



PRESENTED BY **Qualcomm**

---

*FIRST® GAME CHANGERS<sup>SM</sup> powered by Star Wars: Force for Change*  
2020-2021 *FIRST® Tech Challenge*

# Android Studio Guide



[FIRSTINSPIRES.ORG/ROBOTICS/FTC](http://FIRSTINSPIRES.ORG/ROBOTICS/FTC)

© & ™ 2020 Lucasfilm Ltd.

## Sponsor Thank You

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

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



***FIRST*® Tech Challenge  
Official 3D Augmented Reality/  
Virtual Reality Sponsor**



Revision History		
Revision	Date	Description
1	06/29/2020	Initial Release

## Contents

Contents .....	3
1. Introduction.....	6
1.1. What is FIRST® Tech Challenge?.....	6
2. <i>Gracious Professionalism</i> ® .....	6
3. Introduction to Android Studio .....	7
4. The FTC Control System.....	7
4.1. About the FIRST Tech Challenge.....	7
4.2. Autonomous vs. Driver-Controlled.....	8
4.3. Point-to-Point Control System .....	8
4.4. REV Robotics Expansion Hub.....	9
4.5. REV Robotics Control Hub .....	10
4.6. What's an Op Mode?.....	10
5. Required Materials .....	11
6. Using Your Android Device.....	17
6.1. Unlocking Your Screen.....	17
6.2. Navigating in Android .....	19
6.3. Displaying Available Apps on your Android Phone .....	21
6.3.1. Android Marshmallow Users.....	21
6.3.2. Android Nougat Users .....	21
7. Configuring Your Android Devices.....	22
7.1. What Needs to Be Configured for My Control System? .....	22
7.1.1. Control Hub Users.....	22
7.1.2. Users with Two Android Smartphones.....	23
7.2. Renaming Your Smartphones .....	23
7.3. Installing the FTC Apps .....	31
7.4. Placing Phones into Airplane Mode with Wi-Fi On.....	39
7.5. Pairing the Driver Station to the Robot Controller .....	41
7.5.1. Control Hub Users.....	41
7.5.2. Users with Two Android Smartphones.....	53
8. Connecting Devices to a Control or Expansion Hub .....	54

8.1.	Connecting 12V Power to the Hub.....	54
8.2.	Connecting a Motor to the Hub.....	58
8.3.	Connecting a Servo to the Hub.....	59
8.4.	Connecting a Color-Distance Sensor to the Hub .....	61
8.5.	Connecting a Touch Sensor to the Hub.....	63
9.	Configuring Your Hardware .....	65
9.1.	Before You Begin... ..	65
9.2.	Connecting an Android Smartphone to an Expansion Hub .....	65
9.3.	Getting the Control Hub Ready.....	68
9.4.	Creating a Configuration File Using the Driver Station.....	68
9.5.	Configuring a DC Motor.....	74
9.6.	Configuring a Servo.....	78
9.7.	Configuring a Color Distance Sensor.....	82
9.8.	Configuring a Digital Touch Sensor .....	87
9.9.	Saving the Configuration Information.....	90
10.	Installing Android Studio .....	97
10.1.	Android Developer Website .....	97
10.2.	System Requirements.....	97
10.3.	Java Development Kit.....	97
10.4.	Downloading and Installing Android Studio .....	97
11.	Disabling Android Studio Instant Run.....	99
11.1.	Introduction .....	99
11.2.	Locating Instant Run Settings .....	99
11.3.	Additional Information .....	101
12.	Downloading the Android Studio Project Folder .....	101
12.1.	Downloading the Android Studio Project Folder .....	101
12.2.	Extracting the Contents of the Archived Project File.....	103
	Importing the FTC Project into Android Studio .....	106
13.	Enabling Developer Options .....	108
14.	Creating and Running an Op Mode (Android Studio) .....	110
14.1.	TeamCode Module .....	110
14.2.	Javadoc Reference Information .....	111
14.3.	Enabling Auto Import .....	111
14.4.	Sample Op Modes .....	112
14.5.	Creating Your FIRST Op Mode.....	113
14.6.	Examining the Structure of Your Op Mode.....	115



