

FTC Tech Tip: Log Files

1 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.

2 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 FTC 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).

3 The FTC 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 parameter <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 FTC Driver Station app, the path to the log file is as follows,

```
/sdcard/com.qualcomm.ftcdriverstation.logcat
```

4 Viewing the FTC Robot Controller Log File

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

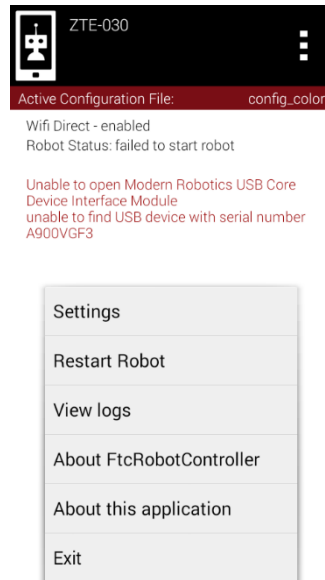


Figure 1 – Click on the three dots in the upper right hand portion of the screen then select View logs from the menu.

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

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

Figure 2 – 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.

5 Finding the Log Files

5.1 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 3 - 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.

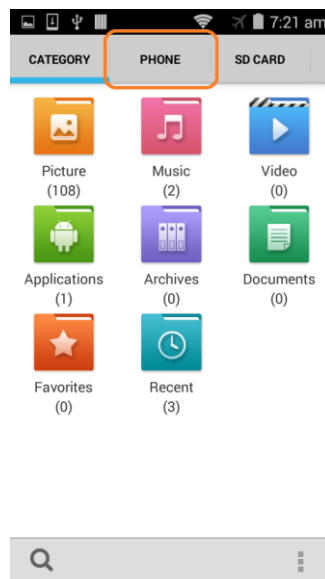


Figure 4 – Click on the Phone tab to browse the contents of your phone.

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:

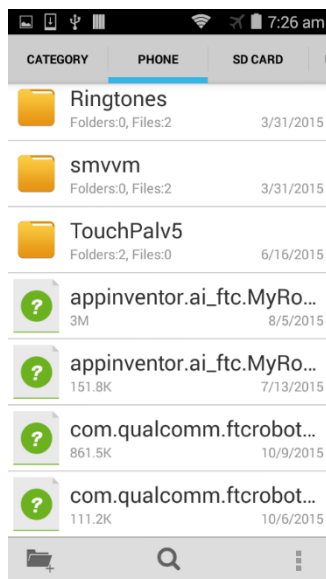


Figure 5 - If you scroll towards the bottom you will see the log files.

In Figure 5 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 FTC 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.

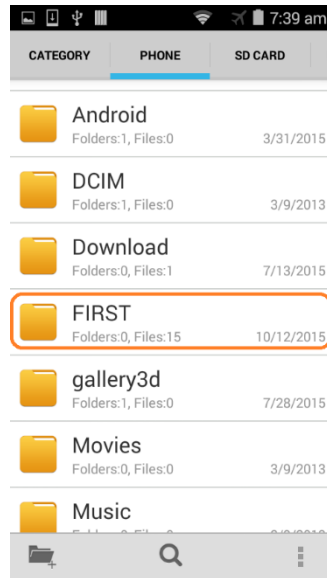


Figure 6 – There is a FIRST subdirectory that contains FIRST-related configuration and other files.

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.

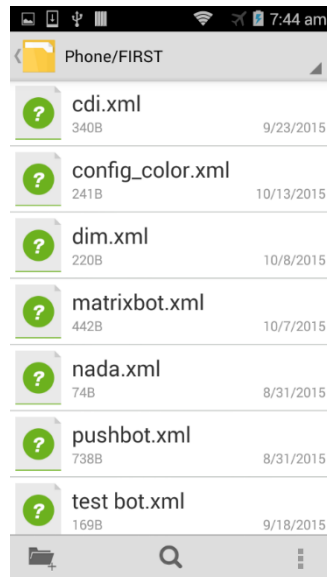


Figure 7 – The FIRST folder contains the robot configuration files (.xml) plus some hidden files (not visible in this image).

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 FTC apps for storing other data.

5.2 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 **FTC 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:

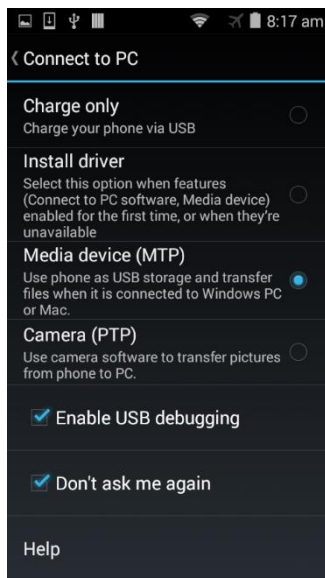


Figure 8 - Make sure your phone is in Media device mode.

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:

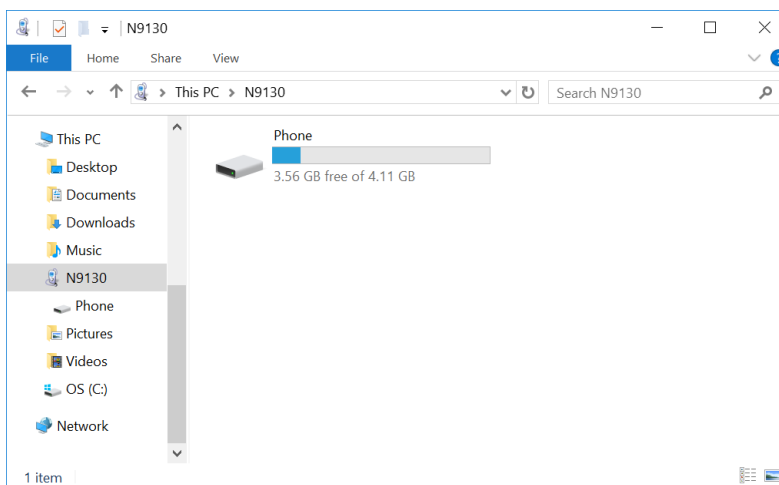


Figure 9 –The phone should appear as a media device connected to your PC.

