

FTC Training Manual

Using the FTC SDK

FIRST Tech Challenge

9/8/2016



This document contains training material to introduce students and mentors how to write programs for the FIRST Tech Challenge Robot Controller platform.

Contents

1	Introduction	3
2	Scope of Material.....	3
3	Demo Robot	3
4	Point-to-Point Control System	4
4.1	Driver Station	4
4.2	Robot Controller	5
5	The Android Operating System.....	5
5.1	What is Android?	5
5.2	Android Studio	5
5.3	MIT App Inventor	6
5.4	Java.....	6
6	Installing Android Studio.....	7
6.1	Android Studio Website.....	7
6.2	System Requirements	7
6.3	Installing the Java Development Kit.....	7
6.4	Downloading and Installing Android Studio	8
7	Downloading and Installing the FTC SDK	9
7.1	Downloading the Android Studio Project Folder	10
7.2	Extracting the Contents of the Archived Project File.....	12
7.3	Importing the FTC Project into Android Studio	13
8	Using Your Robot	15
8.1	Enabling Developer Options	16
8.2	Copying an Example Op Mode.....	17
8.3	Registering Your Op Mode.....	20
8.4	Building the Robot Controller App.....	21
8.5	Configuring the Robot Controller.....	23
8.6	Installing the FTC Driver Station App	26
8.7	Pairing the Driver Station to the Robot Controller	30
8.8	Using a Gamepad	32
8.9	Selecting and Running an Op Mode.....	33
8.10	Swapping USB Devices in the Configuration File	35

9	For More Information...	38
Appendix A:	Note about USB Ports and Cables	0
	USB Host & Client Modes.....	0
	USB Micro Type B.....	0
	USB Mini Type B.....	1
	Micro USB OTG Adapter	1
Appendix B:	Preparing the ZTE Speed for use with FTC.....	2
	Removing the SIM card from the ZTE Speed	2
	Enabling Developer Options on your Android Device	10
	Enabling USB Debugging	12
	Installing the USB Driver for your Device.....	13
	Renaming the ZTE Speed	19

1 Introduction

The *FIRST* Tech Challenge (FTC) uses an Android-based control system for its competition robots. This document provides basic information on how to install, configure and use the FTC Software Development Kit (SDK) to customize the behavior of an FTC competition robot.

2 Scope of Material

This document only provides a very basic introduction to Android programming and how it relates to the FTC Robot Controller platform.

Participants are encouraged to learn more about Java and Android development for other resources, such as the official Android Developer's website:

<http://developer.android.com/index.html>

The Android Developer's website has a series of Android development tutorials (including video-based interactive training):

<http://developer.android.com/training/index.html>

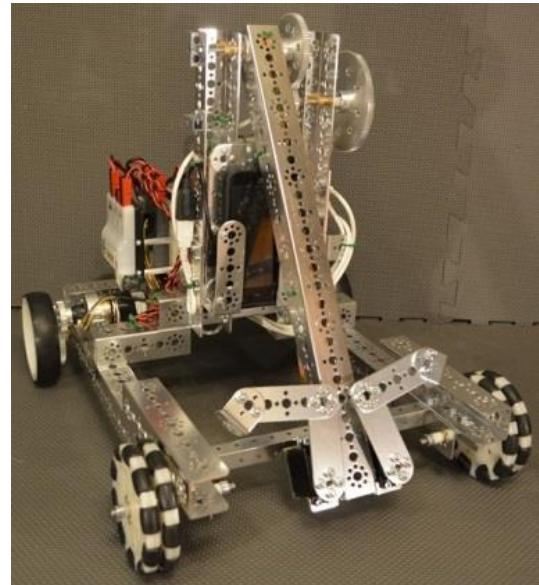
There is an excellent free tutorial from Oracle that teaches basic and advanced Java programming:

<http://docs.oracle.com/javase/tutorial/>

3 Demo Robot

The training material in this document assumes that students will have access to a demo robot. This guide is applicable to most robots, but the specific pictures and examples are drawn from the PushBot design. This is a mid-sized "Competition Ready" robot, that utilizes a hybrid fixed/omni drive system and a motor/gear driven arm with servo activated grippers.

Figure 2 - The PushBot is a competition-ready bot that can be built from the standard FTC kit of parts.



There is a detailed build guide for the PushBot located on the FIRST website. Visit the Robot Building Resources web page (<http://www.firstinspires.org/resource-library/ftc/robot-building-resources>), and the **PushBot Build Guide** is listed under the Build Guides category.

4 Point-to-Point Control System

The next gen platform is a *point-to-point* system. This system has two main components, the *Driver Station* and the *Robot Controller*. The Driver Station communicates wirelessly with the Robot Controller.



Figure 3 - The new FTC platform is a point-to-point solution.

4.1 Driver Station

The Driver Station is the component that the teams use to provide input (from a pair of gamepads) to the Robot Controller. The Driver Station can also display telemetry messages (status information, motor speed, sensor data, etc.) from the Robot Controller.



Figure 1 - Driver Station main screen.

4.2 Robot Controller

The Robot Controller is the “brains” of the robot. It handles communications with the Driver Station. It processes sensor data and user commands and sends instructions to the Motor and Servo Controllers to make the robot move.

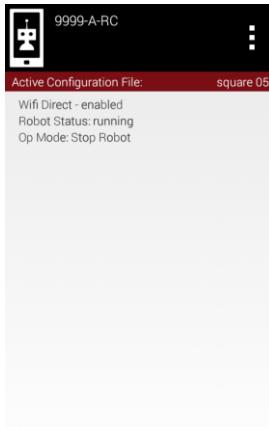


Figure 2 - Robot Controller main screen.

5 The Android Operating System

5.1 What is Android?

The FTC Platform uses Android smartphones to control a robot. Android is the *operating system* that runs on these handheld devices. Similar to a laptop that has Microsoft Windows or MacOS as its operating system, a tablet or smartphone has its own operating system that manages the device’s hardware and software components.

Google produces, develops and maintains the Android code. This source code is available to the public under an open source license. Google provides free developer tools that can be used to write applications or “apps” for the Android platform.



5.2 Android Studio

The official development tool for Android development is known as *Android Studio*. The Driver Station and Robot Controller run special apps for the FTC competition. These apps are created Android Studio.

Android Studio is known as an Integrated Development Environment (IDE). Android Studio is a software package that you install onto a computer or laptop. It has a suite of tools, such as a text editor, debugger, and other tools to help author, build and install apps for the Android operating system.

App development might seem intimidating at first. However, for the FTC robots, the process has been simplified. The FTC SDK includes a framework that makes it easier for a novice to program their robot. This framework takes care of much of the more complex programming tasks. The student or mentor

can focus on programming the robot behavior and not have to worry about developing the framework of the Android app.

5.3 MIT App Inventor

The MIT App Inventor is an alternate platform to develop code for an FTC robot. The App Inventor is a user-friendly design tool to create Android apps using a visual development environment. App Inventor has been modified to support the FIRST Tech Challenge hardware. Teams are able to use the App Inventor to create apps for their competition robots.

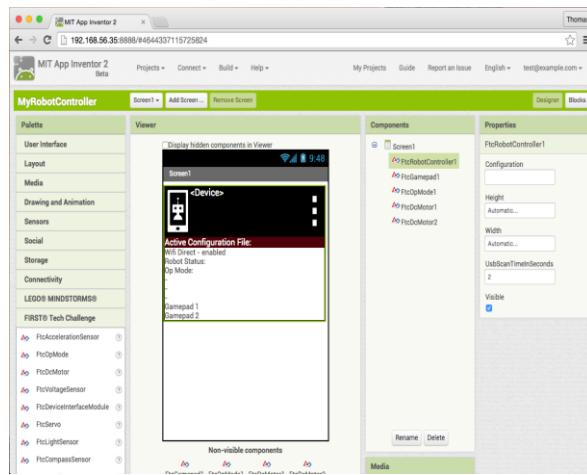


Figure 7 - The App Inventor is a browser-based application that makes it easy to create custom Android apps.

Since this manual focuses specifically on Java development, the App Inventor will not be covered here in any detail. For detailed information about using the MIT App Inventor for the *FIRST* Tech Challenge, please visit the following website:

<https://frc-events.firstinspires.org/ftcimages/2016>

5.4 Java

Java is a popular text-based, object-oriented programming language. Android apps are written using the Java language. The programs that we will be using in this tutorial require a basic knowledge of Java. Unfortunately, the scope of this document does not allow for a detailed examination of the Java programming language. However, the Oracle Corporation maintains an excellent, free online Java tutorial:

<http://docs.oracle.com/javase/tutorial/>

Students are encouraged to review the online Java tutorial before they attempt to try the exercises in this training manual. It will be helpful if students have a basic knowledge of Java to complete the exercises in this manual. Students do not have to be Java experts, but should understand the basic syntax of java and be familiar with concepts such as classes, members, methods and program structure.

6 Installing Android Studio

IMPORTANT NOTE!!! This training manual contains instructions on how to install the Android Studio software onto your PC. Please note that this information is provided to help you with the installation of the software, however, the screen shots and links in this document might be out of date.

6.1 Android Studio Website

In order to create programs for your FTC robot, you will need to have Android Studio installed on your laptop. Android Studio is distributed freely by Google, and the most up-to-date reference for installing and using the Android Studio software can be found on the Android developer website:

<http://developer.android.com/sdk/index.html>

Android Studio is also available on the MacOS, Windows and Linux operating systems. This version of the manual currently only describes how to install Android Studio for the Windows environment.

6.2 System Requirements

Before you download and install the Android Studio you should first check the list of system requirements on the Android developer's website to verify that your system satisfies the list of minimum requirements:

<http://developer.android.com/sdk/index.html#Requirements>

6.3 Installing the Java Development Kit

One primary requirement is that you have the Java Development Kit (JDK) installed on your computer. You will need to download and install JDK onto your computer, prior to installing the Android Studio software.

The JDK software can be downloaded from the Oracle Java Standard Edition web page:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

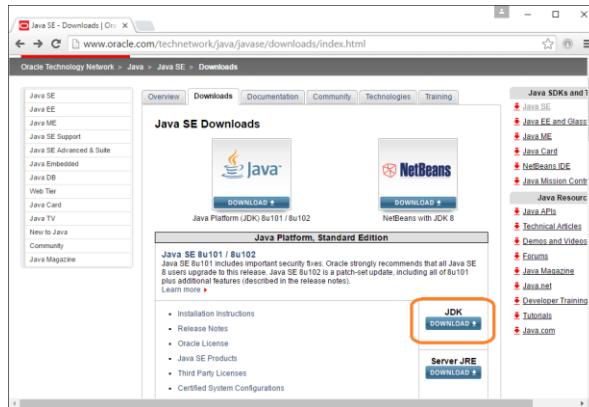


Figure 3 - Click on the DOWNLOAD button to jump to the JDK download page.

Click on the Download button located below the text “JDK” to jump to the JDK download page. Once you are on the JDK download page, find the appropriate download package for your laptop (Windows, Linux or Mac). Accept the license agreement, then click on the appropriate download link for the desired package. Note that your JDK download page might look slightly different from the one depicted in the figure below.

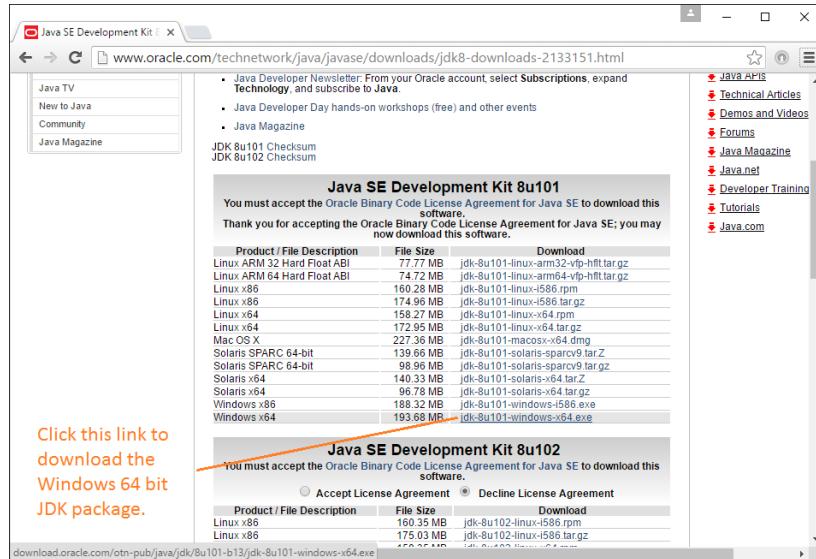


Figure 4 - After you accept the license agreement, click on the appropriate link to download the JDK software.

Once you've downloaded the appropriate JDK package, run the package and follow the on-screen instructions to install the JDK software.

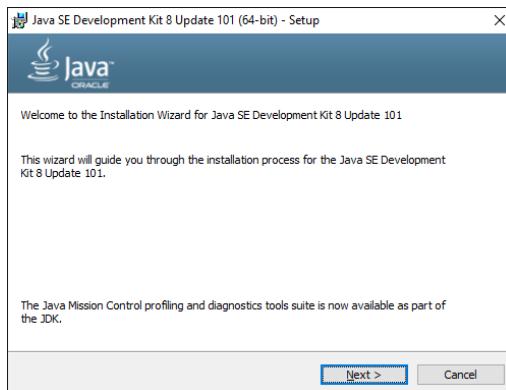


Figure 5 - Follow the setup program's instructions to install the JDK software.

6.4 Downloading and Installing Android Studio

Once you have successfully installed the JDK software, you can go to the Android developer's website to download and install Android Studio. The download URL is as follows:

<https://developer.android.com/studio/index.html>

Click on the green “DOWNLOAD ANDROID STUDIO” button to start the download process.

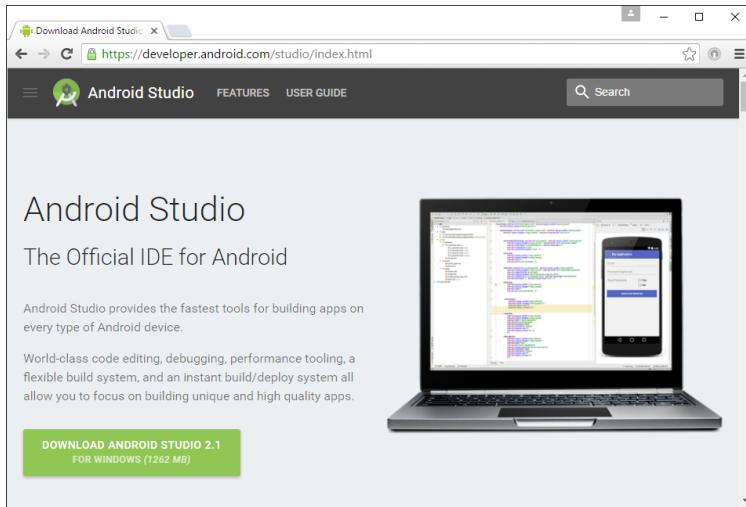


Figure 6 - Click on the green "DOWNLOAD ANDROID STUDIO" button to start the download process.

