed” by Wireshark, but these packets will cause a Kit Kat Android device to reboot when the device attempts to do a Wi-Fi Direct scan. If you look closely at each of the captured packets in the example, you’ll see that the wifi_p2p.dev_info.dev_name field contains several “non word” characters (represented by the “\357\277\275” expression in the dev_name field). Although Wireshark did not flag these packets as “malformed” these packets will cause a problem for Kit Kat devices. When examining the captured data you might need to look for packets that contain such non-word characters to see if they might be causing the reboot problems with your Kit Kat devices.

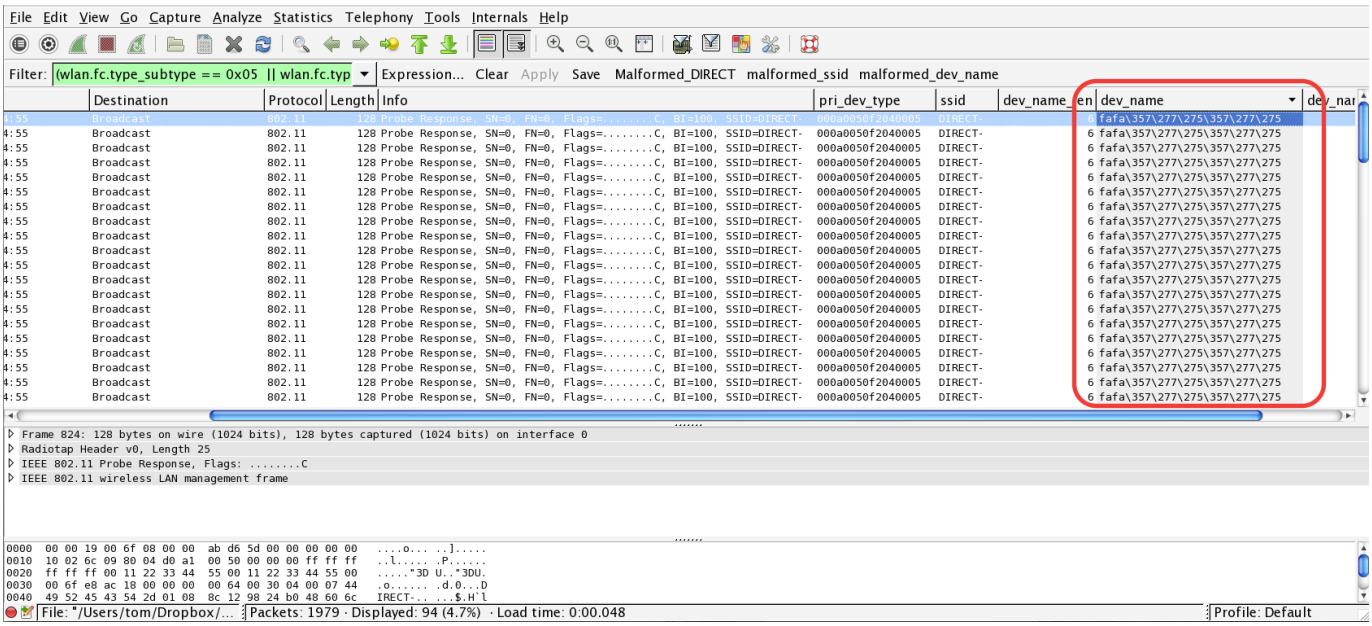



**Figure 54 – The field wifi_p2p.dev_info.dev_name in this example contains several “non word” characters.**

Sometimes it is helpful to refine your Wireshark Display filters to make it easier to find problematic packets. It can be helpful to filter the packets using different criteria. For example, it might be helpful to first use the following filter to see if there are any packets that have SSID's with "non word" characters in them:

(wlan.fc.type_subtype == 0x05|| wlan.fc.type_subtype == 0x08) && wlan_mgt.ssid matches "DIRECT-" && wlan_mgt.ssid matches "\\W"

Then it might be helpful to filter the data using the following filter, which will display packets that have device names with "non word" characters embedded:

(wlan.fc.type_subtype == 0x05|| wlan.fc.type_subtype == 0x08) && wlan_mgt.ssid matches "DIRECT-" && wifi_p2p.dev_info.dev_name matches "\\W"

Wireshark can be a useful tool, especially as you gain experience at filtering and examining the captured packet data.

## 8.0 Getting Additional Help

If you have questions about the *FIRST* Tech Challenge Control System, you can visit the *FIRST* Tech Challenge Technology forum and search for related posts or post your own questions: http://ftcforum.usfirst.org/forumdisplay.php?156-ftc-Technology

There is also a hidden *FIRST* Tech Challenge forum 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.

If you are an FTA/CSA/WTA and would like access to the private FTA/CSA/WTA-only forum, please contact your local *FIRST* Tech Challenge Affiliate Partner and ask them for instructions on how to gain access to this form.

# 2015-2016 *FIRST*® Tech Challenge

# Control System Troubleshooting Guide

# Appendices

# Appendix A: 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 very helpful to review these files to see if we can notice any pattern or clues that can help diagnose the problem.

### *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'd 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.

If you haven't done so already, take the time to check the time on your phones (no pun intended).

### *The FIRST* Tech Challenge *Log Files*

The logcat files are accessible on your phones. By default the FtcRobotController and the FtcDriverStation apps store these files (as text files) in the directory /sdcard on your Android device. This /sdcard directory is the local path on the phone (i.e., it is an internal path to the directory on the phone).

For the FtcRobotController app, if you are using Android Studio to write your app, the path on your phone to the log file is as follows,

> /sdcard/com.qualcomm.ftcRobotcontroller.logcat

If you are using the App Inventor to write your app, the path on your phone to the log file will look something like this,

> /sdcard/app_inventor.ai_ftc.<NAMEOFMYAPP>.logcat

In the example above, the parametere <NAMEOFMYAPP> is the name of your App Inventor-generated app or project name. So for example, if the App Inventor project that you used to create an app is called "MyRobotController" then the path to this app's log file is as follows,

> /sdcard/appinventor.ai_ftc.MyRobotController.logcat

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

> /sdcard/com.qualcomm.ftcdriverstation.logcat

### *Viewing the FIRST* Tech Challenge *Robot Controller Log File*

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

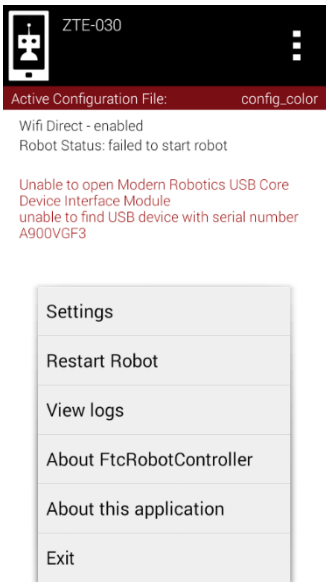



**Figure 55 – Click on the three dots in the upper right hand portion of the screen then select View logs from the menu.**

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


**Figure 56 – 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.

## *Finding the Log Files*

### *File Manager App*

If you are using a phone like the ZTE Speed, then you should be able to locate the files using the phone's File Manager app.