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

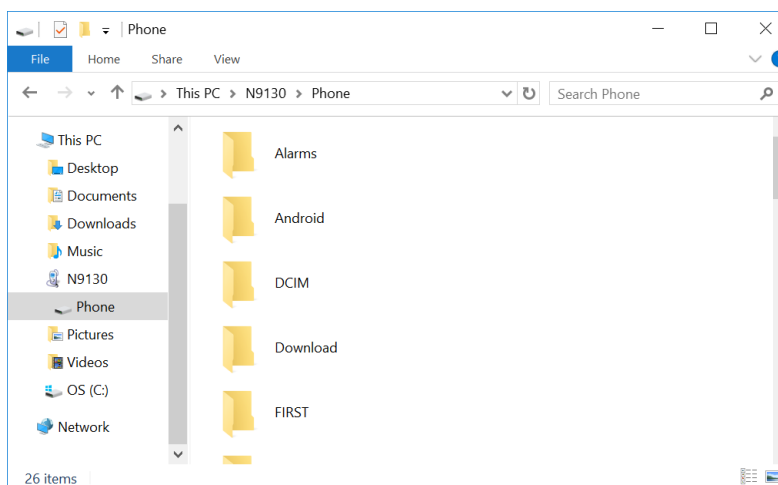


Figure 10 – You can browse the main directory of the phone.

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:

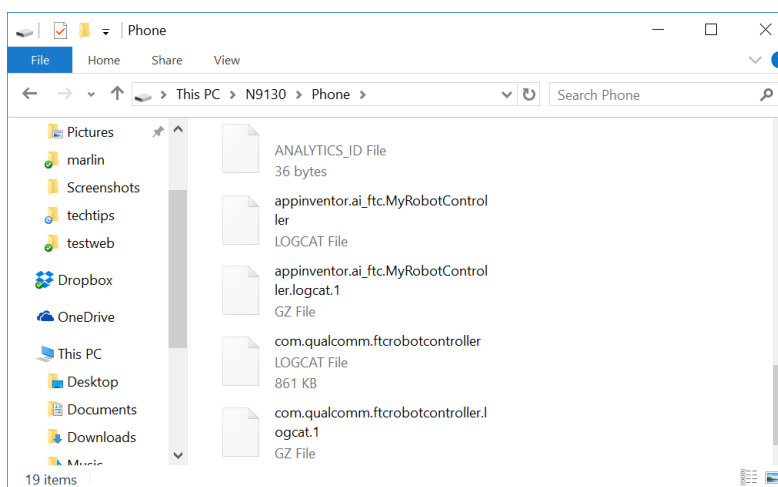


Figure 11 – The log files should be visible in this directory.

In Figure 11 you can see the same log files that you saw in Figure 5. 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.

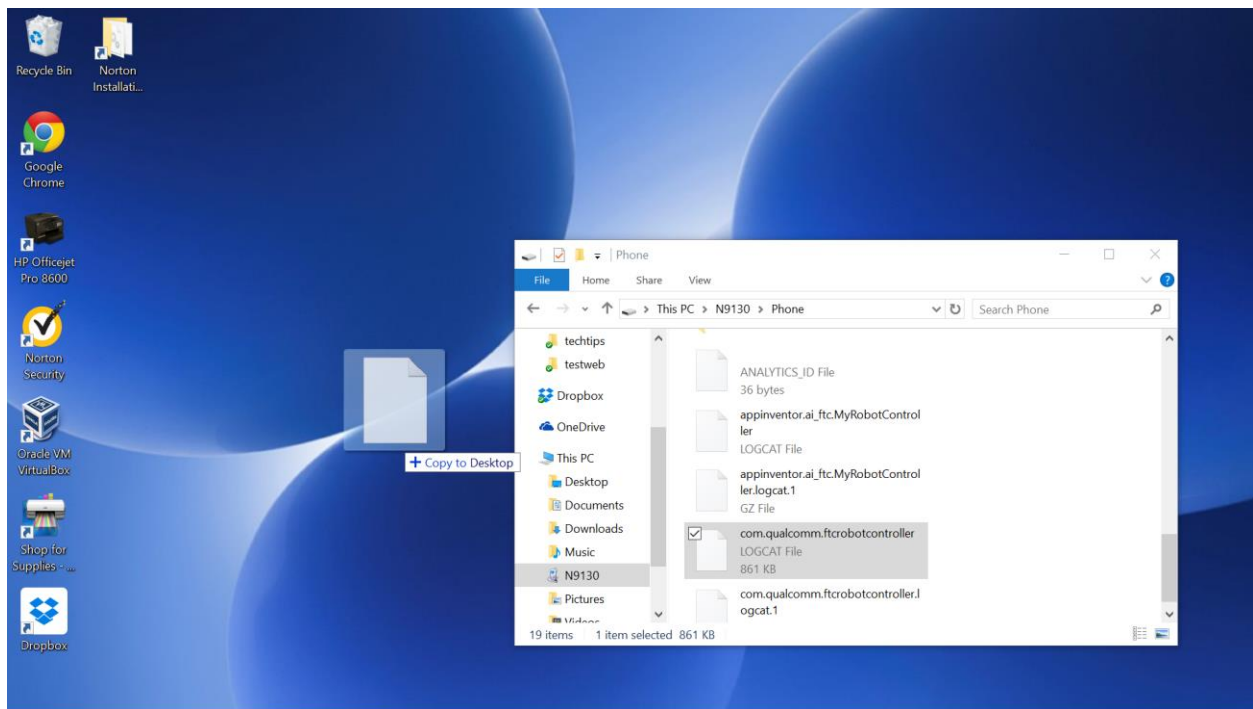


Figure 12 – You can make a copy of a log file by dragging and drop the file to your personal computer.