Accept the license terms and then push the blue "DOWNLOAD ANDROID STUDIO" button to download the software.



Figure 7 - Launch the setup program and follow the instructions to install Android Studio.

Once the setup package has downloaded, launch the application and follow the on-screen instructions to install Android Studio.

Note that the original version of the FTC software development kit required that the user install certain older versions of the Android SDK Platform Tools and the Android SDK Build Tools. The new FTC SDK, however, does **not** require that you download these older packages in order to work properly with the FTC software. Instead, the FTC SDK project has been updated to work with the most current Platform and Build Tools.¹

7 Downloading and Installing the FTC SDK

After successfully installing Android Studio, the last task remaining is to download and import the FTC Software Development Kit (SDK). The FTC SDK is available as an Android Studio project folder. When

¹ At the time that this section of the document was edited, was Android SDK 23 (Marshmallow)

you download this folder to your computer and then import it into the Android Studio IDE, you can use this project to customize, build and install the FTC Robot Controller app.

An *operational mode* (aka, “Op Mode”) is a code module that contains instructions that customize the behavior of a robot. Teams can use the FTC SDK to create their own Op Modes to program their robots for the autonomous and driver controlled phases of an FTC match.

The FTC SDK includes the libraries that are needed to communicate with the robot hardware (such as the Modern Robotics DC motor controller, or the Modern Robotics servo controller), plus it has a host of sample Op Modes than can be copied and used to customize a robot’s behavior. The SDK also includes the source code and resource files that define the look and behavior of the Robot Controller app.

7.1 Downloading the Android Studio Project Folder

The FTC SDK can be downloaded from a GitHub repository. GitHub is a web-based version control company that lets individuals and organizations host content online. In order to access the FTC software, you will need to have a GitHub account. You can create one for free by visiting the GitHub website:

<https://github.com/>

Once you have created a GitHub account, you can visit the FTC public repository for the FTC software:

https://github.com/ftctechnh/ftc_app

Important note for advanced GitHub users: this document assumes that the user is a novice with respect to using GitHub and the *git* version control software. If you are a GitHub power user, you can use *git* to create a local copy of the *ftc_app* repository. This document, however, does not explain how to use *git* to access the repository. It provides instructions on downloading the repository as a .ZIP file instead.

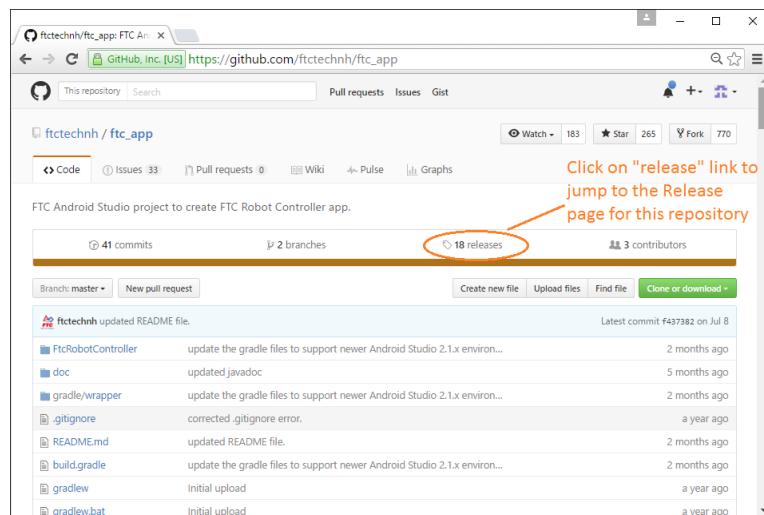


Figure 8 - Click on "releases" link to jump to the Release page for this repository.²

From the main repository web page, click on the “releases” link to jump to the **Releases** page for the repository. The **Releases** page should list the available software releases for the repository. The latest release should be displayed near the top of the page.

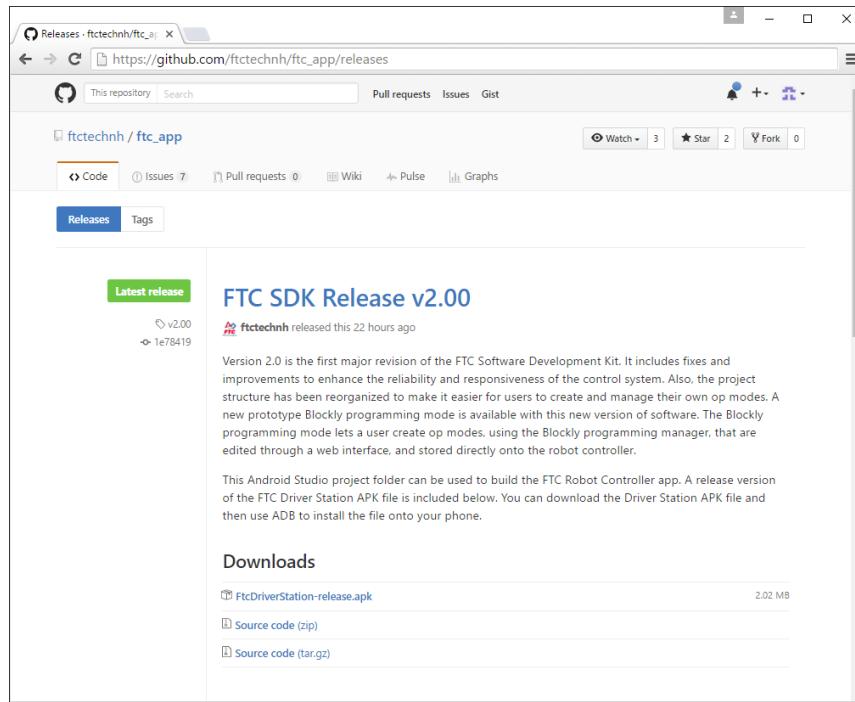


Figure 9 - The Releases page should list the available releases for the repository.

Each software release should include a **Downloads** section that you can use to download the software that you will need to program your robot.

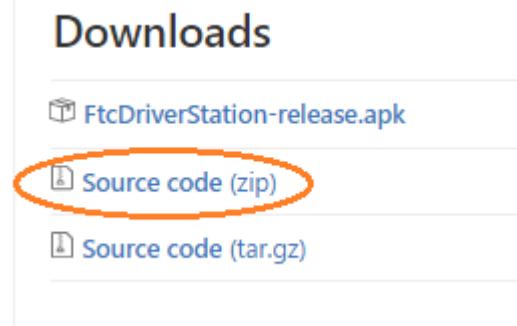


Figure 10 - Click on highlighted link to download the compressed Android Studio project folder.

Click on the *Source code (zip)* link to download the compressed Android Studio project folder.

² Note that the web page that is displayed on your computer might look slightly different than the page depicted in this figure.

7.2 Extracting the Contents of the Archived Project File

Once you have downloaded the archived (.ZIP) project file you can move this file to the location of your choice.

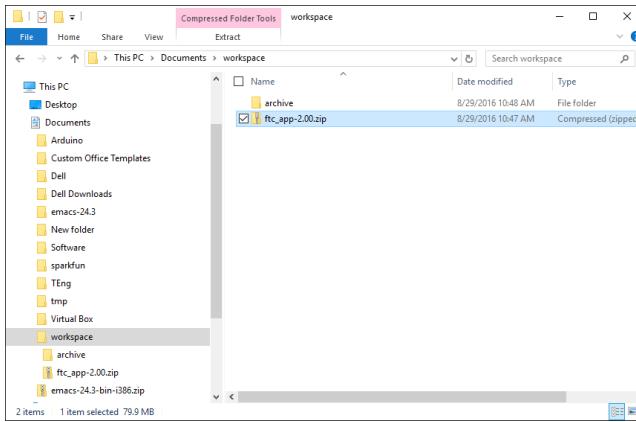


Figure 11 – You can move the downloaded file to a more convenient directory on your computer.

Before you can import the FTC project into Android Studio, you must first extract the contents of the archived project file. For Windows users, right mouse click on the file and select “Extract All” from the pop up menu. Windows should prompt you to select a destination for the extracted project folder. The dialog that appears should look similar to the one show in Figure 12.

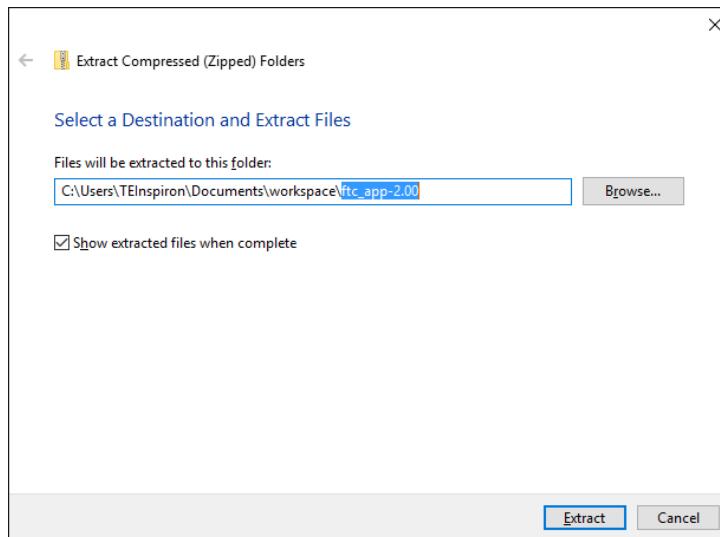


Figure 12 – You will need to provide a name for the folder that will hold the extracted files.

Highlight the suggested name for the destination folder (in Figure 12 above, the suggested name is “ftc_app-2.00”) and change the destination folder name into something more user friendly. In this example, we will change the name of the destination folder to “mycopy”.

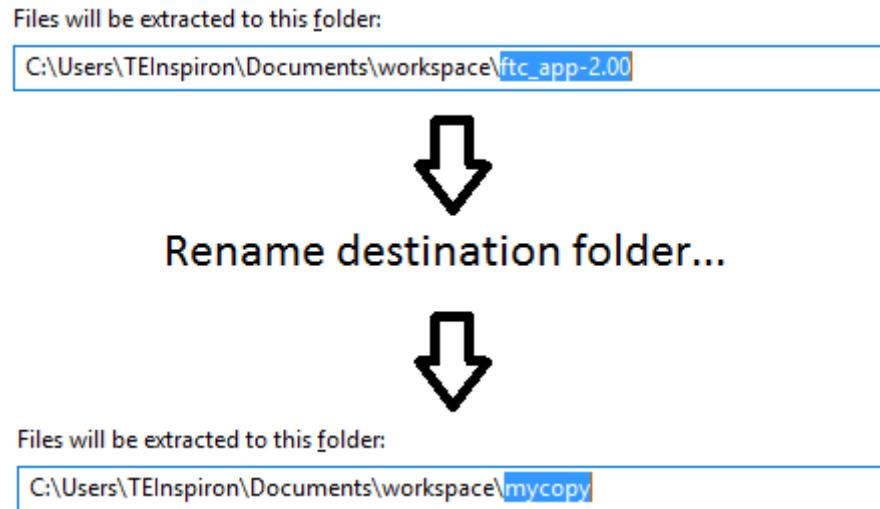


Figure 13 - Rename the destination folder to something more user friendly.

After you have renamed the destination folder, extract the contents of the archive to the folder. After the extraction process is complete, verify that the project folder was successfully extracted to its target destination.

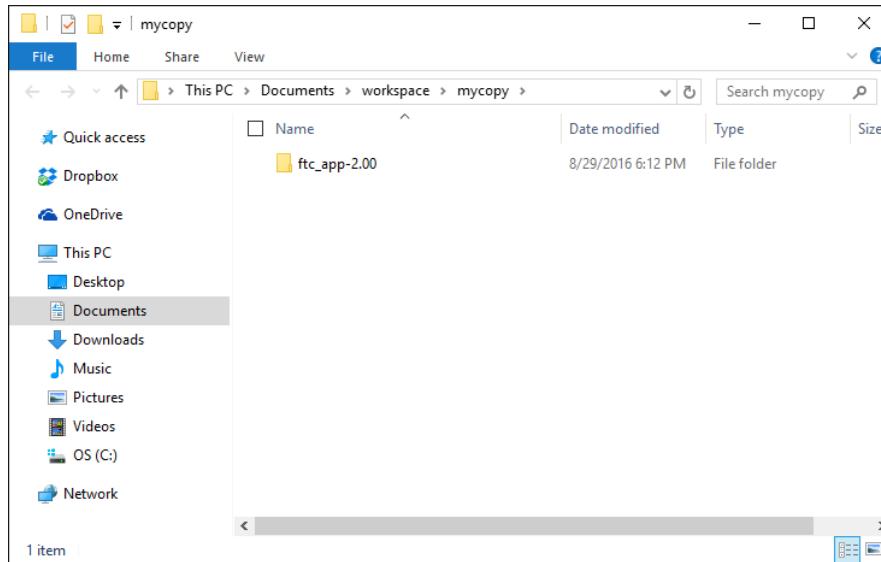


Figure 14 - Verify that the project folder was successfully extracted to the destination folder.

Once you have successfully extracted the contents of the archived file, you are ready to import the FTC project into Android Studio.

7.3 Importing the FTC Project into Android Studio

In order to import the FTC Project, you will need to launch the Android Studio software on your computer. On the main Android Studio Welcome screen, select the option to “Import project (Eclipse, ADT, Gradle, etc.)” to begin the import process.