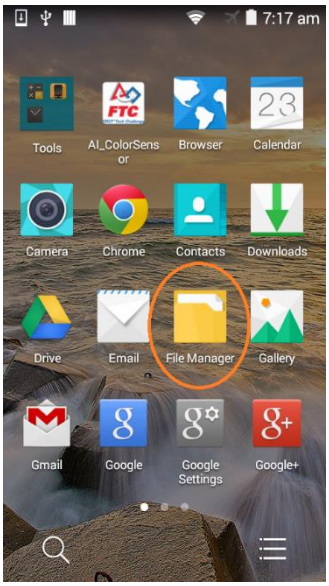



Figure 57 - You can use the File Manager app to locate the files on your phone.

Launch the File Manager app on your ZTE Speed and then click on the **Phone** tab near the top of the main screen.

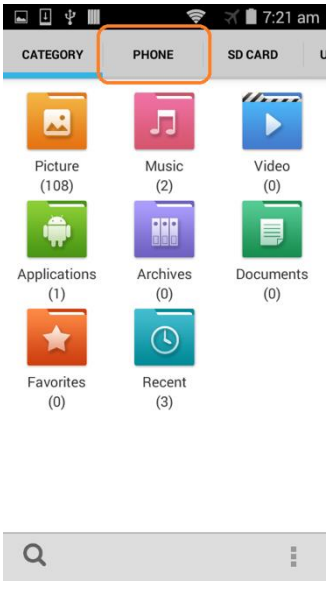



Figure 58 – Click on the Phone tab to browse the contents of your phone.

If you click on the **Phone** tab you should see the directory structure for your phone's local storage. What you are viewing is the contents of the /sdcard directory on the phone. If you scroll down to the bottom of the topmost directory (which corresponds to /sdcard on your phone) you will see one or more log files:

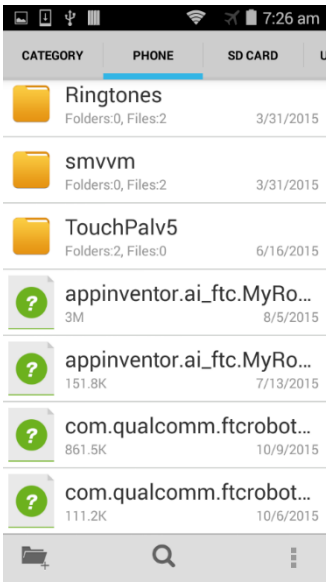



**Figure 59 - If you scroll towards the bottom you will see the log files.**

In Figure 59 you see that the phone in this example has at least two different types of log file. One was generated by the app inventor and one was generated by the *FIRST* Tech Challenge Robot Controller app. In this example, there appears to be four log files. There are actually only two log files that we are mostly concerned with. For example, there is a file that was generated by the App Inventor with the following name,

appinventor.ai_ftc.MyRobotController.logcat

There is also a second file with a similar name in the same directory:

appinventor.ai_ftc.MyRobotController.logcat.1.gz

This second file contains old archived log data. This second file is a compressed file. Typically when you are troubleshooting, you would like to view the more current log data so you are more interested in the file with the ".logcat" suffix.

Even though the File Manager lets you browse the directory structure on your phone and see where these files are located, typically there isn't an app associated with these file types that make it easy to view these files on your Android phone. It is usually easier to copy these files to your PC and use an application on the PC to browse the file.

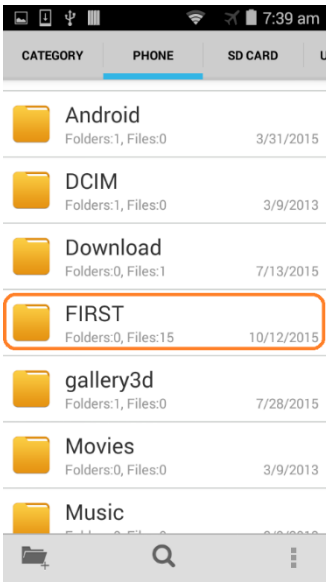



**Figure 60 – There is a *FIRST* subdirectory that contains *FIRST*-related configuration and other files.**

Before we move onto the next topic, it is useful to note that in this main storage directory there is a subfolder (with an internal path of /sdcard/FIRST) that contains useful *FIRST*-related files. You can view these files by opening the *FIRST* folder using the File Manager app.

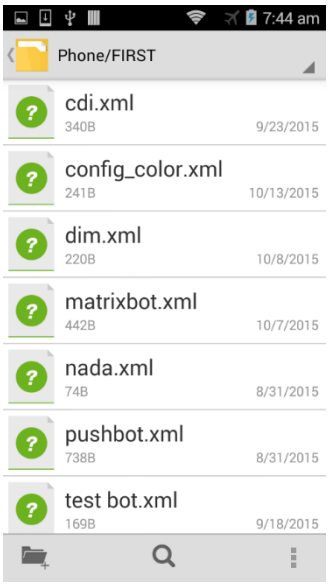



**Figure 61 – The *FIRST* folder contains the Robot configuration files (.xml) plus some hidden files (not visible in this image).**

This *FIRST* folder contains the .xml files that contain the Robot configuration information. They are created by the Robot Controller app whenever a user generates a new configuration for a Robot. Also contained in this folder (but not visible in the File Manager) are some hidden files that are used by the *FIRST* Tech Challenge apps for storing other data.

### *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 ZTE Speed phone and find the log files. The first thing you should do is connect the phone via a USB cable to your windows machine. If you haven't already, you should install the ZTE driver for the Windows. You only need to

do this the first time you ever connect a ZTE Speed phone to the PC. You can refer to **section 7.8.4** of the ***FIRST* Tech Challenge *Training Manual: JAVA Programming for the Next Gen Controller*** for detailed instructions on how to install the ZTE driver onto a Windows PC.

Once the phone is connected, you want to make sure the phone is in media device mode:

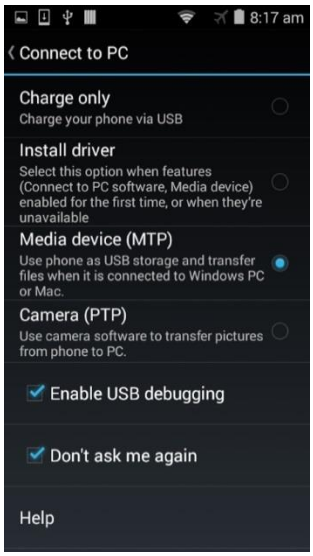



**Figure 62 - Make sure your phone is in Media device mode.**

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 ("N9130") connected to your PC:

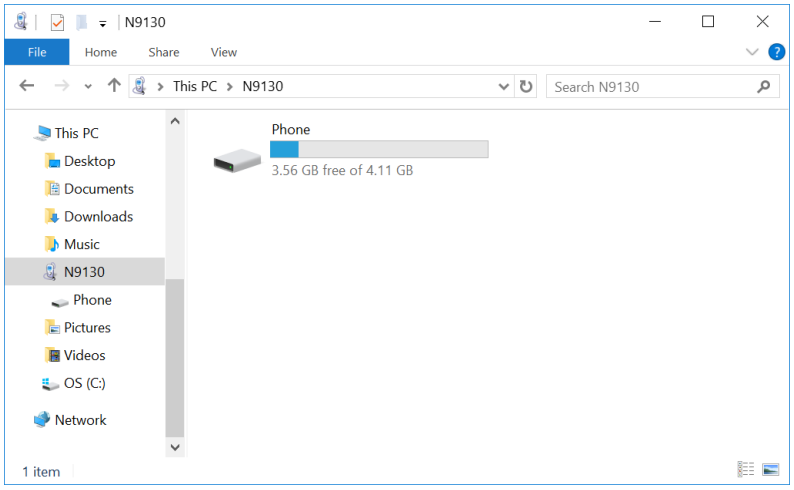



**Figure 63 –The phone should appear as a media device connected to your PC.**

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

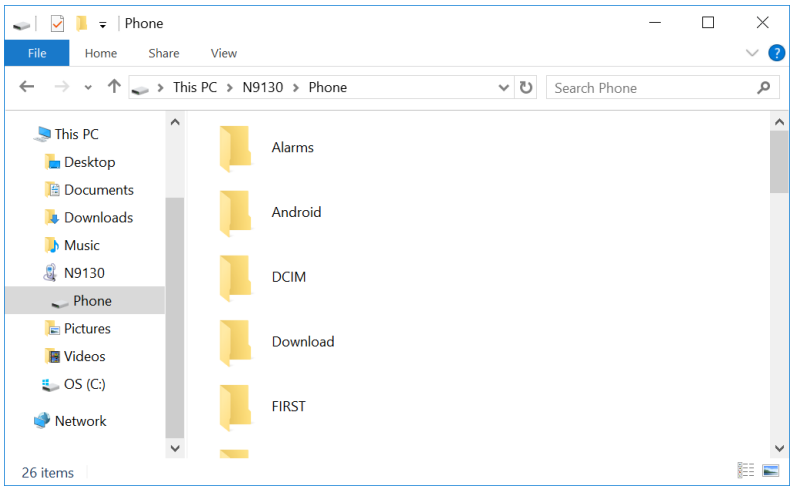



**Figure 64 – You can browse the main directory of the phone.**

When you open up the phone's "hard drive" you are actually exploring the /sdcard directory of your ZTE 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 logcat files:

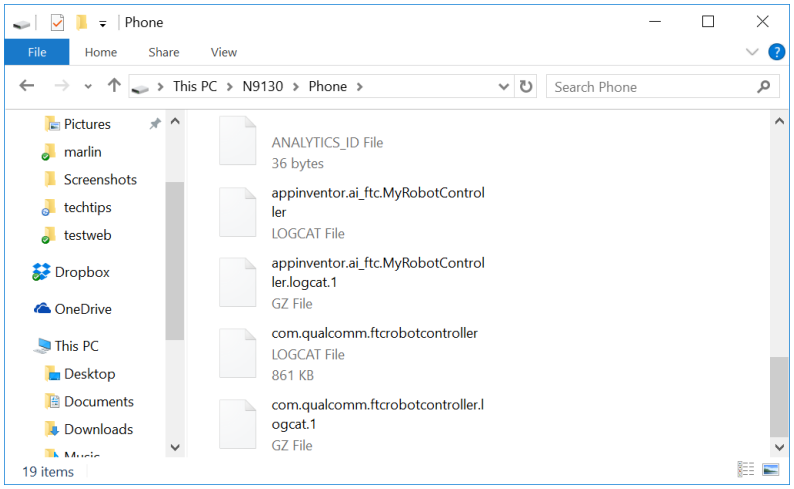



**Figure 65 – The log files should be visible in this directory.**

In Figure 65 you can see the same log files that you saw in Figure 59. However, when you use the Windows File Explorer you can copy one or more log files and then paste the file onto your on PC's hard drive. This allows you to create a local copy of the log file on your computer that you can browse.