5.3 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 13 – 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.

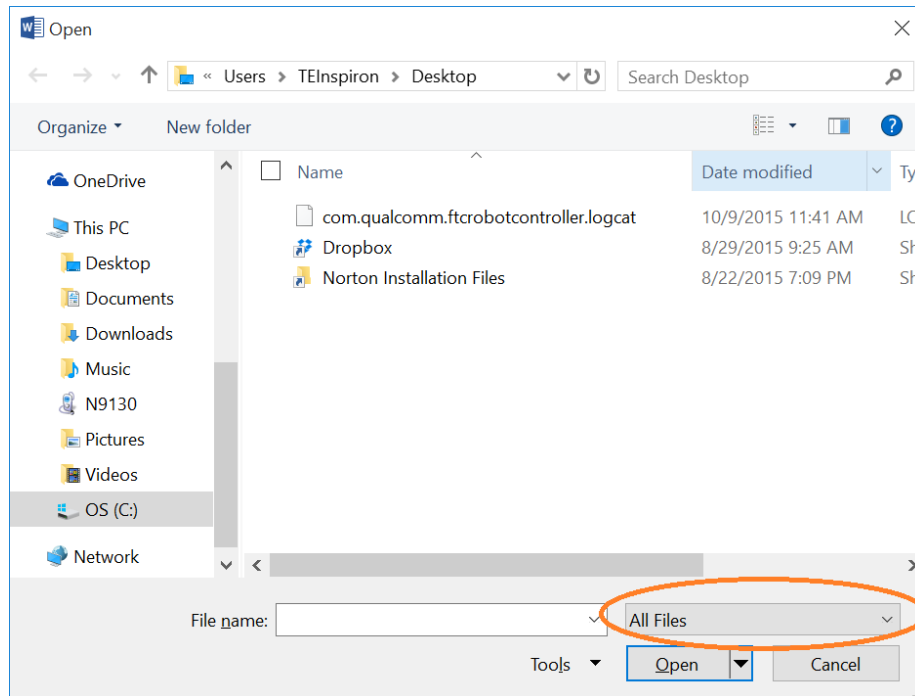


Figure 14 – Make sure the file filter is set to “All Files” when you browse to find your log file.

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

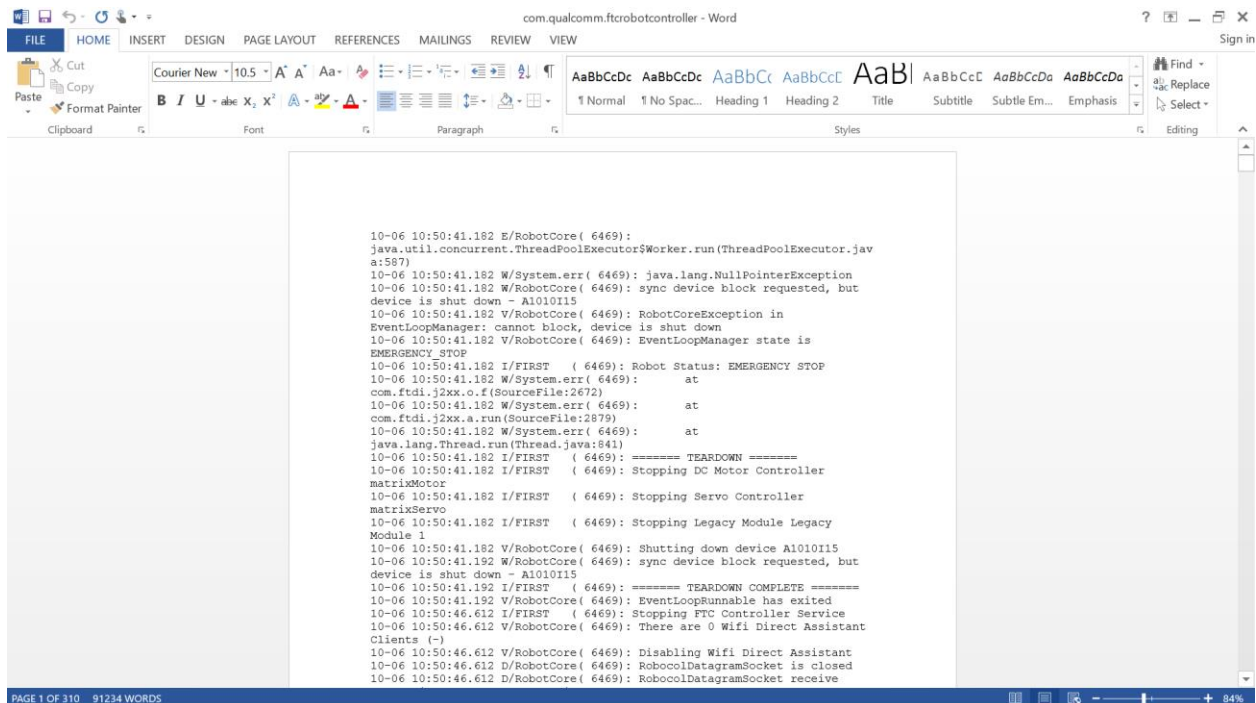


Figure 15 – Using Microsoft Word to view the log file makes it easier to read the contents and search for specific text strings.