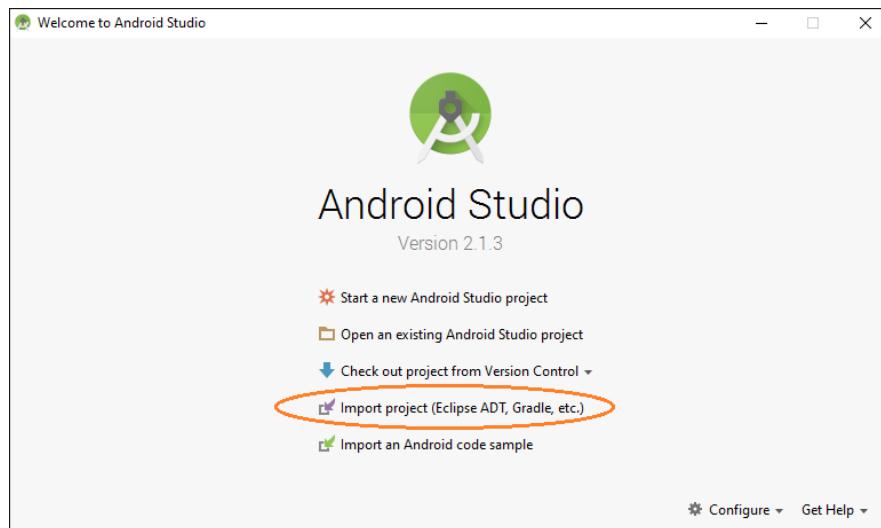


Figure 15 - Select the "Import project" option to start the import process.

Android Studio should prompt you to select the project folder that you would like to import. Use the file browser in the pop up dialog box to locate and then select the folder that you extracted in section 7.2 of this document. Make sure you select the extracted project folder (and not the .ZIP file which might have a similar name to the extracted folder). Hit the “OK” button to import the selected project into Android Studio.

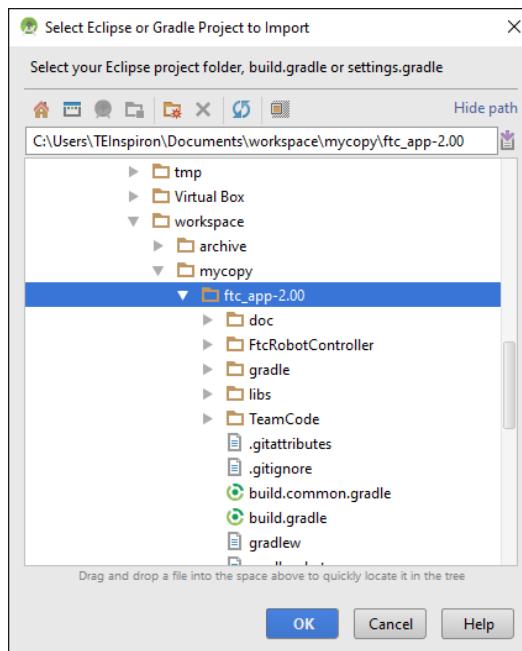


Figure 16 - Select the project folder and hit “OK” to import the project into Android Studio.

In Figure 16 (see above) the project folder called “ftc_app-2.00” is selected to be imported into Android Studio. It might take Android Studio several minutes to import the project. Once the project has been

successfully imported, the screen should look similar to the one depicted in **Error! Reference source not found. Error! Reference source not found..**

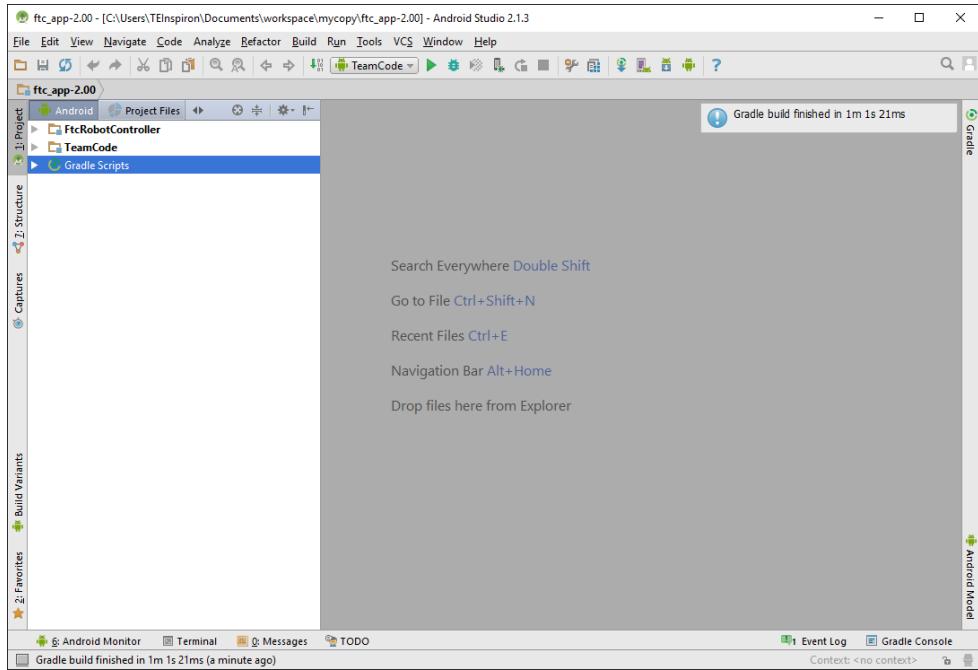


Figure 17 - If the project was successfully imported, then your screen should look similar to this screen.

8 Using Your Robot

Now that you've successfully configured your development environment on your computer, let's use the FTC software to create an op mode that will allow us to control learn the basics of how to get your working with the software.

Before you begin, you will need to have the following components:

1. A pair of approved Android devices (such as the ZTE Speed or the Moto G 2nd or 3rd Gen phones).
2. A single Logitech F310 gamepad and a micro USB OTG adapter (to connect the gamepad to one of the phones)
3. A development laptop with Android Studio and the FTC SDK software installed.
4. A USB micro cable to connect the Android devices to the laptop.

Also, if you are using the ZTE Speed or Motorola Moto G phones, you will need to ensure that your phones are able to connect through USB to your computer. For Window users, the first time you connect either of the phones to your PC, you will need to install the appropriate software onto your PC. ZTE Speed users can consult Appendix B: Preparing the ZTE Speed for use with FTC for details on this process. Motorola Moto G users should consult the Motorola website for details on installing the appropriate software. At the time that this section was edited, the following link was available on the Motorola support website for the Moto G 2nd Gen phone:

https://motorola-global-portal.custhelp.com/app/product_page/faqs/p/30,6720,9151

8.1 Enabling Developer Options

After you have configured your Android phone, you will also have to make sure that your phone is in developer mode before you will be able to install apps onto the phone using the tools that are included with Android Studio. The Android Developer website contains information on how to enable Developer Options onto your phone. If you visit the following link and read the section entitled “Enabling On-device Developer Options” you will see that you can enable Developer Options on your Android phone by going to **Settings>About phone** on the phone, and then tapping the **Build** number seven times.

<https://developer.android.com/studio/run/device.html#device-developer-options>

In order to be able to use the Android Studio tools to install apps onto your phone, you will need to make sure that the **Developer Options** and **USB debugging** are enabled for both of your phones.

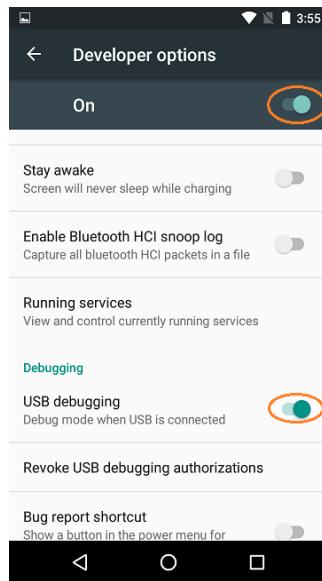


Figure 18 - Make sure that Developer options and USB debugging are enabled on both phones.

When you first connect a phone to your computer with Android Studio running, the phone might prompt you if it is OK to allow the computer to have USB debugging access to the phone. If this happens, make sure that you check the “Always allow from this computer” option and hit the **OK** button to allow USB debugging.

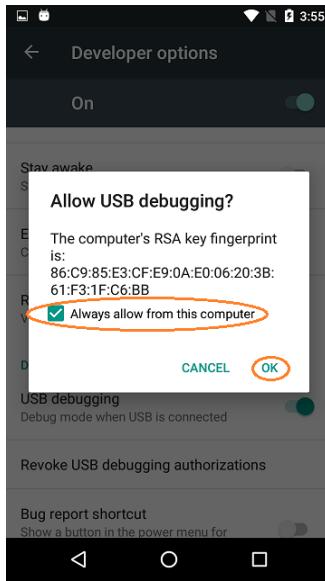


Figure 19 – Check the “Always allow from this computer” option and hit OK to allow USB debugging.

8.2 Copying an Example Op Mode

We can use Android Studio to copy an example Op Mode that can be used to drive a Pushbot robot. Let's take a look at the Android Studio user interface. In the left hand side, there should be a **Project** pane that shows the Project Browser. If you do not see the Project Browser, click on the word **Project** on the left hand border to expand the pane and display the Browser.

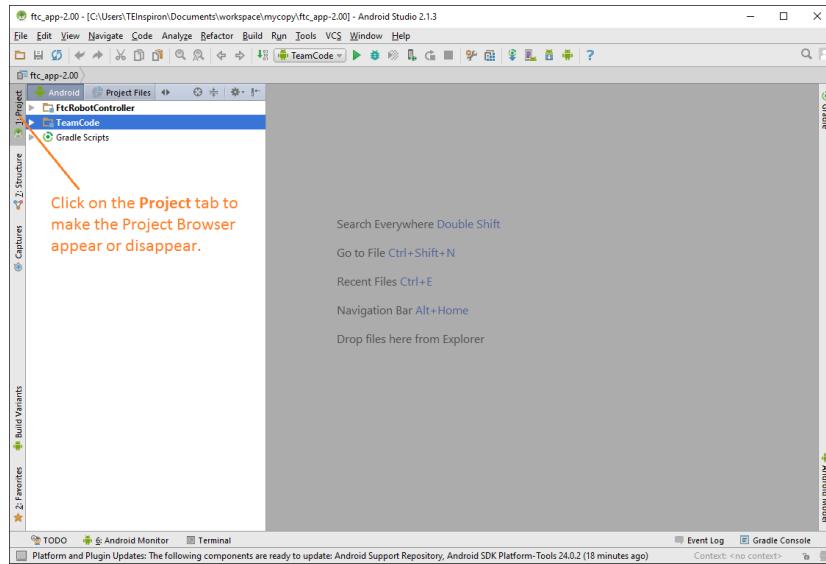


Figure 20 -Click on the Project tab to make the Project Browser appear or disappear.

For the FTC project, there should be two *packages* in the Project Browser. The **FtcRobotController** package contains example Op Modes that can be used to control a PushBot. We are going to copy some sample files from the **FtcRobotController** package and then paste them into the **TeamCode** package. The **TeamCode** package is where your team's custom op modes and other source files should reside.

Use the Project Browser to expand the **FtcRobotController** package and find the **FtcRobotController->java->org.firstinspires.ftc.robotcontroller->external.samples** folder. Look for the file called **HardwarePushbot** and right mouse click on the file and copy it.

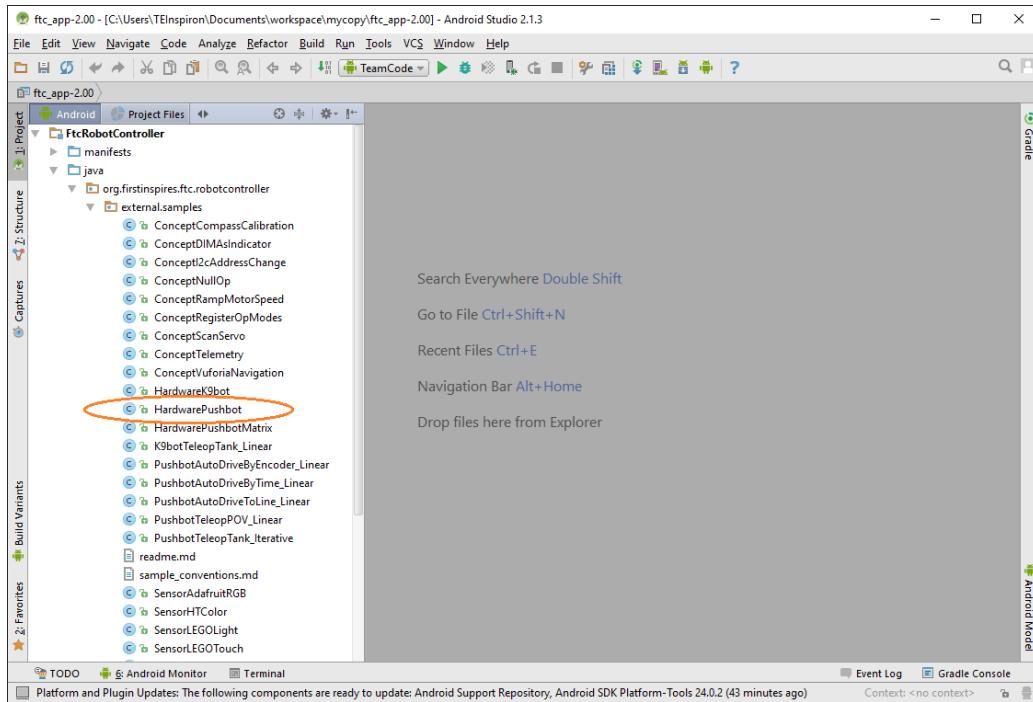


Figure 21 - Right mouse click on and copy the file called HardwarePushbot.

Use the project browser to find the **TeamCode->java->org.firstinspires.ftc.teamcode** folder.

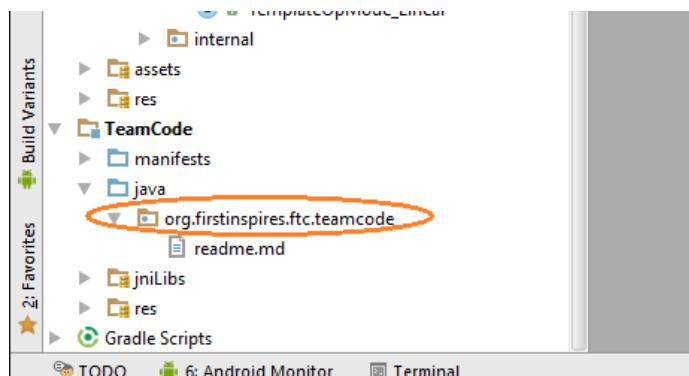


Figure 22 - Use the Project Browser to find the org.firstinspires.ftc.teamcode folder.

Right mouse click and paste the copied HardwarePushbot file into the **org.firstinspires.ftc.teamcode** folder. Android Studio should prompt you for a name for the copied file. Hit **OK** to accept the default name and the destination package.