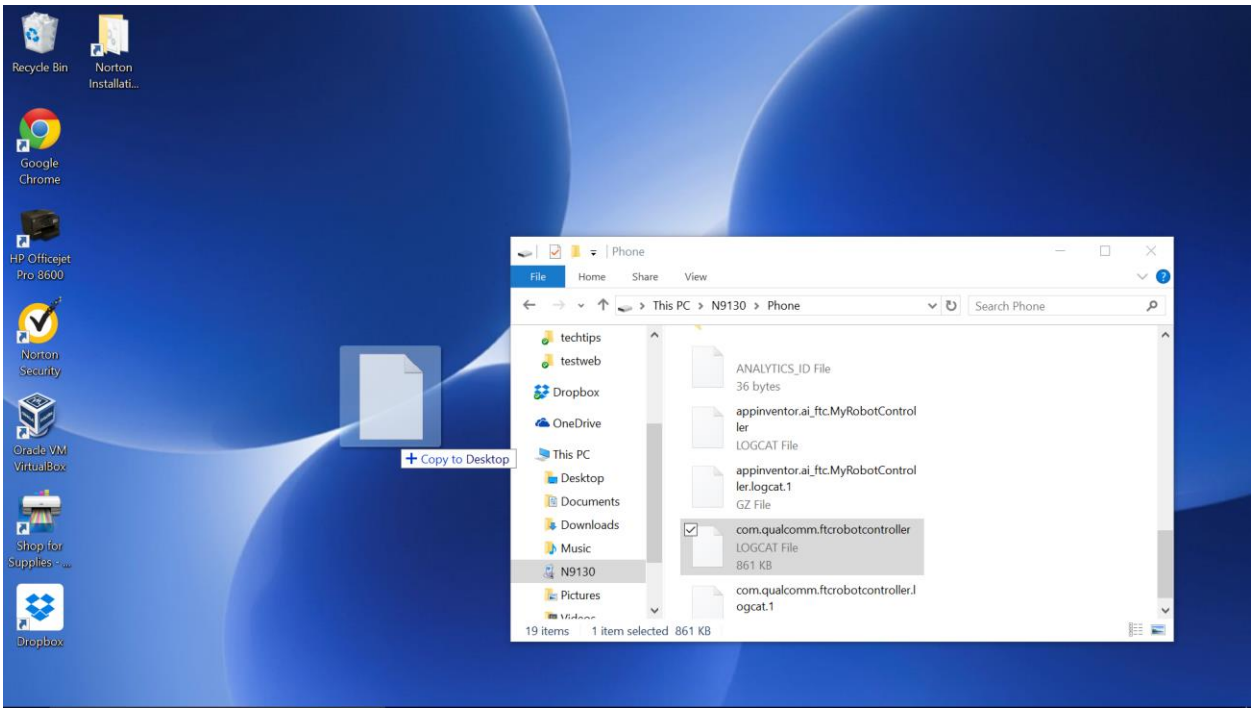



**Figure 66 – You can make a copy of a log file by dragging and drop the file to your personal computer.**

### *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:

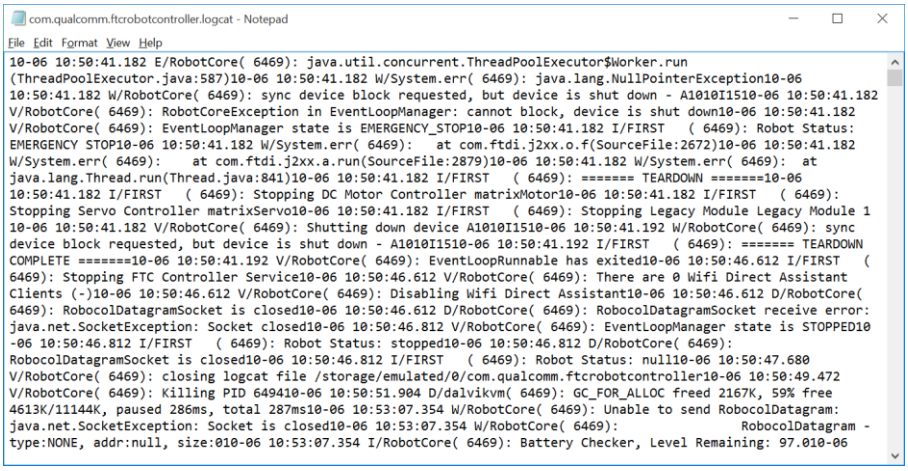



**Figure 67 – Using Notepad might not be as useful to browse the log files.**

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.

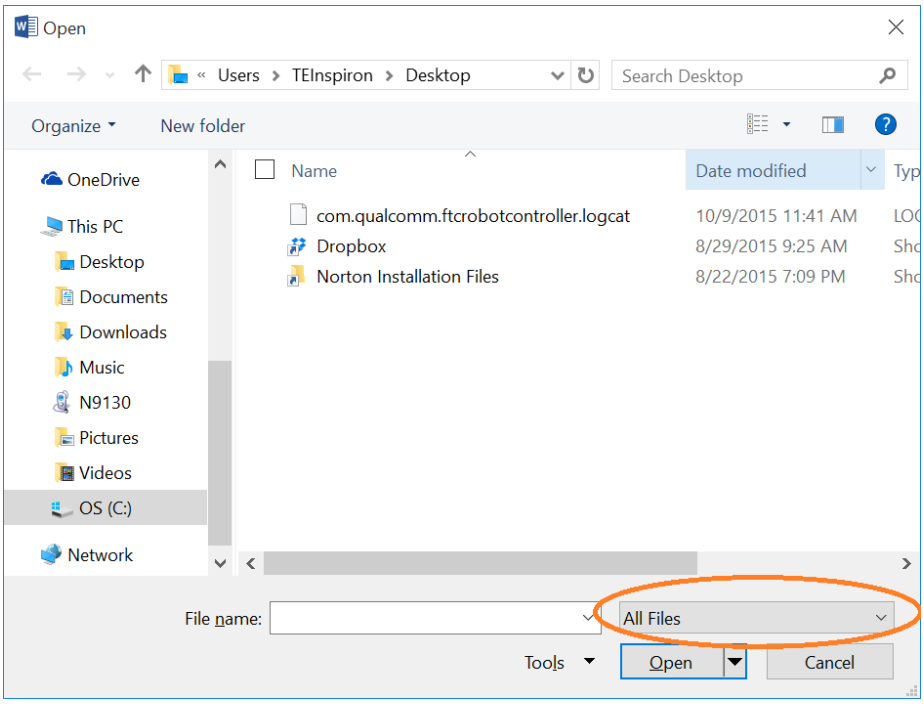



**Figure 68 – Make sure the file filter is set to "All Files" when you browse to find your log file.**

Using Microsoft Word to view the logcat file makes it much easier to read and search for text in a log file.

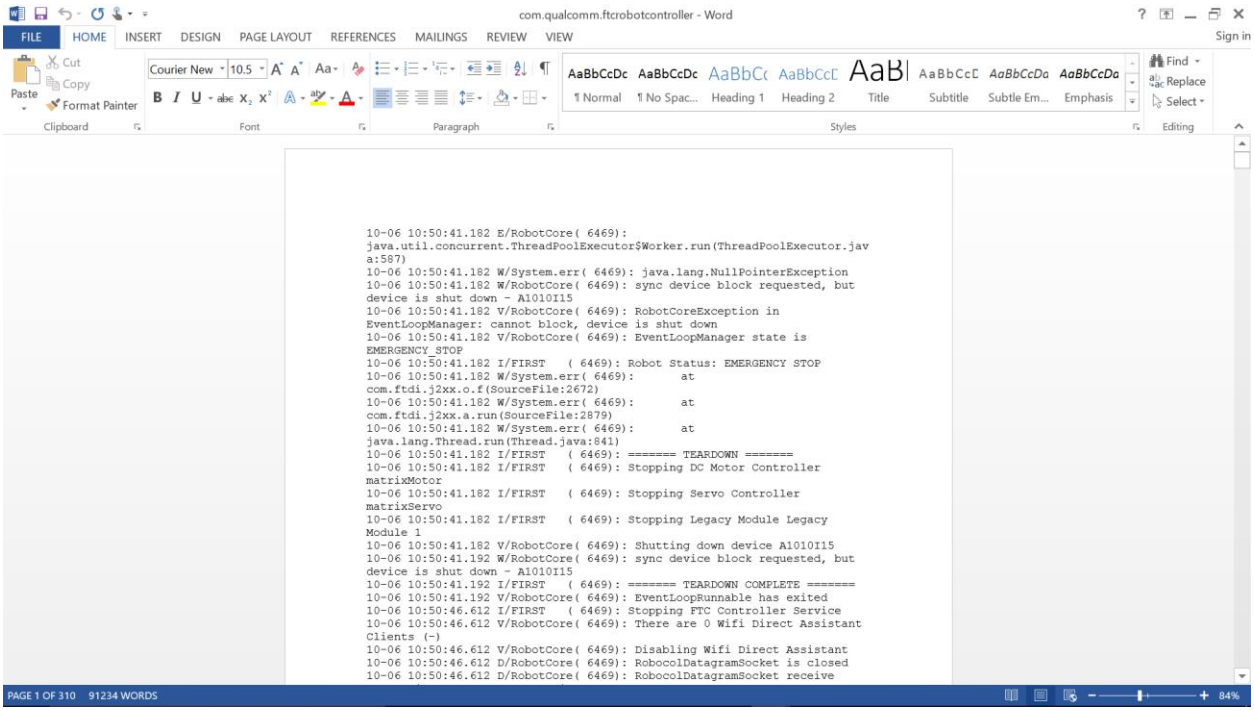



**Figure 69 – Using Microsoft Word to view the log file makes it easier to read the contents and search for specific text strings.**

***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 Android Debug Bridge for Troubleshooting*
The Android Debug Bridge (ADB) is a utility program that is include with the Android Software Development Kit (SDK) platform tools. ADB is a program that you can invoke from a command line. It is very a very helpful utility. In order 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 include 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 sell 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:

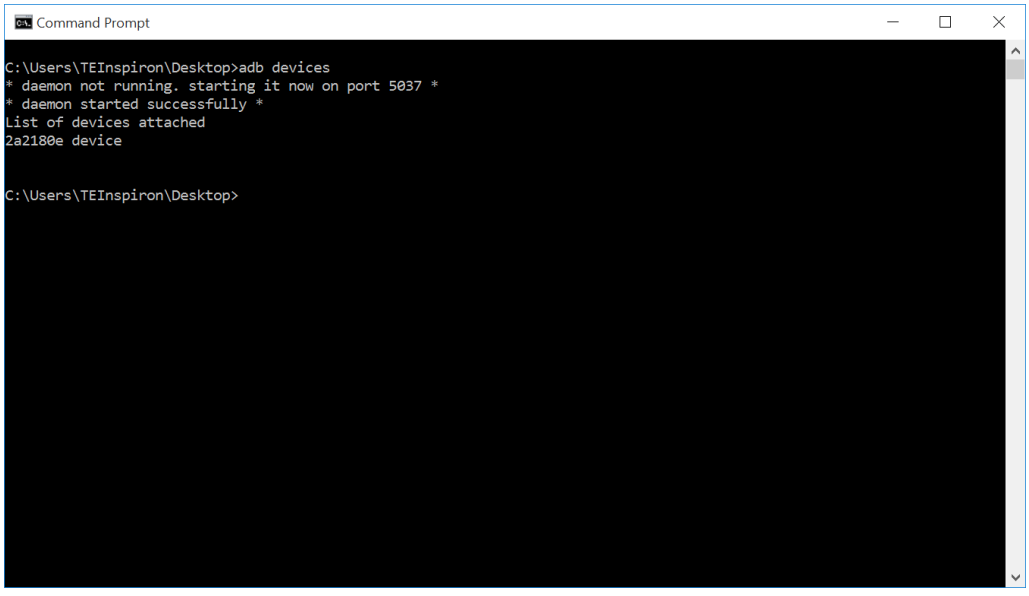


**Figure 70 – From a terminal or command line interface type "adb devices" to see a listing of attached Android devices.**

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

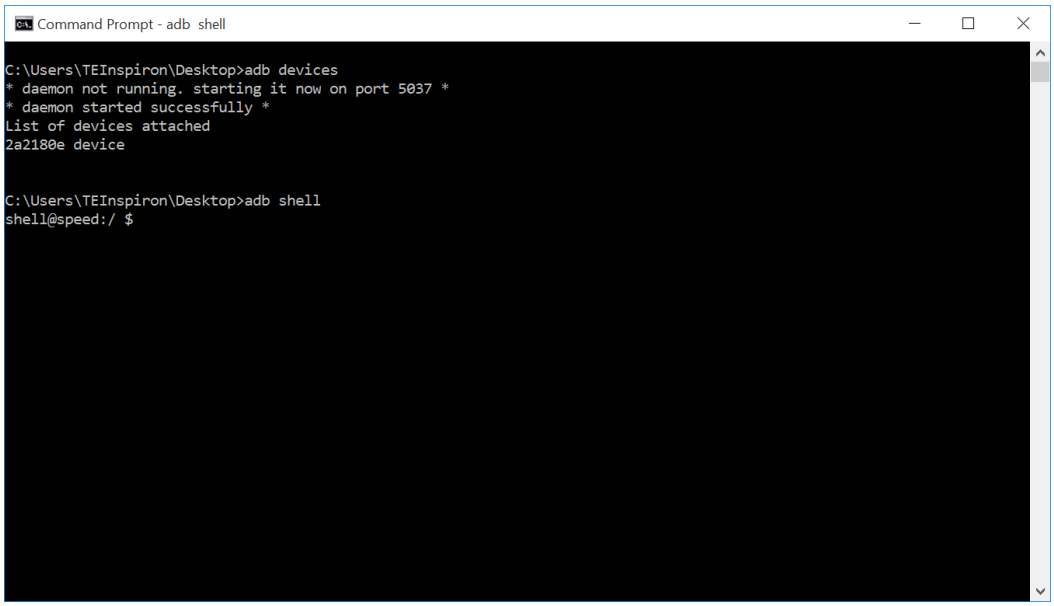



**Figure 71 – Type in "adb shell" to create a terminal session with your Android device.**

If you look at Figure 71 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:

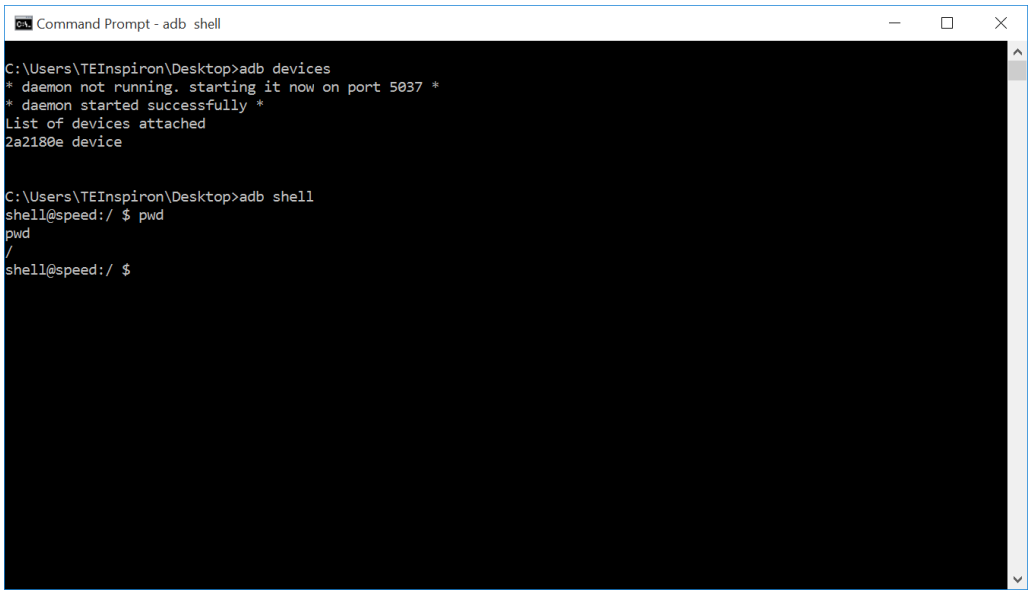



**Figure 72 – The command "pwd" prints the current working directory.**

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’ve changed directories) you’ll see your new location within the file system:

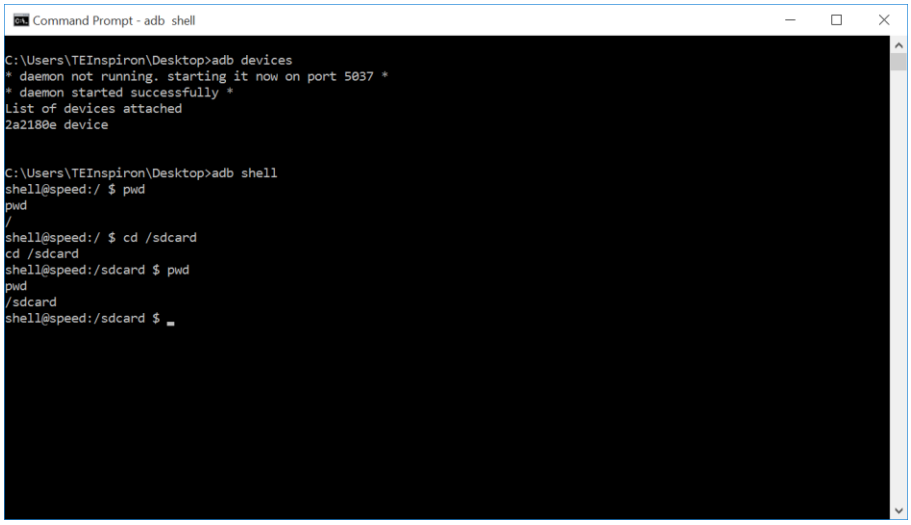



**Figure 73 – Entering in “cd /sdcard” will change the working directory. Entering in “pwd” will print the new working directory.**

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’ll 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:

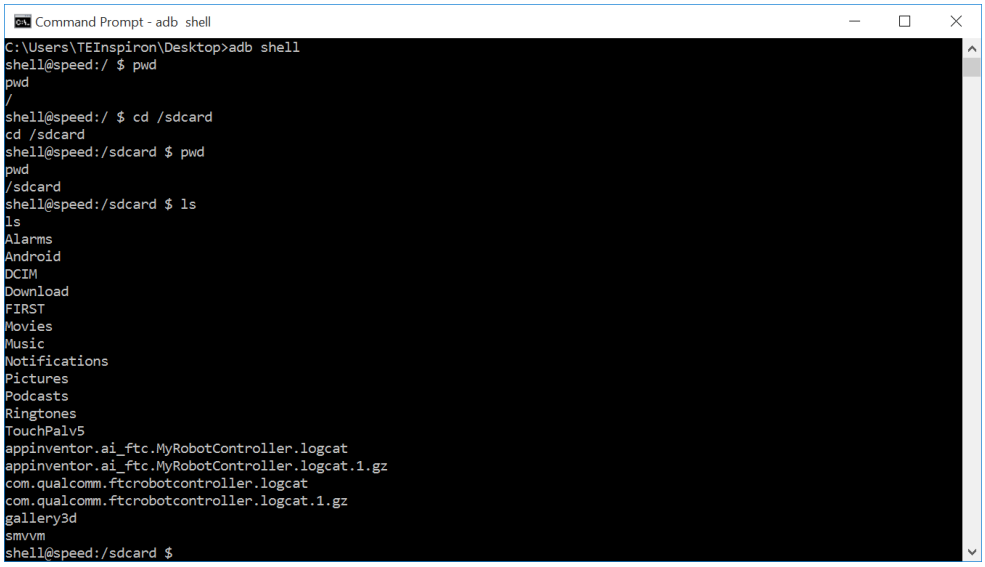



**Figure 74 – The command “ls” will list the folders and files in the current directory.**

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

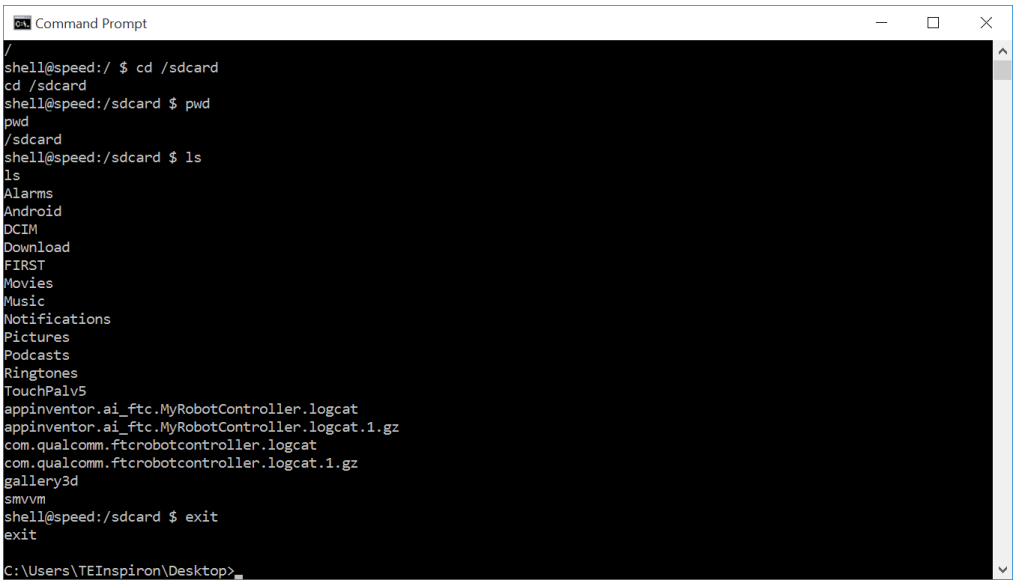



**Figure 75 – The command "exit" will exit you from the phone's shell and return you to your computer's command line.**

### *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 (/sdcard/com.qualcomm.ftcRobotcontroller.logcat) from the Android device and to the local file system (c:\users\TEInspiron\Desktop\rc_log.txt).

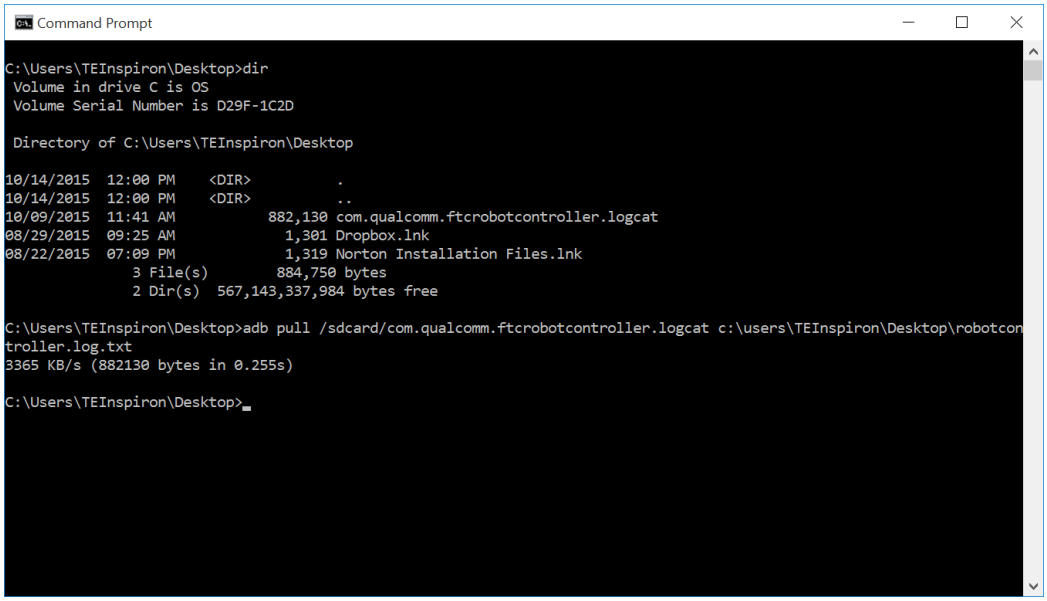



**Figure 76 – You can use the "adb pull" command to copy a file from the phone onto your local hard drive.**

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[9] 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).