5.4 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.

6 Using the Android Debug Bridge for Troubleshooting

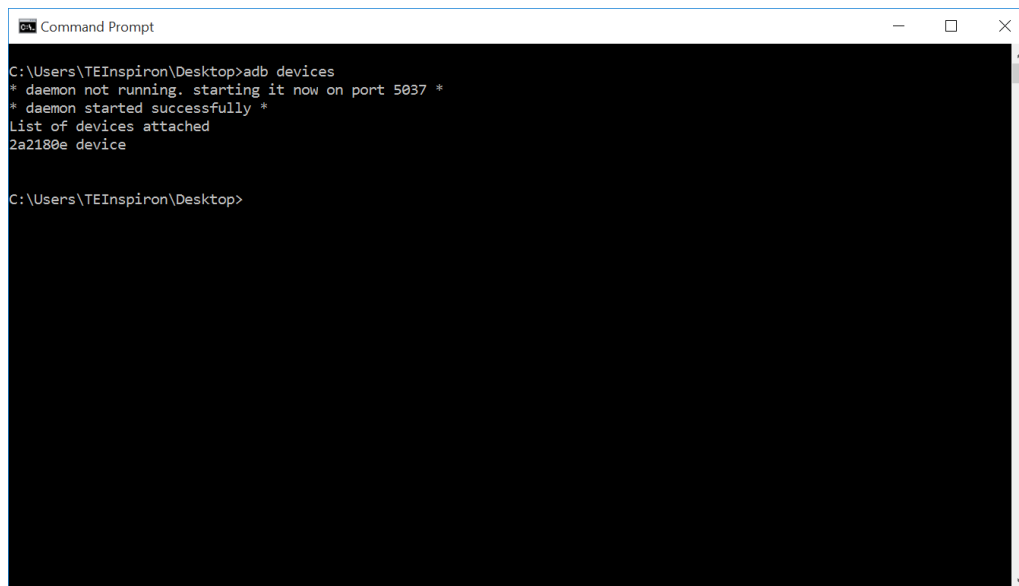
The Android Debug Bridge (ADB) is a utility program that is include with the Android Software Development Kit (SDK) platform tools. ADB is a program that you can invoke from a command line. It is very a very helpful utility. In order to use the ADB utility you will need to have the Android SDK platform-tools 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).

6.1 “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:

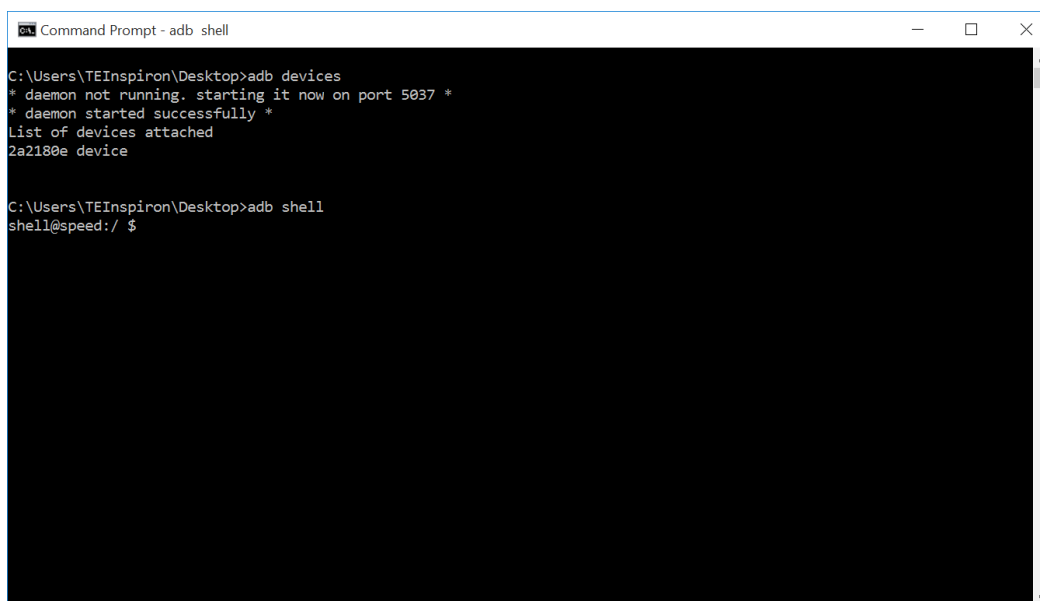


```
Command Prompt
C:\Users\TEInspiron\Desktop>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
2a2180e device

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

Figure 16 – 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:



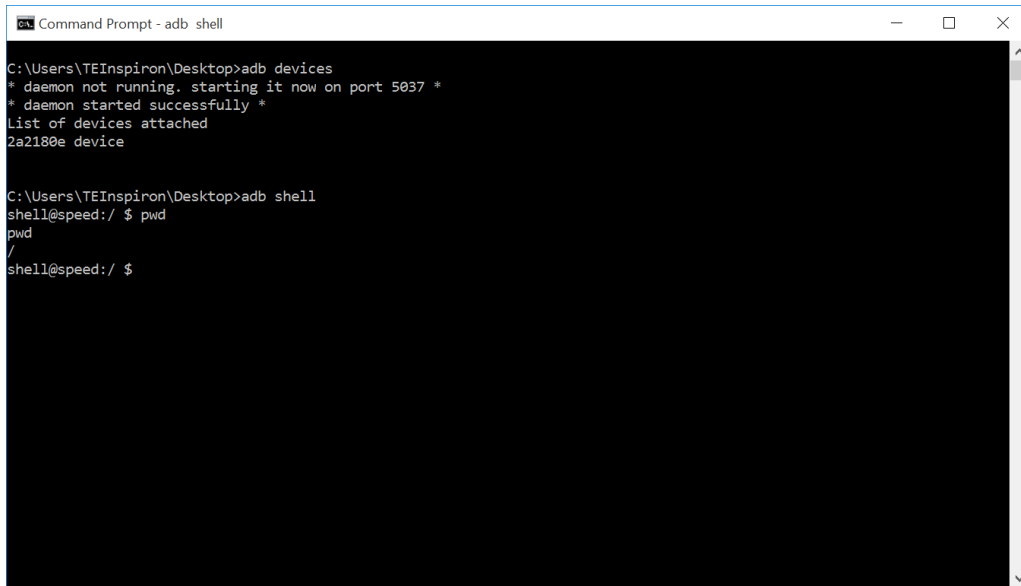
```
Command Prompt - adb shell
C:\Users\TEInspiron\Desktop>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
2a2180e device