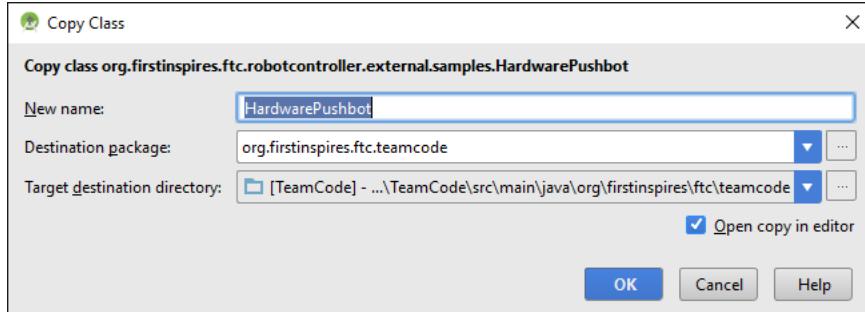


Figure 23 – Android Studio will prompt you for a new name. Hit OK to accept the default name and destination package.

The copied **HardwarePushbot** file should now be visible in the **TeamCode->java->org.firstinspires.ftc.teamcode** folder. The source code for the **HardwarePushbot** file should be visible in the editing pane on the right hand side of the Android Studio user interface.

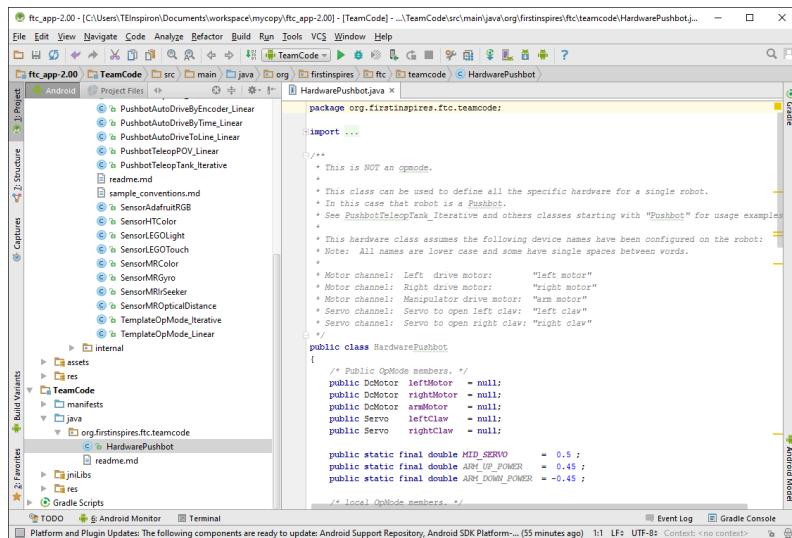


Figure 24 – The copied file should now appear in the destination folder.

HardwarePushbot contains source code that will configure the hardware for a standard FTC Pushbot. It is not an actual Op Mode file. Instead, it is a code module that is used by other op modes.

We need to copy one additional file so that we will have an actual op mode that we can run on your robot. Look in the **FtcRobotController** package for an example file called **PushbotTeleopTank_Iterative**. Right mouse click on the file and copy it over to the **TeamCode->java->org.firstinspires.ftc.teamcode** folder.

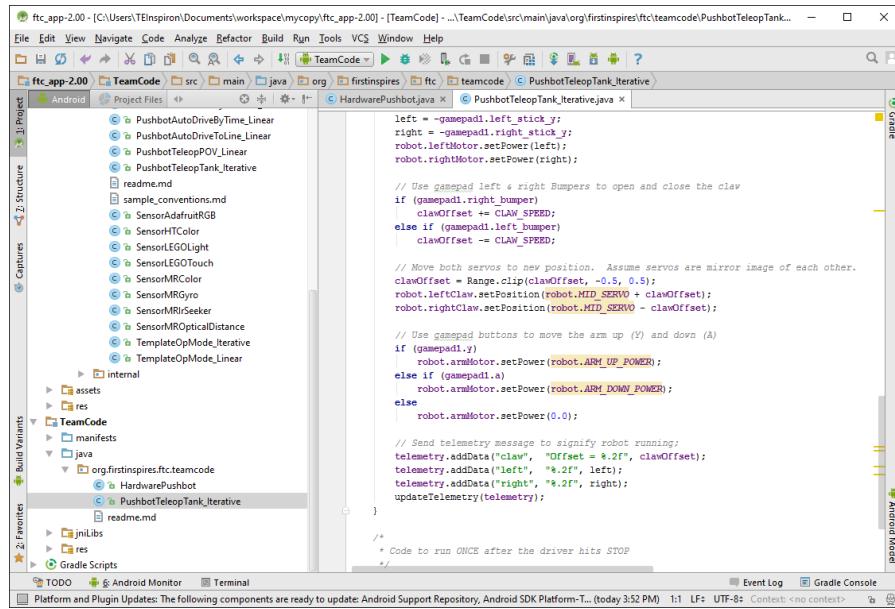


Figure 25 - Copy the file PushbotTeleopTank_Iterative to the same destination folder.

8.3 Registering Your Op Mode

Before your robot can execute your Op Mode, you need to first register it with the Robot Controller. The FTC Android Studio project lets you create multiple Op Modes for your robot. Before a user can select one of these op modes to be executed on the robot, you must first make sure that op mode is registered as either an autonomous or tele-op (driver controlled) Op Mode.

Let's take a look at the source code for the PushbotTeleopTank_Iterative Op Mode. Within the source file there should be the following phrase:

```
public class PushbotTeleopTank_Iterative extends OpMode {
```

This line is the beginning of the class definition for the Op Mode. Above this line, there are two lines which look like the following,

```
@TeleOp(name="Pushbot: Teleop Tank", group="Pushbot")
@Disabled
```

The first line is how the Op Mode gets registered with the Robot Controller. In this case the annotation @TeleOp tells the Robot Controller that this op mode should be registered as a tele-operated (and not autonomous) op mode. The name “Pushbot: Teleop Tank” is the name that will be listed on the Driver Station when the user wants to select this op mode for execution. The group “Pushbot” is the name of the group that this Op Mode belongs too.

The annotation @Disabled tells the Robot Controller that even though this Op Mode has been registered, we want to keep it disabled (not visible to the user) for now. If we comment out or delete this annotation, then the Op Mode will be registered and visible to the user.

Comment out the @Disabled line on your op mode so that the code looks like the following,

```

@TeleOp(name="Pushbot: Teleop Tank",our  group="Pushbot")
//@Disabled
public class PushbotTeleopTank_Iterative extends OpMode{

```

Now that the `@Disabled` annotation has been commented out, we are ready to build and install your Robot Controller app onto your Android phone.

8.4 Building the Robot Controller App

Before we build the Robot Controller app, we should first make sure the WiFi Direct name of your Android phone comply with the rules specified in the Game Manual part 1. Rule <RS01> specifies that “each Team MUST “name” their Robot Controller with the Official FIRST Tech Challenge Team number and an -RC” indicating that it is a Robot Controller. If a team has more than one set of phones, the phone names should have a hyphenated letter in the middle.

You will need to consult the documentation for your phone to determine precisely how to rename your device.

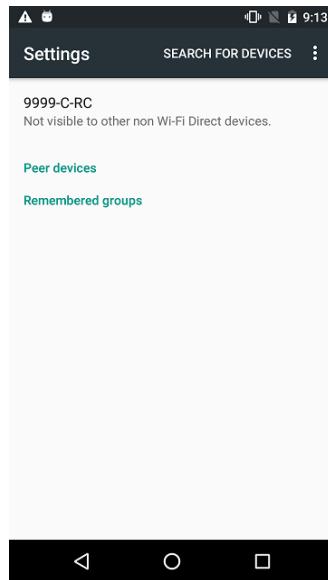


Figure 26 – The Robot Controller’s name is “9999-C-RC” for this example.

In this example, the pair of phones are named “9999-C-RC” (for the Robot Controller) and “9999-C-DS” (for the Driver Station).

Verify that the Robot Controller phone is connected to your laptop and that the laptop has USB debugging permission for the phone. Look towards the top of the Android Studio user interface and find the little green Play button (which is represented by a green triangle) next to the words “Team Code”. Press this green button to build the Robot Controller app and to install it onto your phone (see Figure 27 below).

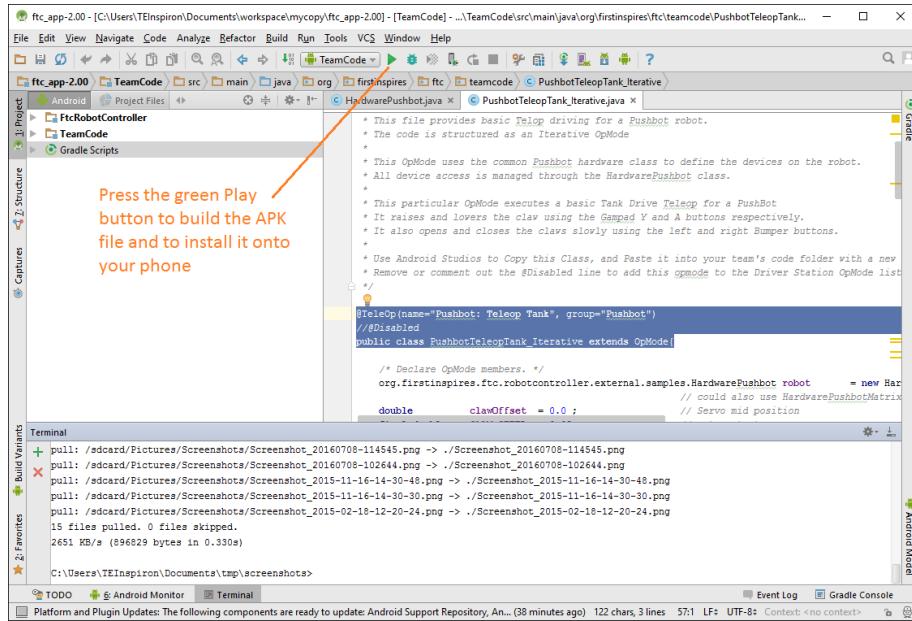


Figure 27 - Press the Play button to build the app and install it on your phone.

Android Studio should prompt you to select a target device to install the Robot Controller app. Your screen might look something like the image shown below.

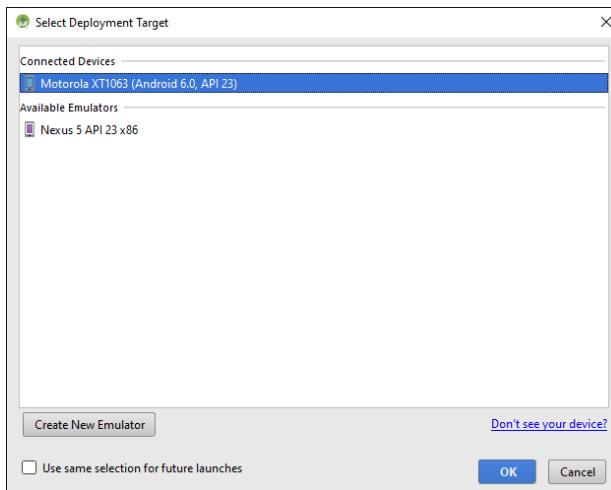


Figure 28 - Android Studio might prompt you for the target device.

Make sure that you select the correct target device. In Figure 28 above the Motorola phone is selected as the target device. The virtual Android device emulator is NOT selected. Hit **OK** to build the APK file and install it on the target device.

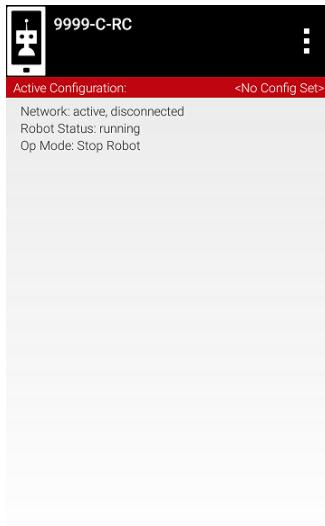


Figure 29 - The Robot Controller app should launch if the file was successfully installed.

If the installation was successful, the Robot Controller app should be launched on the target Android device.

8.5 Configuring the Robot Controller

Before we can execute any op modes you need to first tell the Robot Controller what electronic devices are connected to the Android phone. For this example, we are using an FTC Pushbot robot that was built using the Pushbot Build Guide that is listed on the FTC Robot Building Resources page:

<http://www.firstinspires.org/resource-library/ftc/robot-building-resources>

The Pushbot robot is configured with two DC motor controllers, one servo controller, and one device interface (I/O) module. You need to create a configuration file that contains information describing all of these devices to the Robot Controller.

The Robot Controller app has an auto configuration feature that makes creating a configuration file a little easier. If you successfully installed the Robot Controller app, and you have a working Pushbot on hand, you can power on the Pushbot and then connect the phone's USB port to the Micro USB OTG adapter cable on the Pushbot (which should be connected to the USB port of the Power Distribution module).

The first time you connect the phone to the Pushbot's USB devices, the phone might prompt you if it's OK to associate the Robot Controller app with the newly detected USB device. If this happens, you want to make sure that you check the "Use by default for this USB device" option and then hit **OK** to associate the Robot Controller app with the USB device. You probably will have to associate the app with each USB device that is detected by the Android phone. Often you will be prompted by the phone when you first connect the phone to the robot's Power Distribution module. Also, when you run a hardware scan through the Robot Controller app, the phone might prompt you again. Once you have associated the

app with all of the devices on the robot, you should not have to associate them a second time, unless you erase these default associations (for example, by uninstalling the Robot Controller app, then reinstalling it).

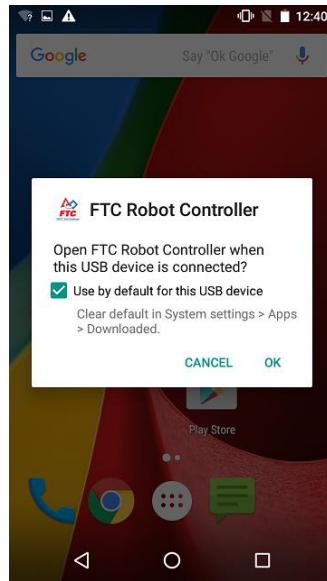


Figure 30 – Check “Use by default” and hit OK to associate the Robot Controller app with newly detected hardware.

Once the phone is connected to the Pushbot, you touch the three vertical dots in the upper right hand corner of the main Robot Controller screen (see Figure 29 on page 23) to display the main menu for the Robot Controller app. Select the “Configure Robot” option from the pop up menu.