14.7.	Building and Installing Your Op Mode .....	118
14.8.	Running Your Op Mode .....	122
14.9.	Modifying Your Op Mode to Control a Motor .....	126
	Running Your Op Mode with a Gamepad Connected.....	128
15.	Controlling a Servo (Android Studio).....	131
15.1.	What is a Servo Motor?.....	131
15.2.	Modifying Your Op Mode to Control a Servo.....	131
16.	Using Sensors (Android Studio).....	134
16.1.	Color-Distance Sensor .....	134
16.2.	Touch Sensor .....	134
17.	Appendix A – Resources.....	136
	Game Forum Q&A .....	136
	FIRST Tech Challenge Game Manuals.....	136
	FIRST Headquarters Pre-Event Support.....	136
	FIRST Websites.....	136
	FIRST Tech Challenge Social Media .....	136
	Feedback.....	136

## 1. Introduction

---

### **What is FIRST® Tech Challenge?**

FIRST® Tech Challenge is a student-centered program that focuses on giving students a unique and stimulating experience. Each year, teams engage in a new game where they design, build, test, and program autonomous and driver operated robots that must perform a series of tasks. To learn more about FIRST® Tech Challenge and other FIRST® Programs, visit [www.firstinspires.org](http://www.firstinspires.org).

### **FIRST Core Values**

We express the FIRST® philosophies of *Gracious Professionalism*® and *Coopertition*® through our Core Values:

- **Discovery:** *We explore new skills and ideas.*
- **Innovation:** *We use creativity and persistence to solve problems.*
- **Impact:** *We apply what we learn to improve our world.*
- **Inclusion:** *We respect each other and embrace our differences.*
- **Teamwork:** *We are stronger when we work together.*
- **Fun:** *We enjoy and celebrate what we do!*

## 2. Gracious Professionalism®

---

FIRST® uses this term to describe our programs' intent.

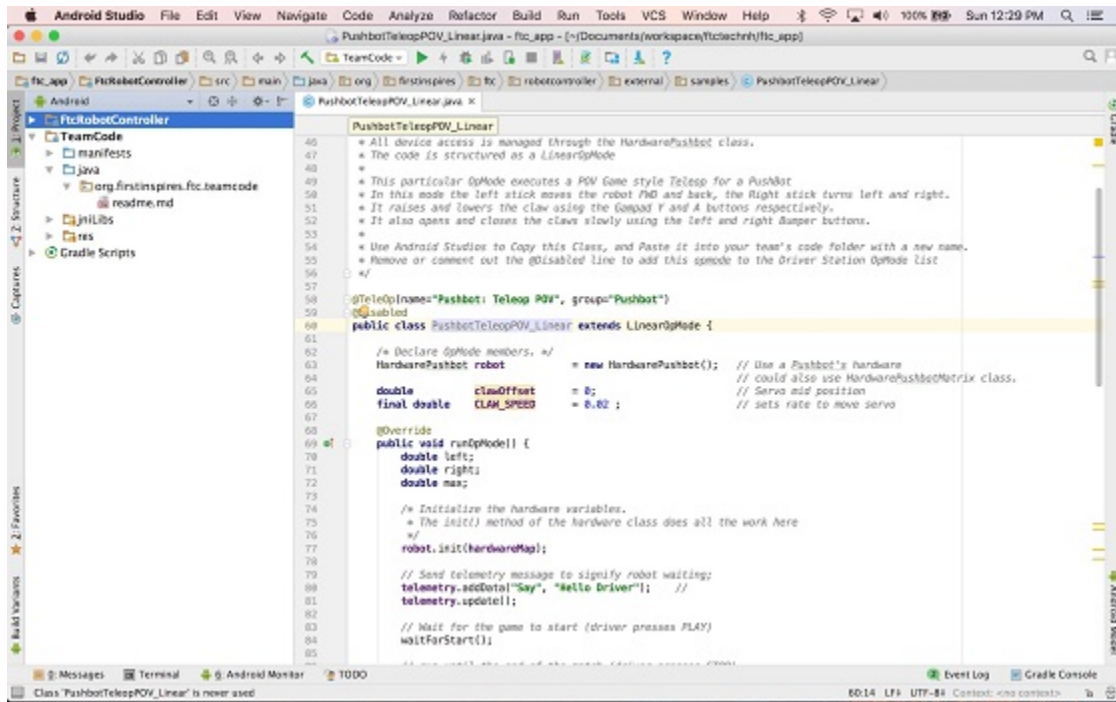
*Gracious Professionalism*® is a way of doing things that encourages high-quality work, emphasizes the value of others, and respects individuals and the community.