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

Figure 17 – Type in “adb shell” to create a terminal session with your Android device.

If you look at Figure 17 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:



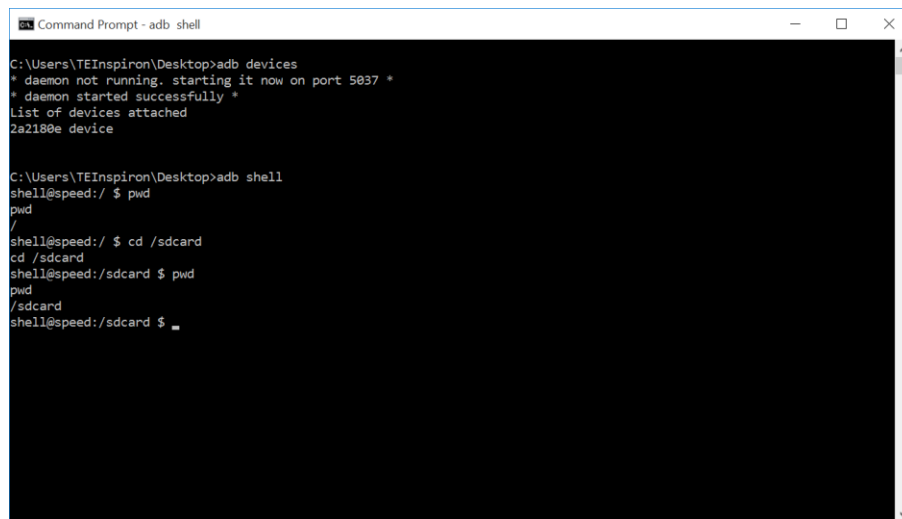
```
Command Prompt - adb shell

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

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

Figure 18 – 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:



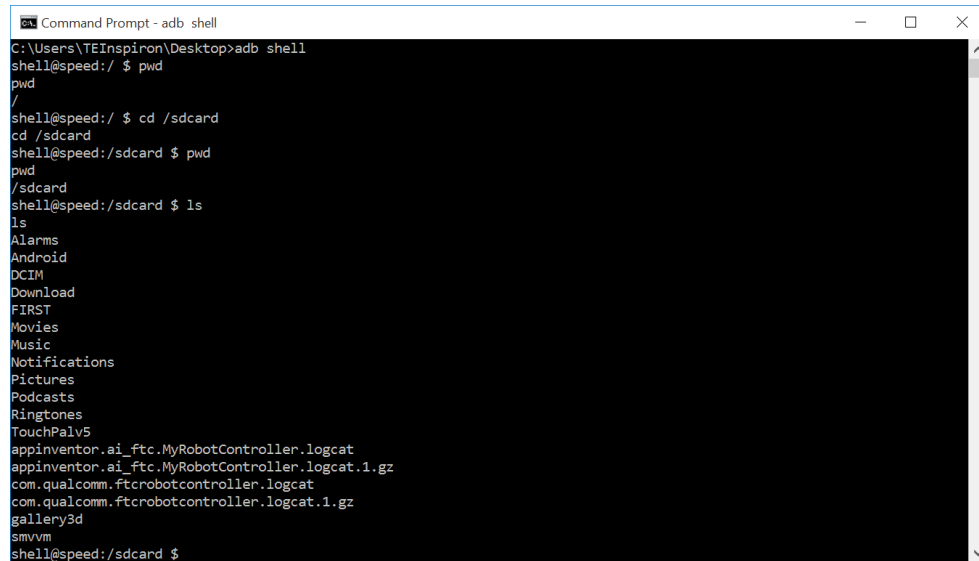
```
Command Prompt - adb shell

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

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

Figure 19 – Entering in “cd /sdcard” will change the working directory. Entering in “pwd” will print the new working directory.