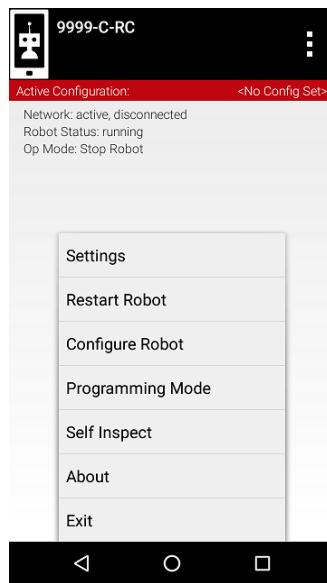


Figure 31 - Select "Configure Robot" from the pop up menu.

On the Configure Robot screen you have the option of creating a new configuration file from scratch, or you can use a pre-made template to configure your robot quickly. In this example, since we are using

the FTC Pushbot, we will push the “Configure from Template” button to use a pre-made template to create our file.

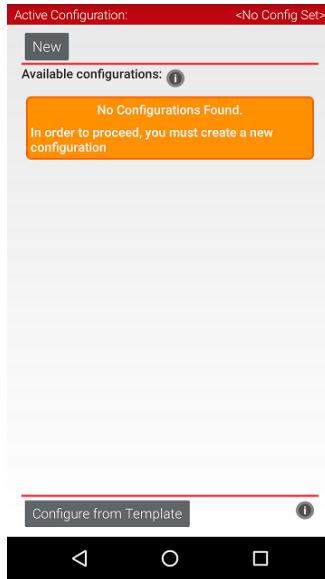


Figure 32 - Press the Configure from Template button.

The Robot Controller app should now display a list of pre-made templates that you can choose to create your configuration file. For this example, we will select the “PushBot” template. This is the template for a standard Pushbot without any sensors configured. If you press the “Info” button for a template, the screen should display information about the configuration template.

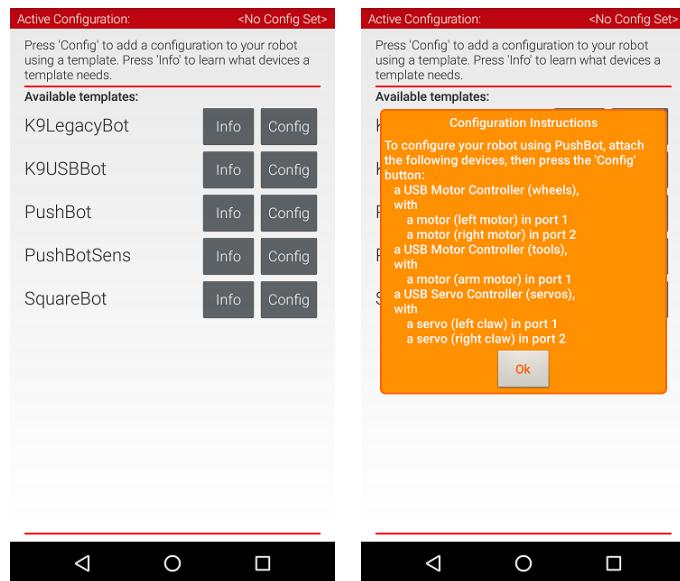


Figure 33 – Pressing the “Info” button displays the configuration info for the template.

Once you have verified that your robot’s configuration matches the information listed in the template, hit the “Config” button to apply the template and create the configuration file.

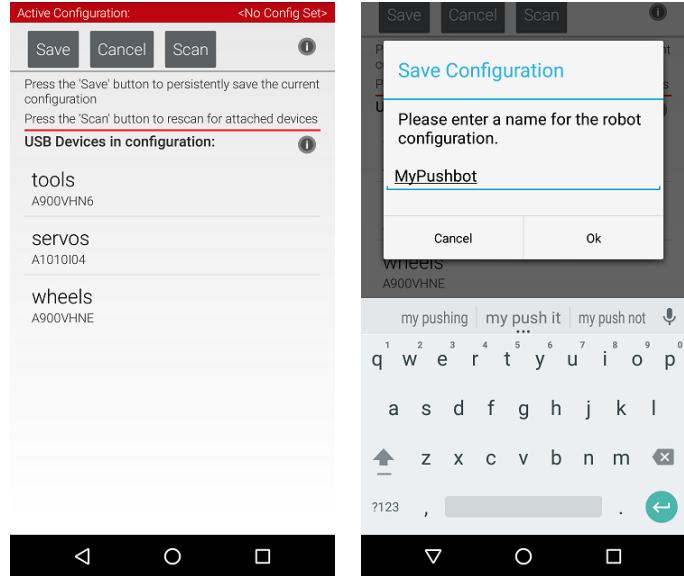


Figure 34 - You can click on a device to view its configuration. Hit Save to save the configuration file.

The Robot Controller app should display the active configuration file. You can click on device's name to inspect the configuration settings for a device. Once you have verified the configuration information, click on the Save button and then specify a file name and hit **OK** to save the information to the file. After you have saved the configuration information, use the Android back arrow (at the bottom of the screen) to exit back to the main screen.

8.6 Installing the FTC Driver Station App

The Driver Station is the Android device that sits near the team drivers and is connected to one or two USB gamepads. Teams use the touch screen and the gamepads to operate their robot remotely. The Driver Station Android device runs a special app called the *FTC Driver Station*.

The FTC Driver Station app is included with the FTC software that you downloaded from GitHub. We are going to use a tool called the *Android Debug Bridge* (adb) to install the Driver Station app onto the phone. The Driver Station app is located in the *doc/apk* subdirectory of the FTC project folder that you downloaded from GitHub. The name of the app is *FtcDriverStation-release.apk*.

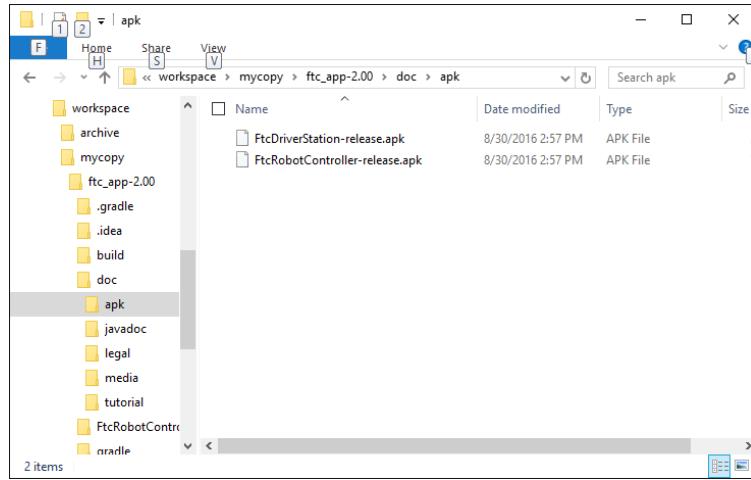


Figure 35 - The Driver Station app is located in the doc/apk subdirectory.

The Android Studio IDE has a built in Terminal interface that you can use to issue commands to install the Driver Station app onto your phone. First, you should connect your phone with a USB cable to your development computer (make sure the computer has USB debugging permission for that phone).

Next you want to click on the **Terminal** tab at the bottom of the Android Studio interface. This will open a Terminal pane that you can use to enter commands for your computer.

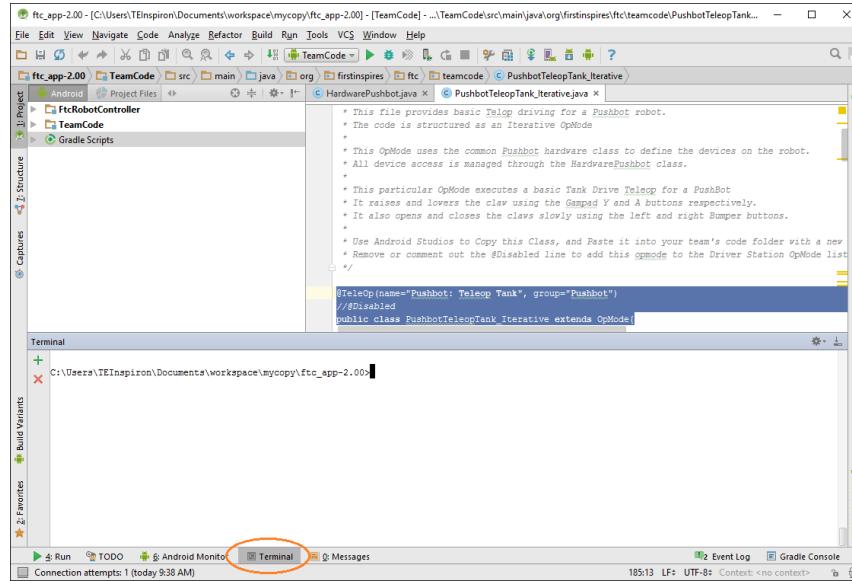


Figure 36 - Click on the Terminal button to bring up the terminal to your computer.

For Windows computers, you can use DOS commands to navigate in the terminal window to the directory that holds the Driver Station APK file.

```
C:\Users\TEInspiron\Documents\workspace\mycopy\ftc_app-2.00>cd doc/apk
C:\Users\TEInspiron\Documents\workspace\mycopy\ftc_app-2.00\doc\apk>dir
Volume in drive C is OS
Volume Serial Number is D29F-1C2D

Directory of C:\Users\TEInspiron\Documents\workspace\mycopy\ftc_app-2.00\doc\apk

08/30/2016  02:58 PM    <DIR>      .
08/30/2016  02:58 PM    <DIR>      ..
08/30/2016  02:57 PM           2,116,152 FtcDriverStation-release.apk
08/30/2016  02:57 PM           7,937,712 FtcRobotController-release.apk
              2 File(s)     10,053,864 bytes
              2 Dir(s)   425,976,217,600 bytes free

C:\Users\TEInspiron\Documents\workspace\mycopy\ftc_app-2.00\doc\apk>
```

Figure 37 - Use terminal commands to change to the directory that holds the Driver Station file.

In order to use the adb command to install the app onto the phone from the terminal, we need to first tell the computer where the adb command file is located. By default, when you install Android Studio, although the installation process installs the adb command file, it does not configure your computer to automatically know where the file resides when called from the computer terminal.

For Windows 10 users, the file typically resides in a path similar to the following:

```
C:\Users\TEInspiron\AppData\Local\Android\sdk\platform-tools
```

In this example, “TEInspiron” is the user name for the demo account used to help write this document. Your path to the platform-tools folder might differ slightly from the path used in this example.

Once you have verified this path, in the terminal window type the following command (except replace “C:\Users\TEInspiron\AppData\Local\Android\sdk\platform-tools” with the path that matches the folder location for your computer):

```
setx PATH "%PATH%;C:\Users\TEInspiron\AppData\Local\Android\sdk\platform-tools"
```

Once you have entered in this command, the location of the adb.exe file should have been appended to the PATH variable for your computer terminal. If you are still in the directory where the Driver Station file is located, you can type in the following command to use ADB to install the app onto your phone

```
adb install FtcDriverStation-release.apk
```

If the installation was successfully, then the words “Success” should be displayed in the terminal window.

```
Terminal
+
 Directory of C:\Users\TEInspiron\Documents\workspace\mycopy\ftc_app-2.00\doc\apk

08/30/2016 02:58 PM <DIR> .
08/30/2016 02:58 PM <DIR> ..
08/30/2016 02:57 PM 2,116,152 FtcDriverStation-release.apk
08/30/2016 02:57 PM 7,937,712 FtcRobotController-release.apk
2 File(s) 10,053,864 bytes
2 Dir(s) 425,668,866,048 bytes free

C:\Users\TEInspiron\Documents\workspace\mycopy\ftc_app-2.00\doc\apk>adb install FtcDriverStation-release.apk
1968 KB/s (2116152 bytes in 1.049s)
pkg: /data/local/tmp/FtcDriverStation-release.apk
Success

C:\Users\TEInspiron\Documents\workspace\mycopy\ftc_app-2.00\doc\apk>
```

Figure 38 - If the installation was successful, the word "Success" should be displayed in the terminal.

If successful, you should be able to find the icon for the FTC Driver Station on the phone. Touch the icon to launch the Driver Station app.

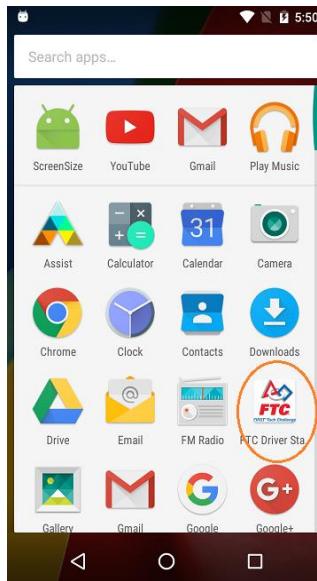


Figure 39 - Locate the Driver Station and tap it to launch the app.

The Driver Station main screen should appear.

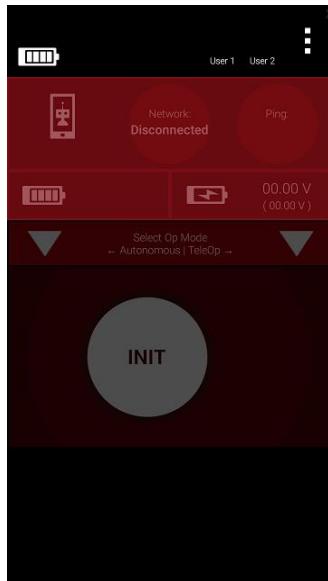


Figure 40 - The Driver Station main screen should appear.

8.7 Pairing the Driver Station to the Robot Controller

Now that you have the Driver Station app installed, you need to pair (i.e., create a wireless connection) between the Driver Station to the Robot Controller. From the main Driver Station screen, touch the three dots that appear in the upper right hand corner to open the main menu. Select **Settings** and then “Connect with Robot Controller” to open the Pairing activity on the Driver Station app.

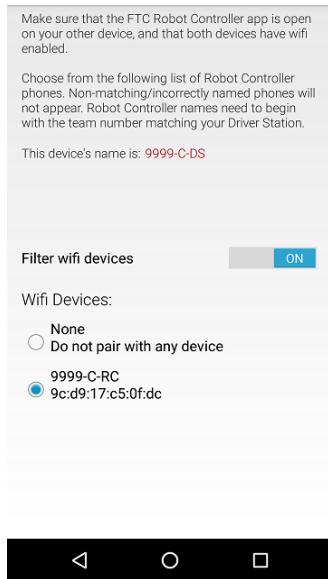


Figure 41 - Using the Pairing activity on the Driver Station to find and connect to the Robot Controller.