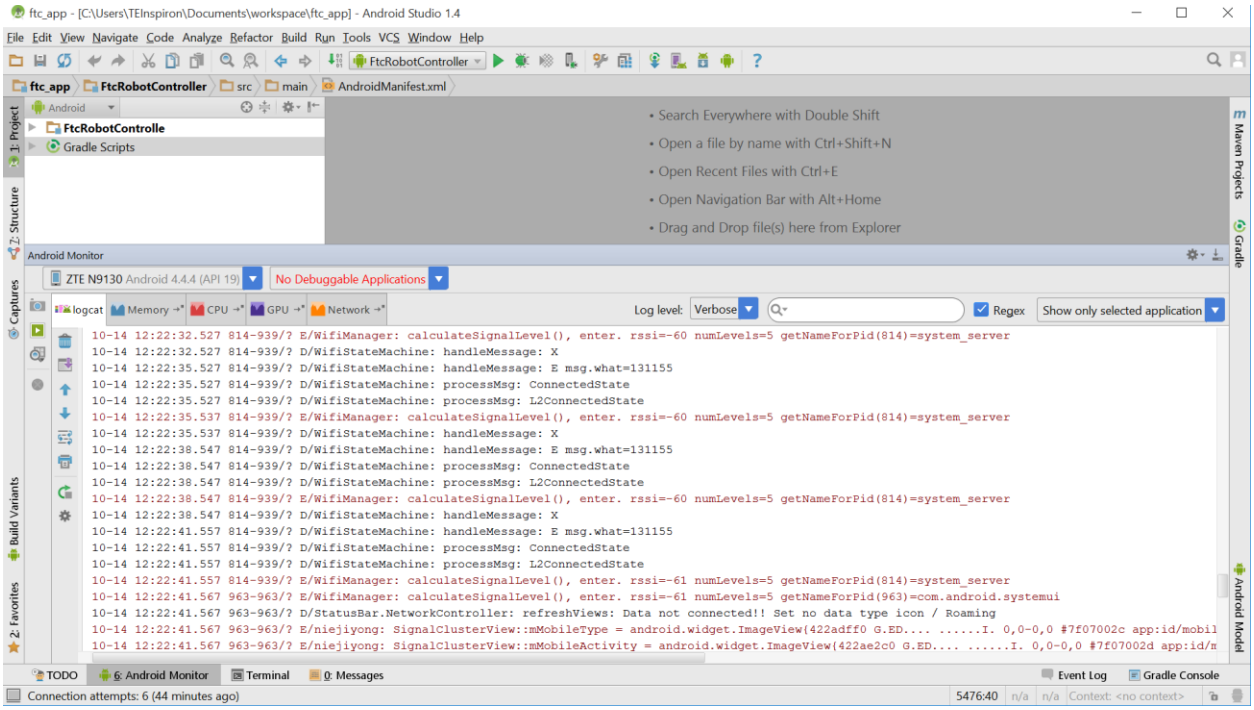



**Figure 77 – You can view (and filter) logcat statements through Android Studio.**

### *Creating Your Own Log Statements within an Op Mode*

It is possible (and often helpful) to insert your own log statements within an op mode 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)
```

[9] Refer to http://developer.android.com/tools/help/adb.html#wireless to see information about how to use ADB wirelessly.

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 embedded these methods within your op mode 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 Op Mode*

The following text is an example op mode that shows how to use the DbgLog class to embed log statements within your op mode:

```
package com.qualcomm.ftcRobotcontroller.opmodes;

import com.qualcomm.ftccommon.DbgLog;
import com.qualcomm.Robotcore.Eventloop.opmode.LinearOpMode;
import com.qualcomm.Robotcore.robocol.Telemetry;

/**
 * Created by TEInspiron on 10/14/2015.'
 * This op mode demonstrates how to use log statements within an Op Mode.
 */
public class MyLogDemo extends LinearOpMode {

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

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

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

        waitForStart();

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

This linear op mode example shows how to use the *DbgLog.msg* 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 op mode listed in section 0 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 78). The Create New Logcat Filter window should appear (see Figure 79). 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[10] that is used to filter statements. In Figure 79 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.