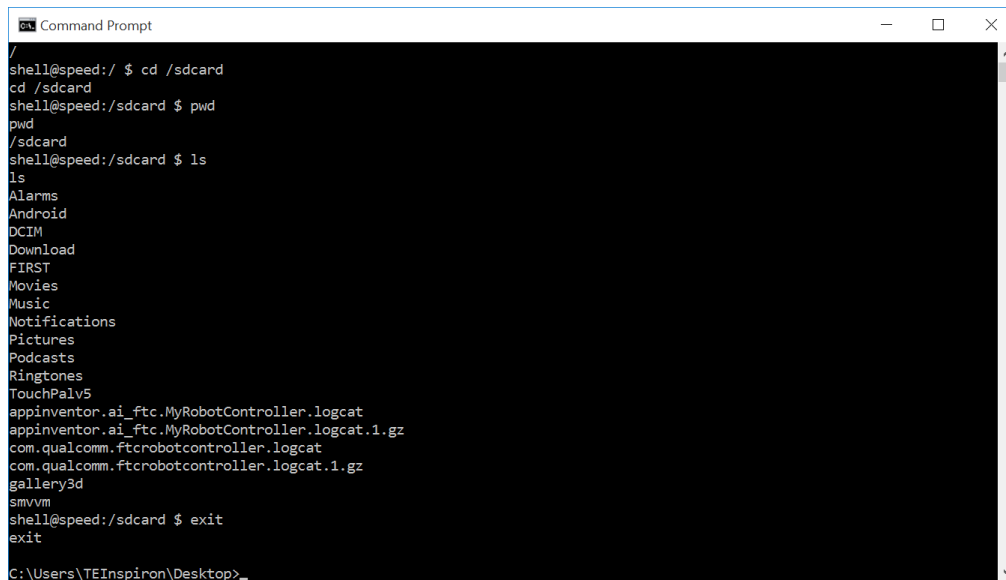
If you type “ls” at the command prompt the phone will list the contents of the current directory. If you look at the directories and files you’ll see that they match the folders and files that you saw using the Android device’s File Manager app, or that you would see using the Windows File Explorer application:



```
Command Prompt - adb shell
C:\Users\TEInspiron\Desktop>adb shell
shell@speed:/ $ pwd
/
shell@speed:/ $ cd /sdcard
cd /sdcard
shell@speed:/sdcard $ pwd
/sdcard
shell@speed:/sdcard $ ls
ls
Alarms
Android
DCIM
Download
FIRST
Movies
Music
Notifications
Pictures
Podcasts
Ringtones
TouchPalv5
appinventor.ai_ftc.MyRobotController.logcat
appinventor.ai_ftc.MyRobotController.logcat.1.gz
com.qualcomm.ftcrobotcontroller.logcat
com.qualcomm.ftcrobotcontroller.logcat.1.gz
gallery3d
smvwm
shell@speed:/sdcard $
```

Figure 20 – 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”:



```
Command Prompt
/
shell@speed:/ $ cd /sdcard
cd /sdcard
shell@speed:/sdcard $ pwd
/sdcard
shell@speed:/sdcard $ ls
ls
Alarms
Android
DCIM
Download
FIRST
Movies
Music
Notifications
Pictures
Podcasts
Ringtones
TouchPalv5
appinventor.ai_ftc.MyRobotController.logcat
appinventor.ai_ftc.MyRobotController.logcat.1.gz
com.qualcomm.ftcrobotcontroller.logcat
com.qualcomm.ftcrobotcontroller.logcat.1.gz
gallery3d
smvwm
shell@speed:/sdcard $ exit
exit
C:\Users\TEInspiron\Desktop>
```

Figure 21 – The command “exit” will exit you from the phone’s shell and return you to your computer’s command line.

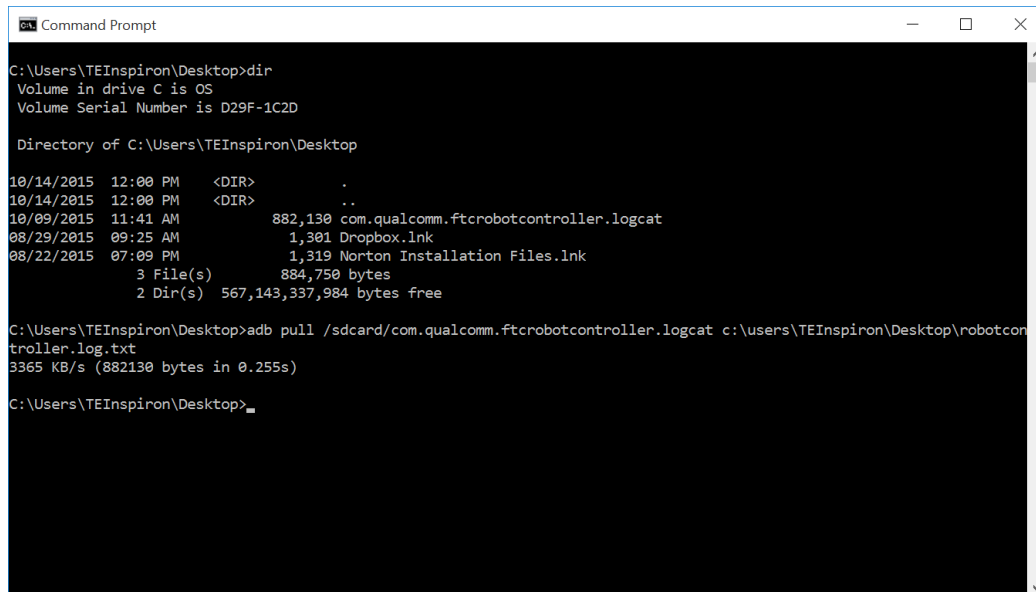
6.2 Pulling a File from the Android Device

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).



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

Directory of C:\Users\TEInspiron\Desktop

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

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

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

Figure 22 – 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.

7 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 ADB¹ 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).

¹ Refer to <http://developer.android.com/tools/help/adb.html#wireless> to see information about how to use ADB wirelessly.