The Pairing screen or activity is the interface that you use to find and connect to a Robot Controller. If your Robot Controller was up and running when you started up the Driver Station app, then when you enter the Pairing screen you should see the name of the Robot Controller listed as one of the available

devices to pair to (“9999-C-RC” in the example show in Figure 41 above). Select the desired Robot Controller from the list and use the back arrow to exit the screen and return to the main screen.

On the Robot Controller, the first time you attempt to create a WiFi connection, the Android device will prompt you to accept or reject the connection request. Press **ACCEPT** to accept the connection request and to connect to the Driver Station.

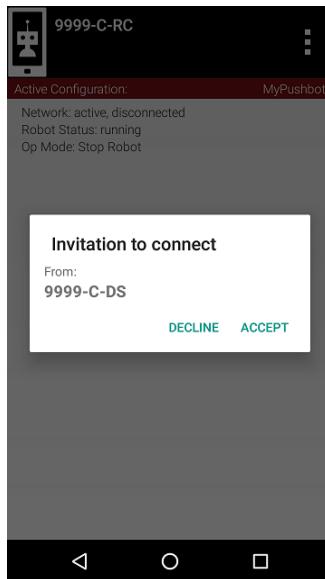


Figure 42 - The Robot Controller will prompt you the first time you try to connect to accept or reject the request.

In a few moments, the Driver Station should establish a connection to the Robot Controller and Status information should be displayed on the Driver Station screen.

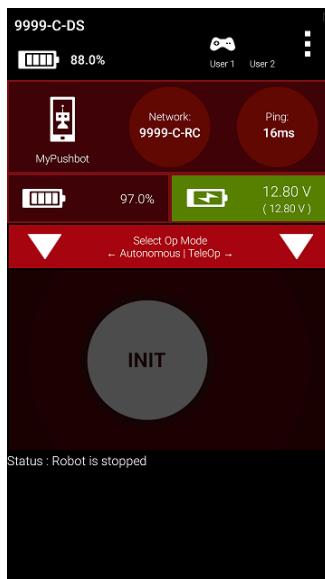


Figure 43 - Status information will be displayed upon establishing a connection to the Robot Controller.

8.8 Using a Gamepad

The sample Op Mode that we incorporated into our Robot Controller requires a gamepad to control the robot. Connect a Logitech F310 gamepad controller to the USB Micro port on your Driver Station using the USB Micro OTG adapter. Make sure the switch on the bottom of the Logitech F310 controller is set to the “X” position.



Figure 44 - Make sure the switch is in the "X" position.

When you connect your gamepads to the Android device (through a USB hub), you need to tell the FTC Driver Station app which gamepad will be used to represent driver #1 (User 1) and which gamepad will be used to represent driver #2 (User 2).

You can select which driver a gamepad will represent by pushing the **Start** key on the gamepad while simultaneously pressing the **A** button if you want to be driver #1, or the **B button** if you want to be driver #2.



Figure 45 - Press Start + A to be driver 1 or Start + B to be driver 2.

When you first connect your gamepads to the Android device, you must push a button combination to designate which driver your gamepad will represent.

Once you have designate which driver your gamepad will represent the gamepad status information will appear below the appropriate area near the top of the screen.

For example, supposed the gamepad that represents driver #1 is active, then the gamepad icon for gamepad 1, near the upper right hand corner of the screen, will be highlighted green.

If the gamepad that represents driver #2 is active, then the gamepad icon for gamepad 2, near the upper right hand corner of the screen, will be highlighted green.

If both gamepads are being used simultaneously, then both gamepad icons will be highlighted.

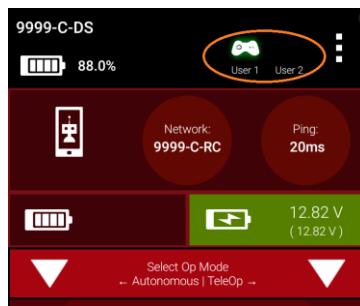


Figure 46 – Gamepad activity is displayed in the upper right hand corner of the screen.

For our example Op Mode, we will want to have a gamepad designated as User 1.

8.9 Selecting and Running an Op Mode

In our example, we registered our Op Mode as a “Teleop” Op Mode. To find and select this Op Mode, touch the TeleOp drop down control (the white downwards pointing triangle) on the right hand side of the screen to display the list of available TeleOp Op Modes. For our example, since we only registered a single Op Mode, we only see the “Pushbot: Teleop Tank” Op Mode displayed in the list. Touch the Op Mode to select it from the list.

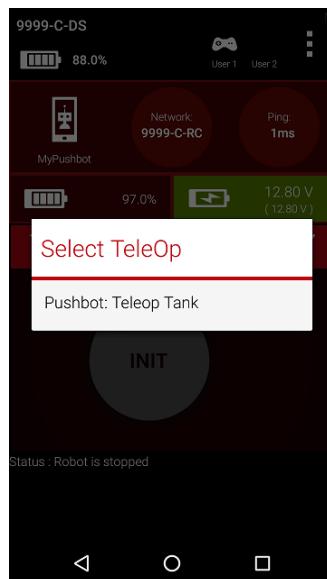


Figure 47 – Touch the TeleOp drop down control to display a list of available TeleOp Op Modes.

Once you have selected the Op Mode, the **INIT** button on the Driver Station should become enabled. Press the **INIT** button to initialize the Op Mode.

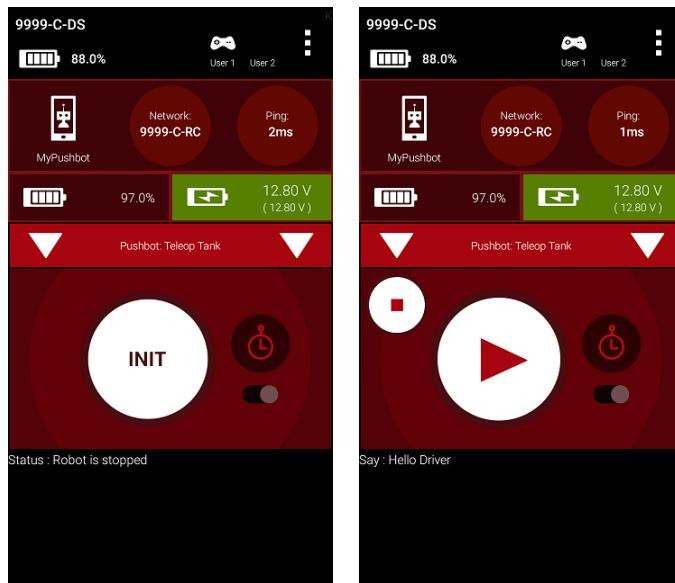


Figure 48 – Press the INIT button to initialize the Op Mode.

To start the Op Mode, press the triangular shaped START button on the Driver Station interface. Use the gamepad controller to drive the robot around. Press the square shaped STOP button to stop the Op Mode run.

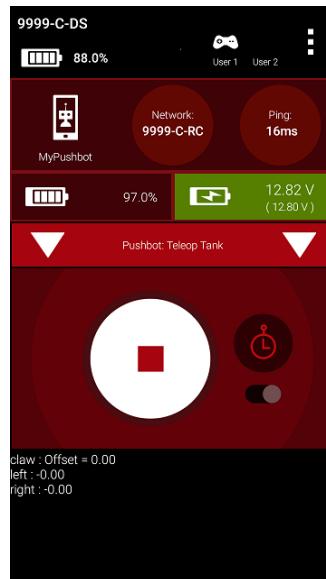


Figure 49 – If you would like to stop the Op Mode run, press the square shaped STOP button.

Ideally, when you run “Pushbot: Teleop Tank”, the two joysticks on the gamepad should control the left and right drive motors on the robot. The buttons on the right hand side of the gamepad should control

the DC motor that raises and lowers the arm. If this is not the case for your robot, and the controls seem reversed, do not panic! The next section will show you how to correct this problem.

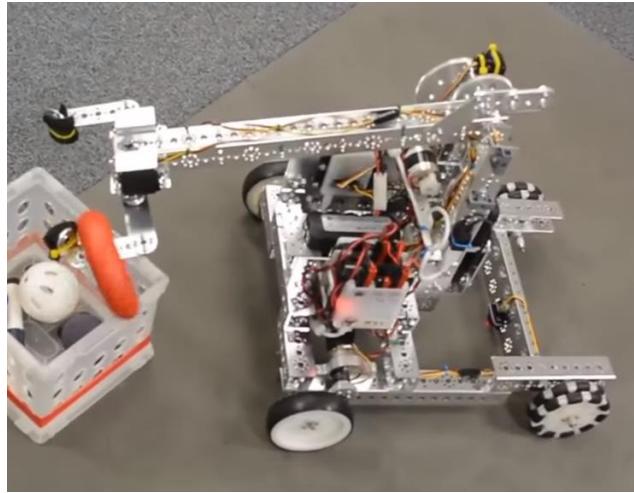


Figure 50 - You should now be able to drive your Pushbot using the gamepad.

8.10 Swapping USB Devices in the Configuration File

A typical Pushbot robot is configured with the following electronic modules:

- One DC motor controller that is connected to the drive wheels of the robot.
- One DC motor controller that is connected to the arm motor of the robot.
- One servo controller that is connected to the manipulator servos.
- One device interface module that is connected to the sensors. Note that in our example, we ignored the device interface module, and only configured the motor and servo controllers.

Each USB device has its own unique serial that can be used to identify one device from another. Currently, with the FTC Control System, the Android phone does not distinguish by USB port between one USB device and other. If your robot is equipped with two USB devices of the same type (for example, if your robot has two DC motor controllers) the configurator mechanism does not currently have the ability to distinguish one from the other by USB port. It can only distinguish between the two by serial number.

When we used the auto configuration mechanism to create a configuration file for our robot (see Section 8.5, page 23) of this document the configurator arbitrarily chose one of the USB DC motor controllers that it detected as the “wheels” motor controller. It chose the other USB DC motor controller as the “tools” motor controller. If you edit your active configuration file, you should be able to see the configured USB devices with their device names (“tools”, “servos” and “wheels”) and with their serial numbers listed below the device names (“A900VHN6”, “A1010I04” and “A900VHNE”).

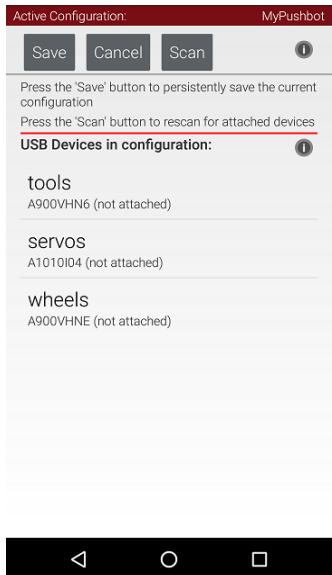


Figure 51 - The Device Names and USB Serial Numbers should be listed in a configuration file.

If during the configuration step, the devices that control the “wheels” and the “tools” were incorrectly configured, the Robot Controller user interface has a **Swap** mechanism that lets a user swap the devices quickly.

In Figure 51 (see above) if the DC motor controller with serial number “A900VHNE” should be designated as the “tools” controller and the motor controller with serial number “A900VHN6” should be designated as the “wheels” controller, then the **Swap** mechanism to switch the two in the configuration file.

To switch the devices, touch one of the devices (for example touch “tools” in the screen shown in Figure 51) to display the detailed configuration information for that USB device. Press the **Swap** button at the top of the USB device screen to start the swap process.

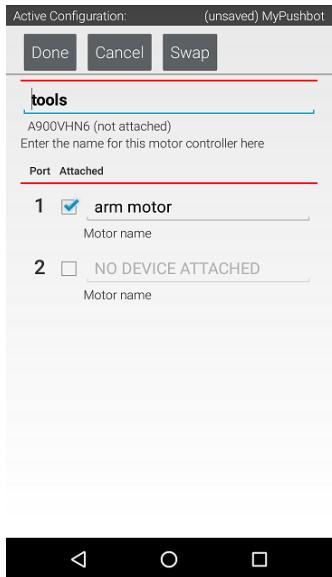


Figure 52 - Press the Swap button to start the swap process.

After pushing the **Swap** button the Driver Station will list the available USB devices of the same type. Tap on the device that you want to swap.

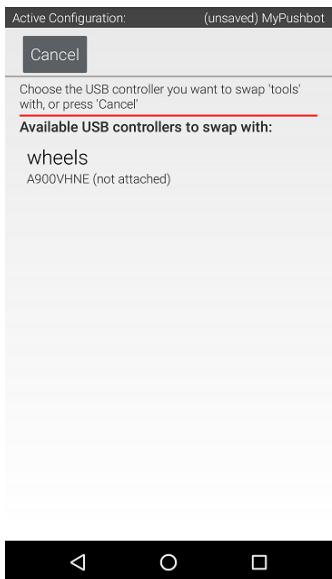


Figure 53 – Touch or tap on the device that you want to swap.

The devices should now be swapped. In our example, the motor controller with serial number “A900VHNE” should now be assigned as the “tools” controller. The motor controller with the serial number “A900VHN6” should be assigned as the “wheels” controller.

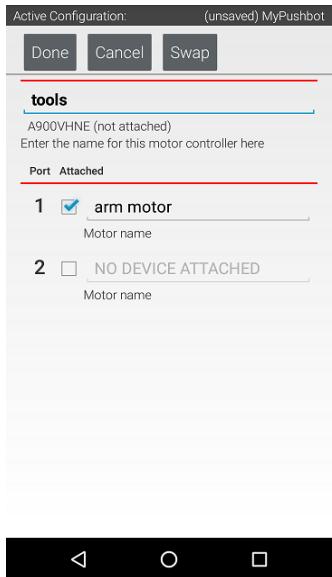


Figure 54 – The Robot Controller should automatically swap the devices.

After the devices have been swapped, don't forget to save the changes to your configuration file!

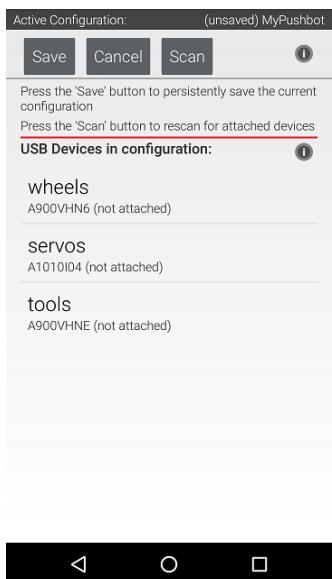


Figure 55 – Save your changes after the swap is complete.