Watch Dr. Woodie Flowers explain *Gracious Professionalism* in this [short video](#).

### 3. Introduction to Android Studio

This tutorial will take you step-by-step through the process of configuring, programming, and operating your Control System. This tutorial uses Android Studio to help you get started programming your robot.

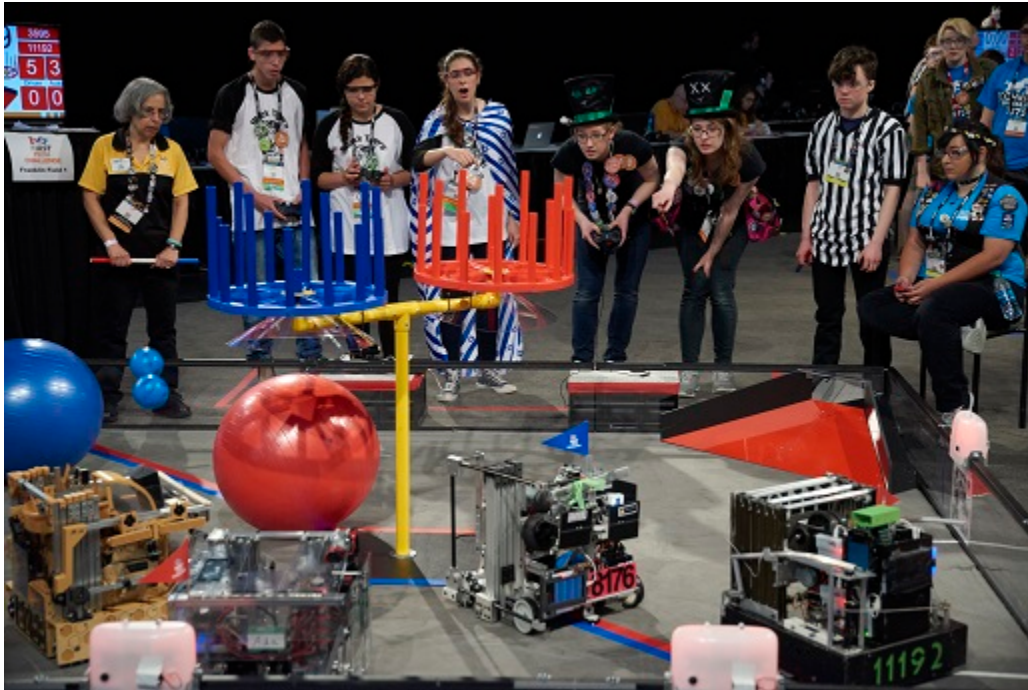
Android Studio is an advanced integrated development environment for creating Android apps. This tool is the same tool that professional Android app developers use. Android Studio is only recommended for **advanced users** who have **extensive Java programming experience**.



### 4. The FTC Control System

#### 4.1. About the FIRST Tech Challenge

The FIRST Tech Challenge seeks to inspire youth to become the next generation of STEM leaders and innovators through participation in mentor-guided robotics competition. Teams who participate in the FIRST Tech Challenge must build a robot that performs a variety of tasks. The tasks vary from season to season, and are based on a set of game rules that are published at the start of each season. The more tasks that a robot can complete, the more points a team will earn.



(Photo courtesy of Dan Donovan, ©2017 Dan Donovan / www.dandonovan.com)

#### 4.2. Autonomous vs. Driver-Controlled

A FIRST Tech Challenge match has an *autonomous* phase and a *driver-controlled* or "*tele-operated*" phase. In the autonomous phase of a match the robot operates without any human input or control. In the driver-controlled phase, the robot can receive input from up to two human drivers.