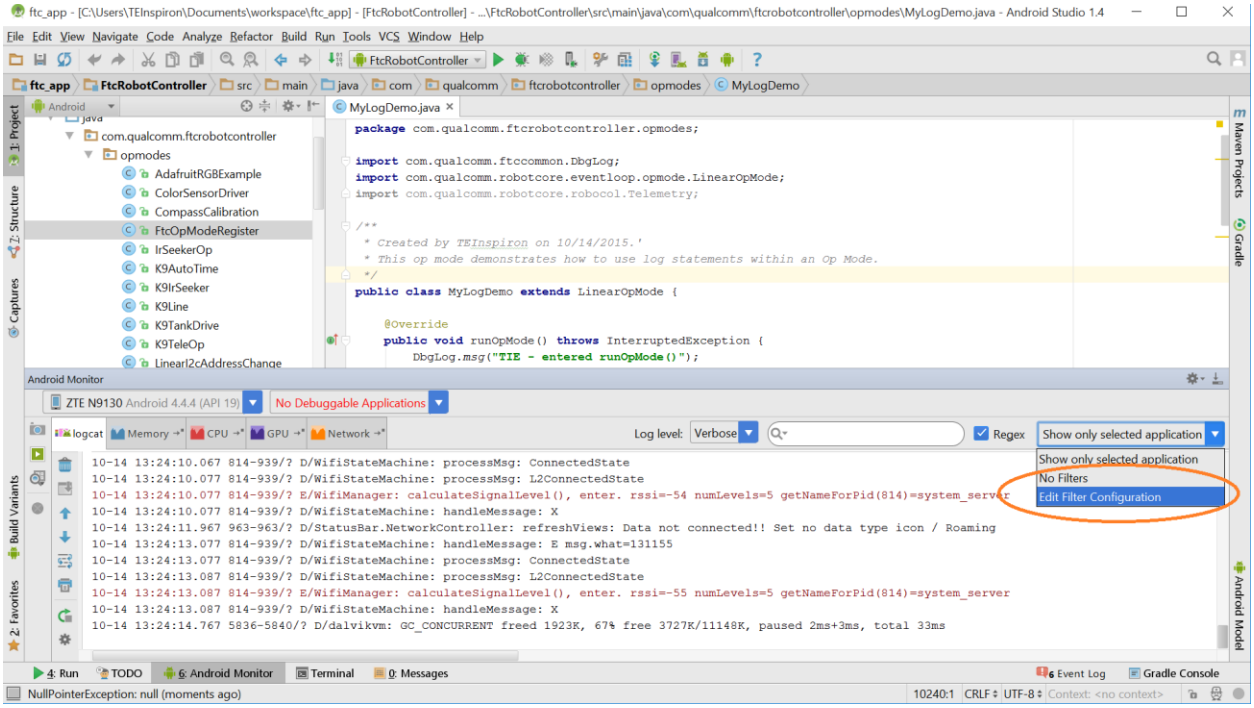



**Figure 78 – Select Edit Filter Configuration to create a new filter**

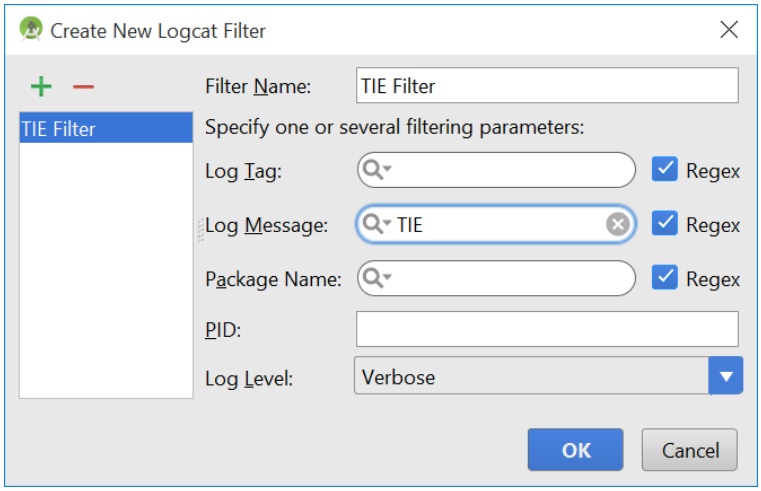



**Figure 79 – Specify the Filter Name and a regular expression that you want to use for your filter (in this case "TIE").**

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

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 op mode is currently running, you can see them in real time.

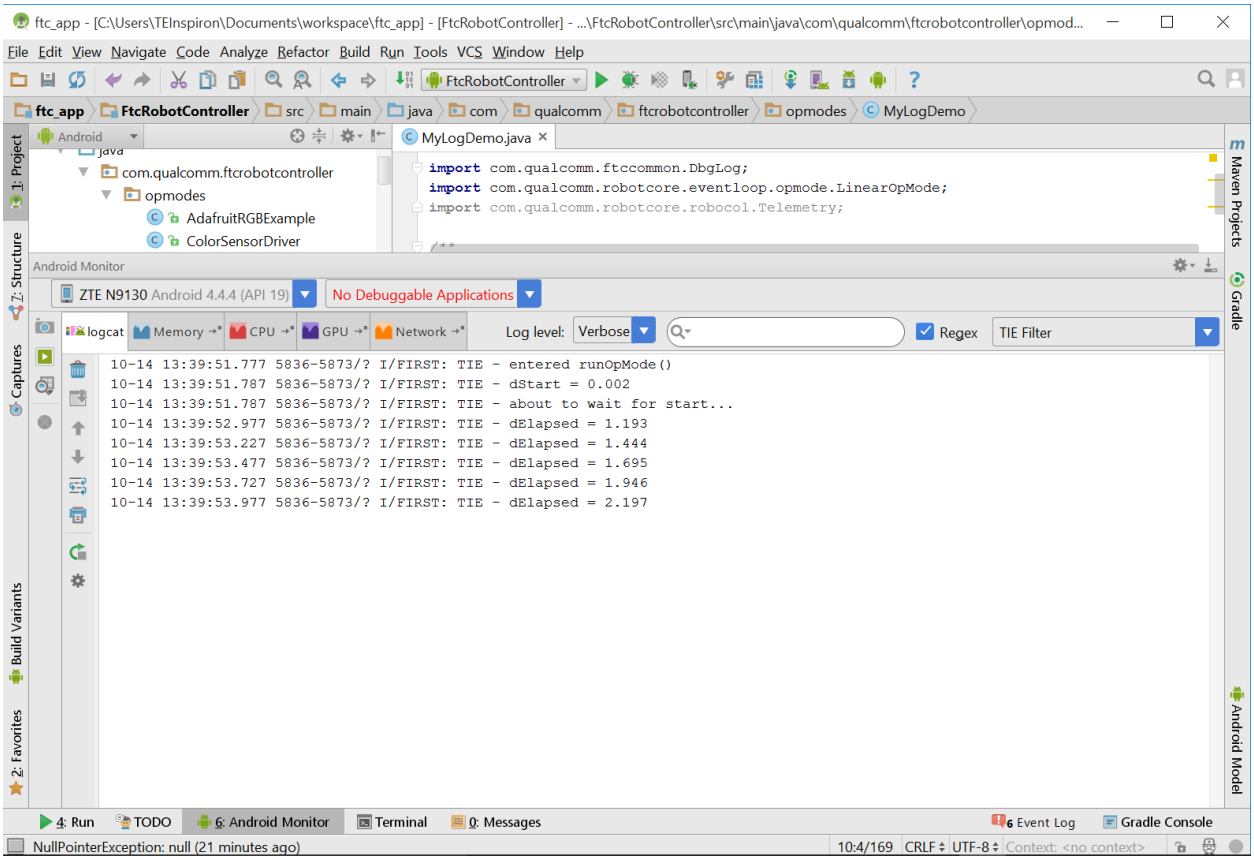



**Figure 80 – If the op mode is running you can see your log statements in real time with a USB or wireless ADB connection.**

## Appendix B: Resources & Support

***Game Forum Q&A*** - *http://ftcforum.usfirst.org/forum.php*

***FIRST Tech Challenge Game Manuals – Part I and II -*** *http://www.firstinspires.org/node/4271*

***FIRST Headquarters Support***

Phone: 603-666-3906

Email: FTCTeams@firstinspires.org

***FIRSTINSPIRES.ORG***

*FIRST* Tech Challenge Page – For everything *FIRST* Tech Challenge.

*FIRST* Tech Challenge Volunteer Resources – To access public Volunteer Manuals.

*FIRST* Tech Challenge Event Schedule – Find *FIRST* Tech Challenge events in your area.

***FIRST Tech Challenge Social Media***

*FIRST* Tech Challenge Twitter Feed - If you are on Twitter, follow the *FIRST* Tech Challenge twitter feed for news updates.

*FIRST* Tech Challenge Facebook page - If you are on Facebook, follow the *FIRST* Tech Challenge page for news updates.

*FIRST* Tech Challenge YouTube Channel – Contains training videos, Game animations, news clips, and more.

*FIRST* Tech Challenge Blog – Weekly articles for the *FIRST* Tech Challenge community, including Outstanding Volunteer Recognition!

*FIRST* Tech Challenge Team Email Blasts – contain the most recent *FIRST* Tech Challenge news for Teams.

*FIRST* Tech Challenge Google+ community - If you are on Google+, follow the *FIRST* Tech Challenge community for news updates.

***Product Support***

*FIRST* will handle questions about team registration, grants, events and partners. Pitsco will handle questions about ordering, payment and delivery of competition sets and materials.

- U.S. & Canada: 800-835-0686 FREE
- Outside U.S. and Canada: 620-231-0100
- U.S. Fax: 800-533-8104

## Feedback

We strive to create support materials that are the best they can be. If you have feedback regarding this manual, please email ftcteams@firstinspires.org. Thank you!