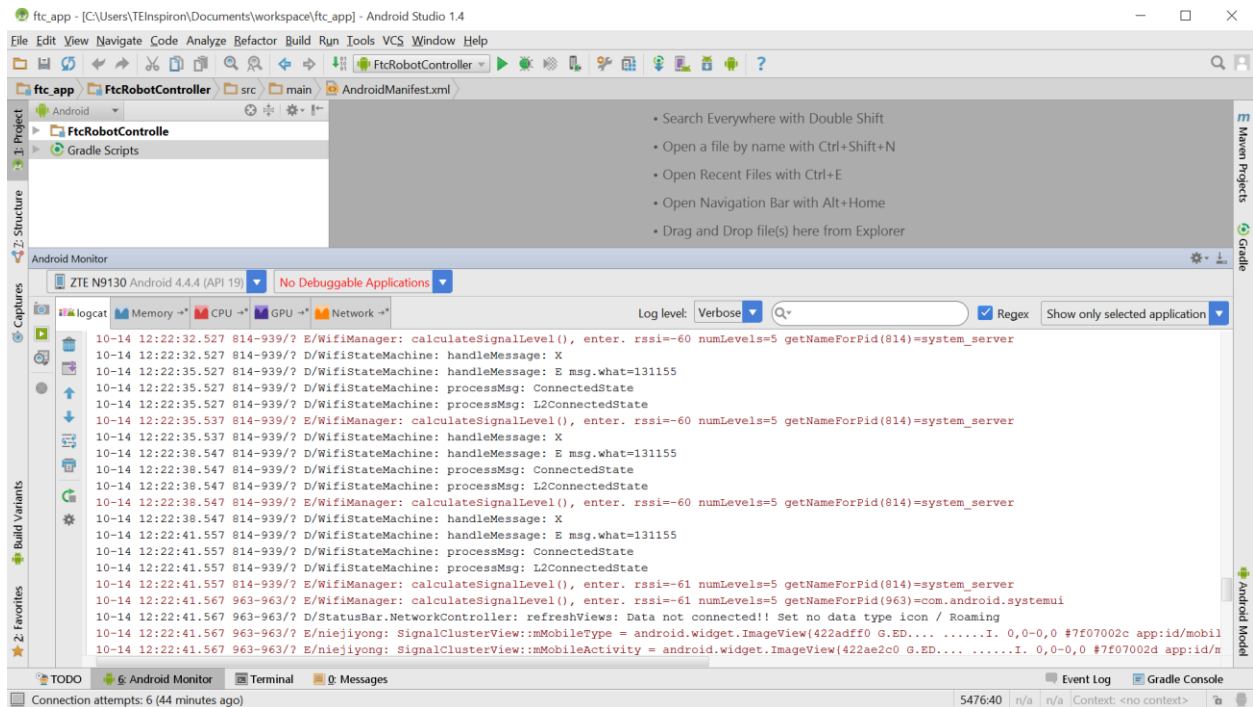


Figure 23 – You can view (and filter) logcat statements through Android Studio.

8 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 FTC SDK contains a class called `DbgLog` that has two static methods that can be used to log messages to the log file:

```
DbgLog.err(String message)

DbgLog.msg(String message)
```

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

You can 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.

8.1 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.robotcol.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.

8.2 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 24). The Create New Logcat Filter window should appear (see Figure 25). 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 expression² that is used to filter statements. In Figure 25 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.

² Visit https://en.wikipedia.org/wiki/Regular_expression for more information about regular expressions.