9 For More Information...

For additional information on the FTC control system technology, please visit the Robot Building Resources page of the FIRSTInspires.org website:

<http://www.firstinspires.org/resource-library/ftc/robot-building-resources>

Appendix A: Note about USB Ports and Cables

The next gen platform uses USB (which stands for Universal Serial Bus) to connect the Android devices to the robot and game controller hardware. The USB cables allow the devices to communicate at a very high speed. The USB cables also provide power to the devices that are “downstream” to the Power Module.

USB Host & Client Modes

A USB-enabled device such as a smartphone or tablet can act as a USB host or as a USB client device. A USB host is the device that initiates all communications on the bus. The USB client responds to communications from the host. The Android devices that are used for the next gen platform have the ability to act as either a USB host or a USB client. When you plug the Android device into a laptop the Android device acts as a USB client. When you plug a gamepad into the Android device the Android device acts as a USB host. The term *USB On-the -Go* (or *USB OTG*) is used to describe a device’s ability to switch between host and client modes.

Note that when an Android device is acting as a USB host, it cannot simultaneously be charged.³ This means that when you have your Android tablet connected to the Logitech gamepads or when you have your Android phone connected to the Power Module or Legacy Module (with the phone acting as a USB host) the Android device *will be drawing power from its internal battery*. The Android device’s internal battery *currently cannot be charged while it is acting as a host for the gamepad or robot modules*.

USB Micro Type B

The Android devices have a special type of USB port called a USB On-the-Go (OTG) port. This USB OTG port allows the Android device to automatically switch between host and client modes. The USB OTG devices typically have a *USB Micro Type B* receptacle, so a micro USB-B cable is required:

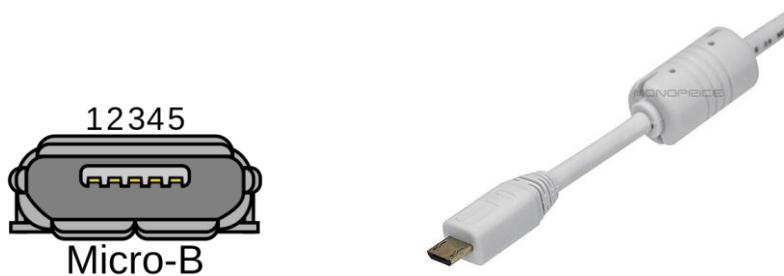


Figure 56 - USB Micro B receptacle and male cable.

³ From <http://electronics.stackexchange.com/questions/34741/can-an-android-tablet-serve-as-usb-host-and-be-charged-simultaneously-through-a> accessed January 15, 2015.

USB Mini Type B

The electronic modules from *Modern Robotics* that are used to communicate between the Android phone and the robot hardware (motors, servos and sensors) are equipped with *USB Mini Type B* receptacles. This type of receptacle was selected by the manufacturer because they are the most durable type of connector, designed for repeated connect and disconnect cycles. Mini B receptacles require a *USB Mini Type B* style cable

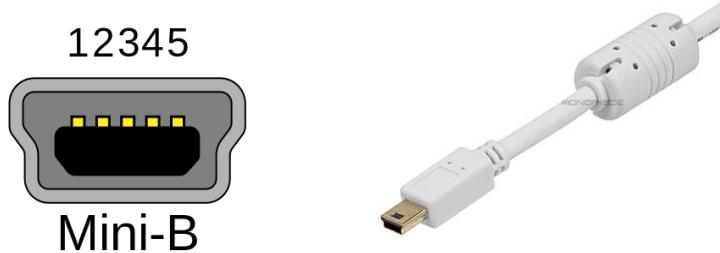


Figure 8 - USB Micro B receptacle & male cable.

Micro USB OTG Adapter

A special adapter cable is used to connect the Android device to the Power Module or to the USB hub. This Micro USB OTG Adapter cable has a male USB Micro B connector on one end and a female standard USB Type A receptacle on the other end.



Figure 9 - USB Micro OTG Adapter Cable.

When used with the driver station, the USB hub is connected to the Android tablet through the Micro OTG adapter cable. When used with the robot controller, the Power Module is connected to the smartphone through the Micro OTG adapter cable.

Appendix B: Preparing the ZTE Speed for use with FTC

There are several critical setup changes that need to be made to any Android device before it can be used for FTC. We need to first make sure that your Android device has the *Developer Options* and *USB Debugging* enabled. We also have to make sure that your Windows PC has the correct USB driver installed for the Android device that you are using.

Removing the SIM card from the ZTE Speed

Before you use your phone, you should physically remove the SIM card, and then place the phone into Airplane Mode (with WiFi still enabled) in order to make sure that the Speed phone does not try to connect to the Boost Mobile network whenever it is turned on.



Figure 57 - ZTE Speed

The first step in this process is to make sure that your phone is powered off. You will need to remove the plastic back cover of the phone. With the screen facing you, if you look in the lower right hand corner of the device, you will see a gap that you can use to pry the plastic rear cover off of the phone. Pry the back cover off of the phone and then place the phone screen down onto the table.

Along the right hand side of the phone, you should see the SIM card. Push in on the SIM card – this will eject the card partially from its slot. Remove the card and store it in a safe place. Replace the back cover and then power the phone on.



Figure 58 - There is a gap in lower right hand corner that you can use to pry the back cover off the phone.



Figure 59 - Pry the back cover off of the phone.



Figure 60 - The SIM card is installed in the right hand side.



Figure 61 - Push in on the SIM card to eject it from its slot.



Figure 62 - Remove the card from its slot and replace the back cover.

After you have removed the SIM card from the ZTE Speed phone, power the device on and step through the opening screens (see images below):

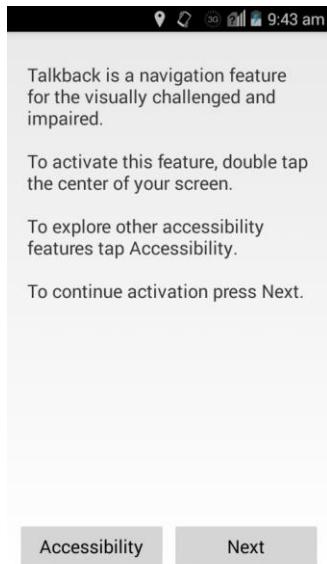


Figure 63 - The phone will ask you if you would like to set up Talkback. Hit Next to skip this step.

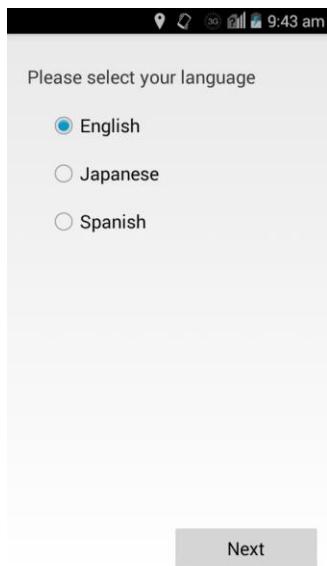


Figure 64 - Select your language and hit Next to continue.

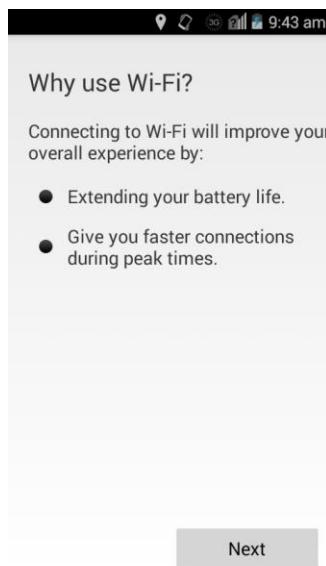


Figure 65 - The phone will prompt you to connect to a WiFi Network. Hit Next to continue.

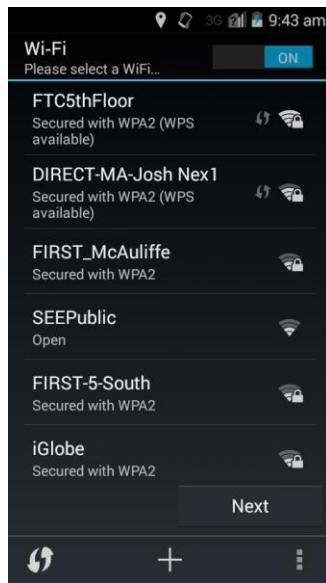


Figure 66 – You can connect to a WiFi network at this point or skip this step (hit Next).



Figure 67 - You might see this message indicating that the SIM card was not detected. Hit OK to continue.

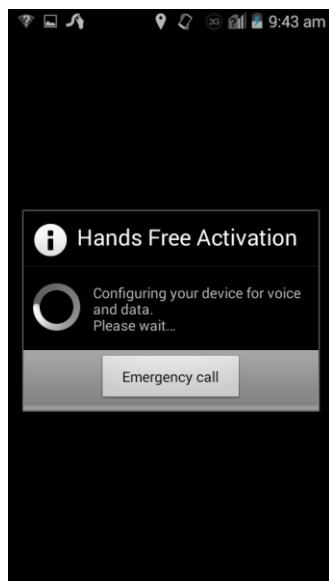


Figure 68 - The phone will display a Hands Free Activation screen.

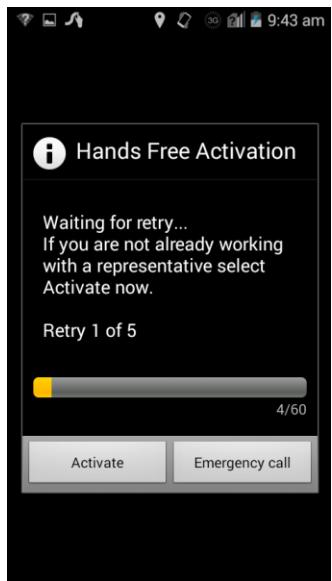


Figure 69 - The phone will try to activate itself. Hit the "Activate" button to continue.

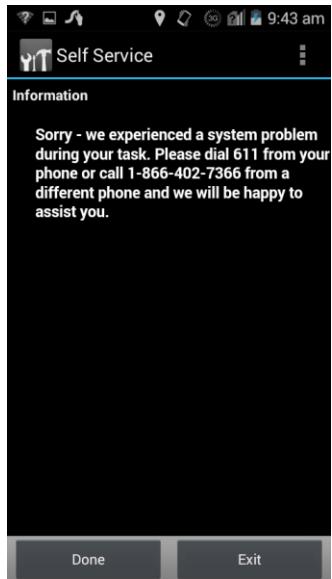


Figure 70 – The activation should fail. Hit “Done” to complete the process.



Figure 71 - You should now see your home screen. Press the Apps symbol to display the available apps on your phone.



Figure 72 - Find the Settings app and press it to launch the Settings activity.

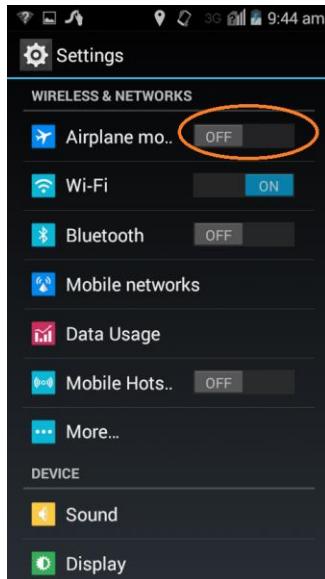


Figure 73 - Turn on Airplane Mode so the device won't try to connect to mobile network.

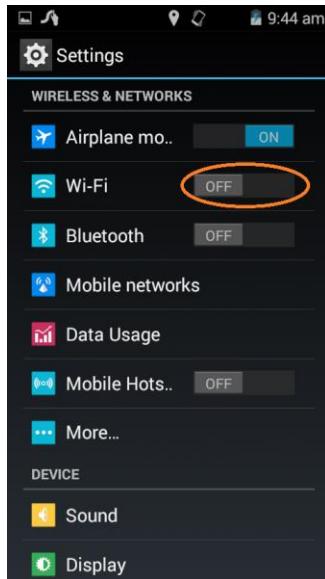


Figure 74 - Turn WiFi back on once the phone is in Airplane Mode.

Once you have removed the SIM card and switched the phone to Airplane Mode it is now ready for use.

Enabling Developer Options on your Android Device

For this training manual, we assume you will be using a ZTE Speed Android phone as your primary development device. In order to be able to be able to use this device with the Android Studio software, you will need to make sure the phone has “Developer Options” enabled and that the USB Debugging mode is enabled.

Detailed instructions on how to enable the Developer Options mode (and how to connect an Android device to Android Studio) are available on the Android developer website:

<http://developer.android.com/tools/device.html>

If your Android device is not yet in developer options mode, you can go to the settings screen of your Android device (touch the icon that looks like a cog or gear) and go to the **Settings->About phone** activity. Look for the **Build number** or in the case of the ZTE Speed look for the **SW Version** of your phone. Tap the Build number or SW Version number seven times to enable Developer Options on your device.

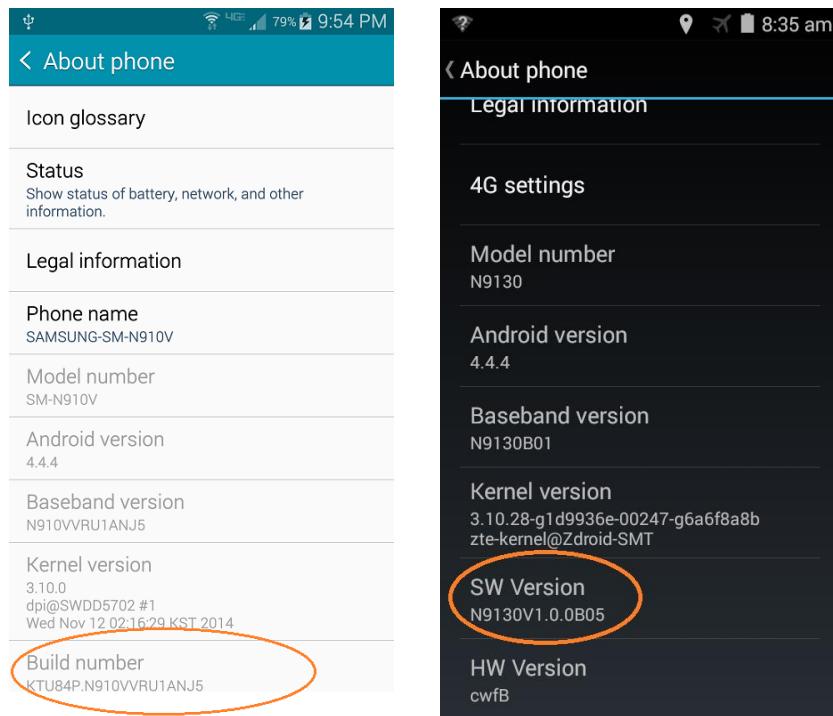


Figure 75 - Tap on the Build number (left) or SW Version number (right) 7 times to enable Developer Options.