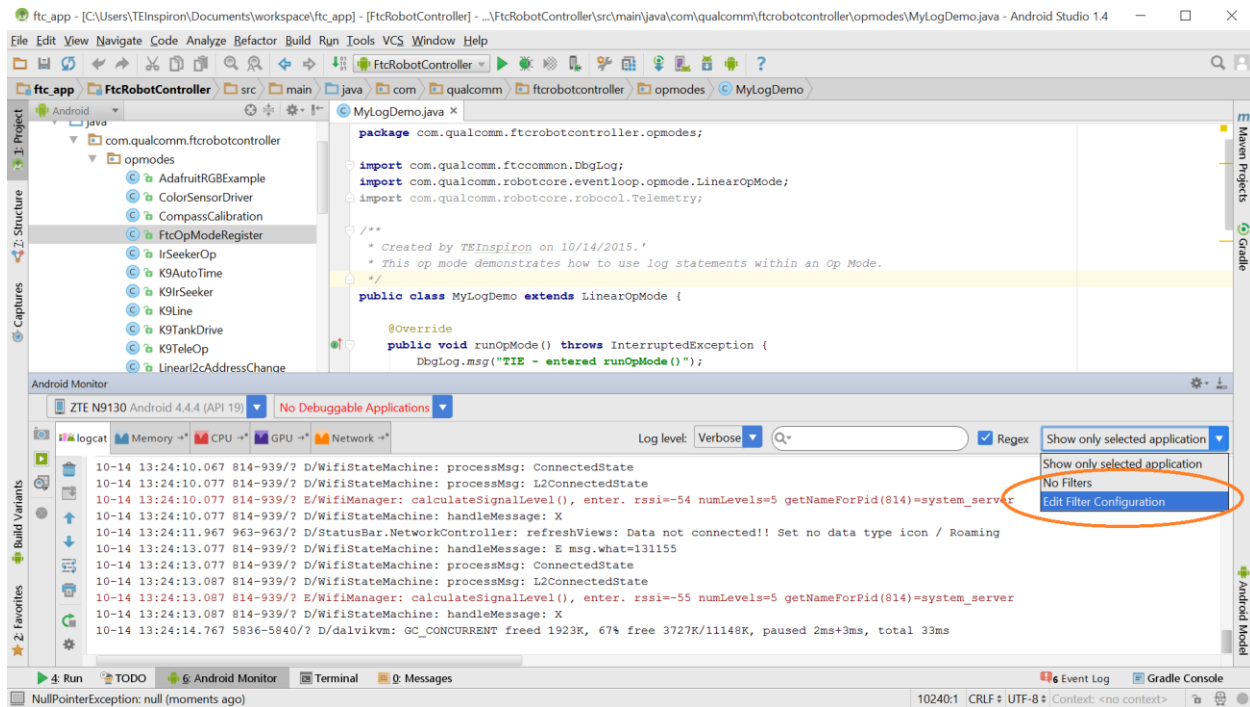


Figure 24 – Select Edit Filter Configuration to create a new filter

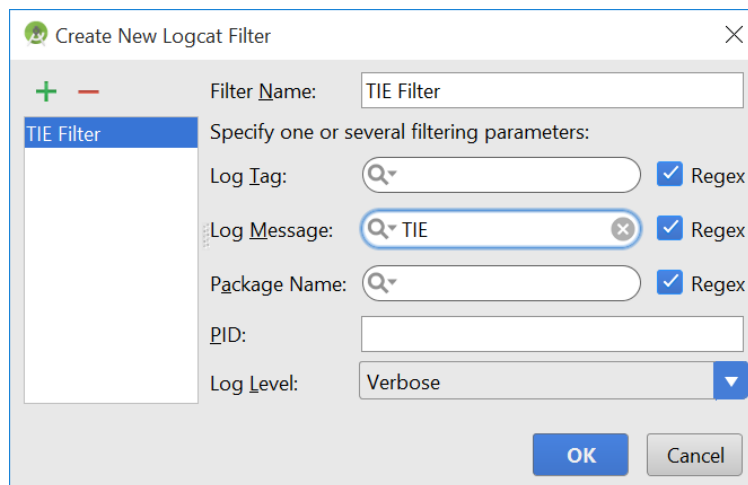


Figure 25 – Specify the Filter Name and a regular expression that you want to use for your filter (in this case “TIE”).

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

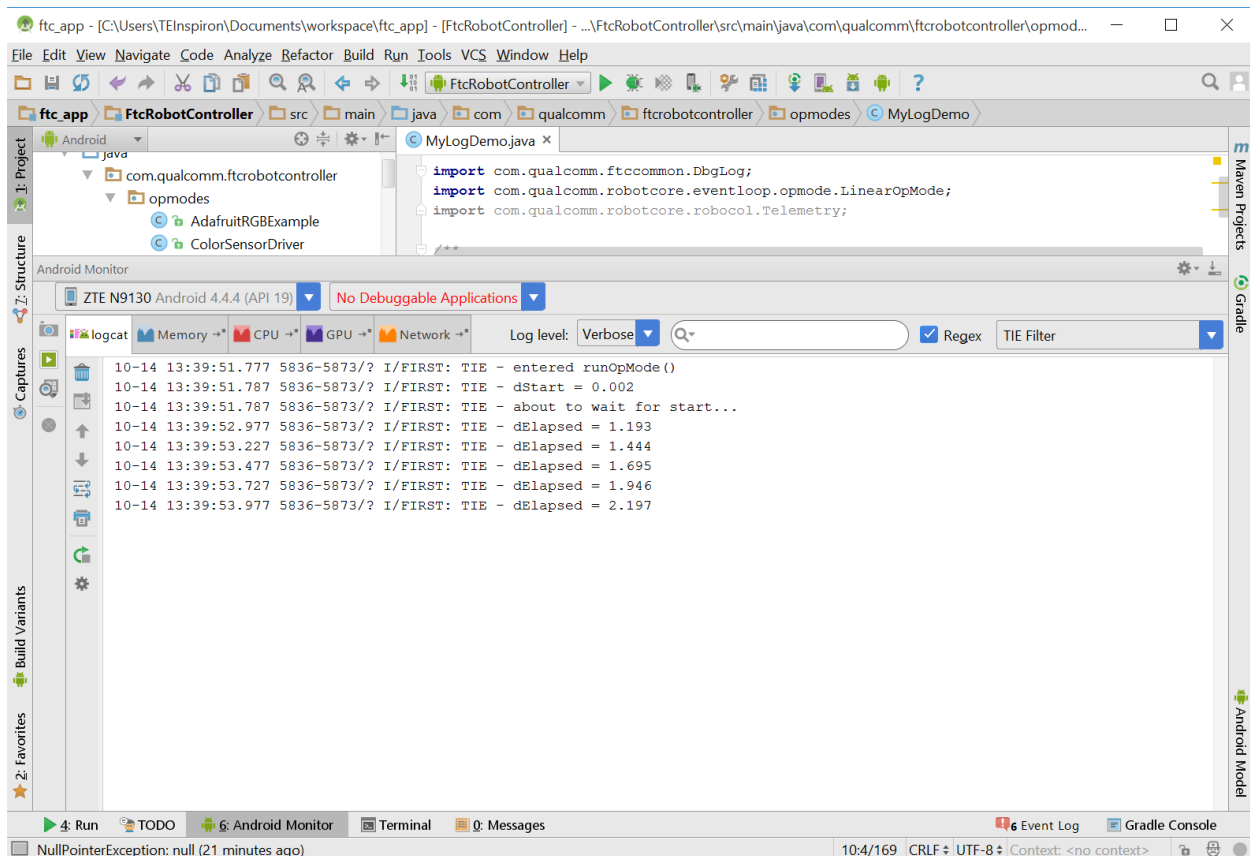


Figure 26 – If the op mode is running you can see your log statements in real time with a USB or wireless ADB connection.