Exit the Click the back button and look at your Settings screen. You should now see the “Developer options” item listed above “About phone” in the Settings menu.

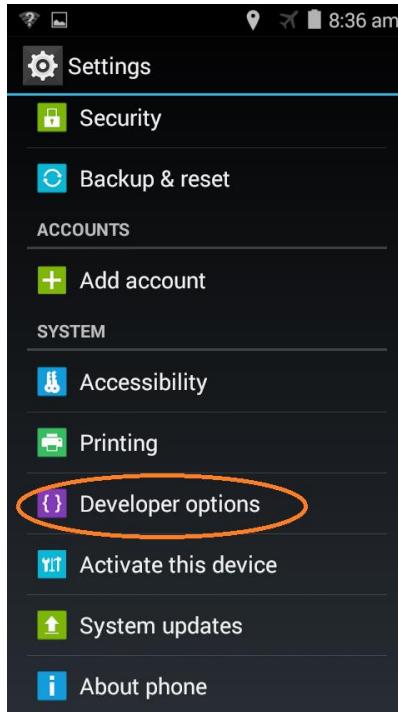


Figure 76 - Once Developer Options are enabled you should see a listing in the Settings menu.

Enabling USB Debugging

In order for your Android Studio computer to be able to “talk” to your Android phone properly, you will need to enable USB Debugging for your phone. Click on the “Developer options” item of the Settings menu (see **Error! Reference source not found.**). The Developer options screen should appear on your phone. Look for the “USB debugging” option and check the checkbox to enable this option on your phone:



Figure 77 - Make sure USB debugging is enabled for your phone.

Once the “USB debugging” option is enabled for your phone, when you connect your phone it will connect in debug mode, which will allow Android Studio to “talk” to your phone as an Android device.

Installing the USB Driver for your Device

If you’ve enabled “USB debugging” for your Android device, there is one last step that you need to do in order to connect your device successfully to Android Studio.

You will need to install the correct driver for your Android device onto your Windows computer. If your Windows computer does not have the correct driver for your phone installed, then when you connect your device to the computer via a USB cable, the Windows computer will think that the device is a media player or storage device (like a FLASH drive). You might be able to access files on your device in this mode, but your Android Studio software will not recognize the device as an available Android device that it can talk to.

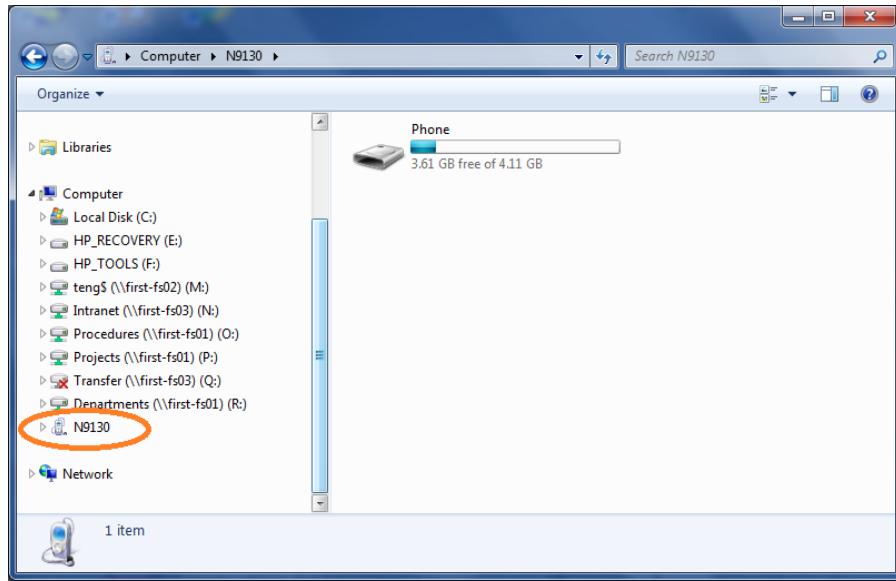


Figure 78 - If the proper driver is not installed, your PC might assume the device is a USB media player or storage device.

If you are able to install the correct USB driver onto your computer, then the next time you plug your Android device in with a USB cable, the computer and Android studio will recognize it as an Android device and Android Studio will be able to “talk” to it properly.

The ZTE Speed driver is conveniently included on the phone itself. When you first connect your ZTE speed to your computer using a USB cable, you should notice a USB symbol appear near the top of the screen (see figure below).



Figure 79 - When connected to PC, a USB icon should be visible near top of screen (in orange circle).

To display the **Connect to PC** activity, swipe downwards from the top of the touch screen and click on the “Touch for other USB options” item.

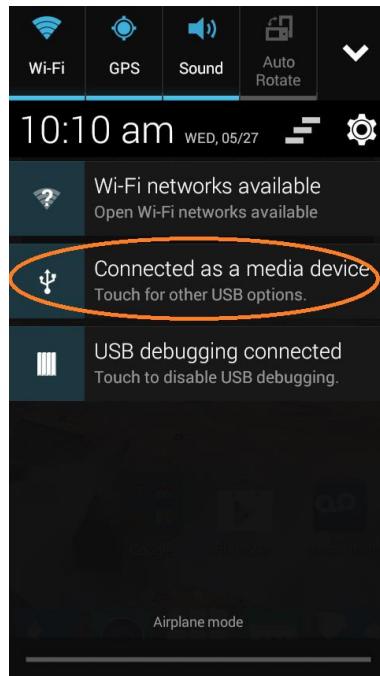


Figure 80 - Swipe downwards and select "Touch for other USB options"

To install the driver, select the “Install driver” option from the screen.

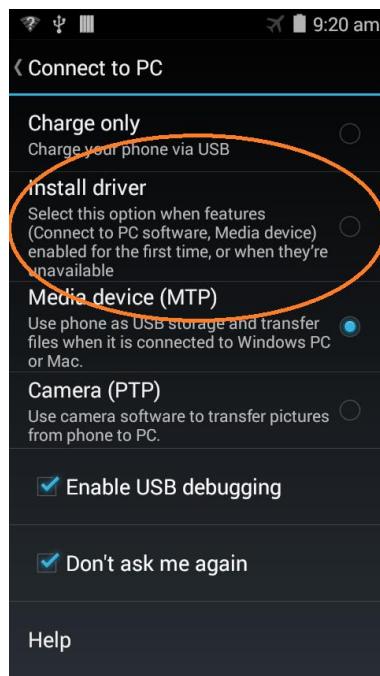


Figure 81 - Select the "Install driver" option.

Once you select the “Install driver” option on the phone your PC should prompt you if you want to run the install program (“AutoRun.exe”). Select the “Run AutoRun.exe” on the PC dialog box to start the installation.



Figure 82 - Select Run AutoRun.exe option to start installation.

The install wizard should prompt you for the Setup Language. Choose your language and continue with the installation.

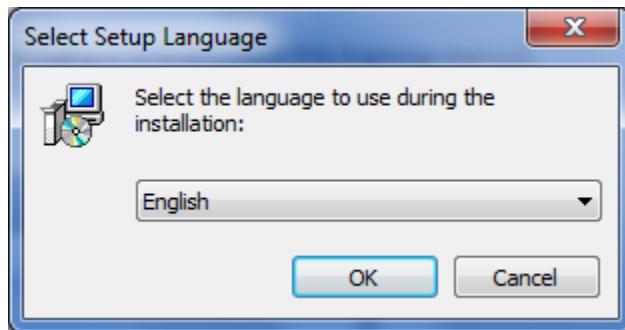


Figure 83 - Select language.



Figure 84 - Follow the on screen instructions to install the driver.

Follow the instructions on the setup wizard screens to install the ZTE handset USB driver. The setup wizard will display a progress bar showing the status of the install. When the install is complete, click on the “Finish” button to end the installation process.



Figure 85 - Click on Finish to complete installation.

After installing the ZTE Speed driver, you want to go back to your ZTE phone and go to the **Connect to PC** activity again (swipe downwards from the top on the main screen and select the “Touch for other USB options”). Select the “Charge only” option from the list of options.

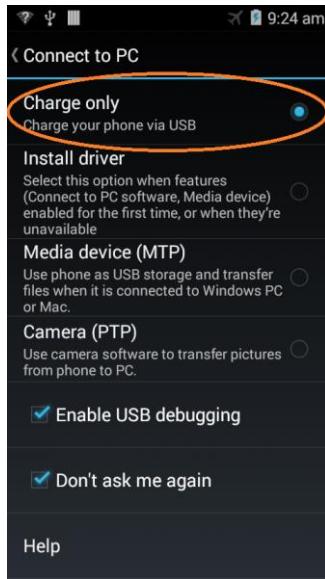


Figure 86 - Select the Charge only option.

If have successfully installed the USB driver on your PC, then your ZTE Speed phone should prompt you on whether or not to allow USB debugging by your PC.

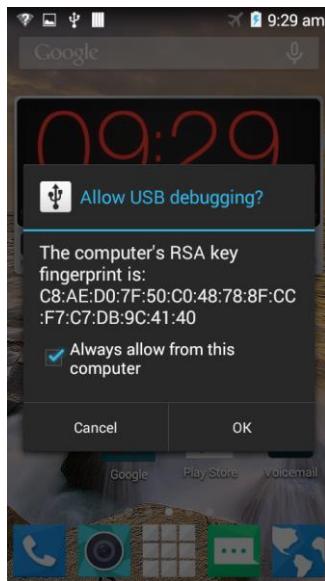


Figure 87 - Select the "Always allow" option and hit OK.

Note that if you do not see this prompt, you can unplug your phone from the USB cable, wait a few seconds, and then reconnect your phone to the cable. You should now see the prompt. Select “Always allow from this computer” option and hit “OK” to allow USB debugging access for your PC.

Once you have successfully installed the driver and configured your ZTE Speed phone, you should see the phone appear as an available Android device in the Android panel of Android Studio.

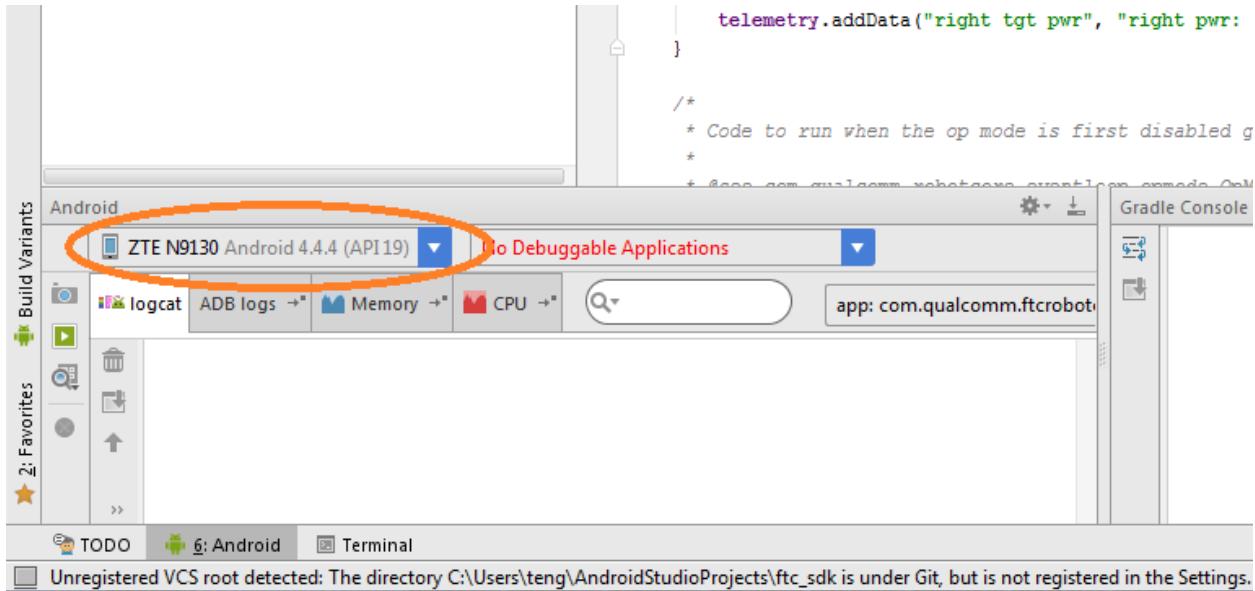


Figure 88 - Device should be visible in the Android panel of Android Studio.

Once you have installed your driver successfully, you are ready to build and install your first app!

Renaming the ZTE Speed

Before you compete, you also should rename your ZTE Speed devices (both the Robot Controller and Driver Station devices). Please consult the FTC Game Manual Part 1 (rule <RS02>) on the naming convention that should be applied to your phones. In order to change the name on your device, launch the Android **Settings** menu again and click on the **Wi-Fi** item to launch the Wi-Fi Settings screen.



Figure 89 - Click on Wi-Fi to launch Wi-Fi Settings.

Click on the three dots in the lower right hand corner of the WiFi Settings window to display a pop-up menu.

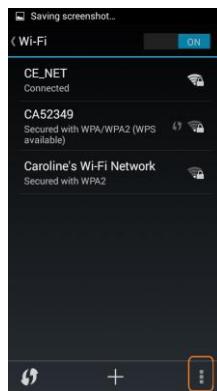


Figure 90 - Click on three dots to bring up pop-up menu.

Select the **Wi-Fi Direct** item from the pop-up menu.

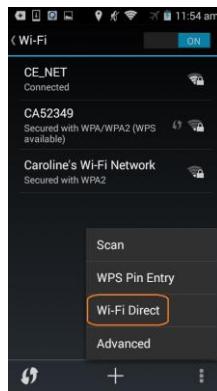


Figure 91 - Select Wi-Fi Direct.

Click on the **RENAME DEVICE** button in the Wi-Fi Direct settings screen.



Figure 92 - Click on the RENAME DEVICE button.

Specify the new name and click on **OK** to save the name.

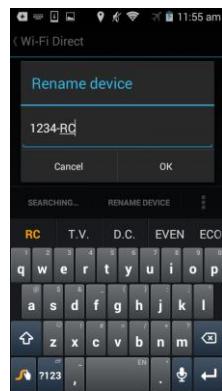


Figure 93 - Specify the new name and click on OK to save the name.

After you have changed your device's name, you might want to power cycle your Android device. Also, you will most likely need to re-pair the Driver Station device to the Robot Controller device after a name change.