#### 4.3. Point-to-Point Control System

The FIRST Tech Challenge uses Android devices to control its robots. During a competition, each team has two Android devices.



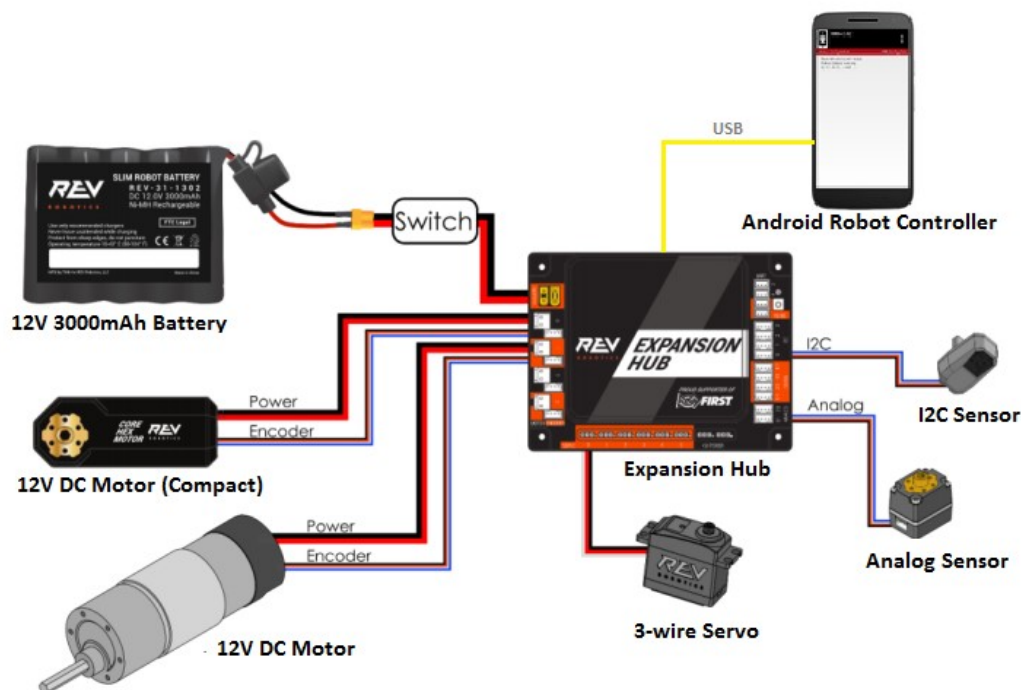
One Android device is mounted onto the robot and is called the *Robot Controller*. The Robot Controller acts as the “brains” of the robot. It does all of the thinking for the robot and tells the robot what to do. It consists of an Android device running an FTC Robot Controller app. There are two hardware options currently being used: REV Robotics Expansion Hub or the REV Robotics Control Hub.

A second Android device sits with the team drivers and has one or two gamepads connected. This second device is known as the *Driver Station*. The Driver Station is sort of like a remote control that you might use to control your television. The Driver Station allows a team to communicate remotely (using a secure, wireless connection) to the Robot Controller and to issue commands to the Robot Controller. The Driver Station consists of an Android device running an FTC Driver Station app.

#### 4.4. REV Robotics Expansion Hub

The REV Robotics Expansion Hub is the electronic input/output (or “I/O”) module that lets the Robot Controller talk to the robot’s motors, servos, and sensors. The Robot Controller communicates with the Expansion Hub through a serial connection. For the situation where an Android smartphone is used as the Robot Controller, a USB cable is used to establish the serial connection. For the situation where a REV Robotics Control Hub is used, an internal serial connection exists between the built-in Android device and the Expansion Hub.

The Expansion Hub is also connected to a 12V battery which is used to power the Expansion Hub, the motors, the servos and sensors. If an Android smartphone is used as the Robot Controller, then the smartphone will have its own independent battery. If a REV Robotics Control Hub is used as the Robot Controller, then the Control Hub will use the main 12V battery to power its internal Android device.



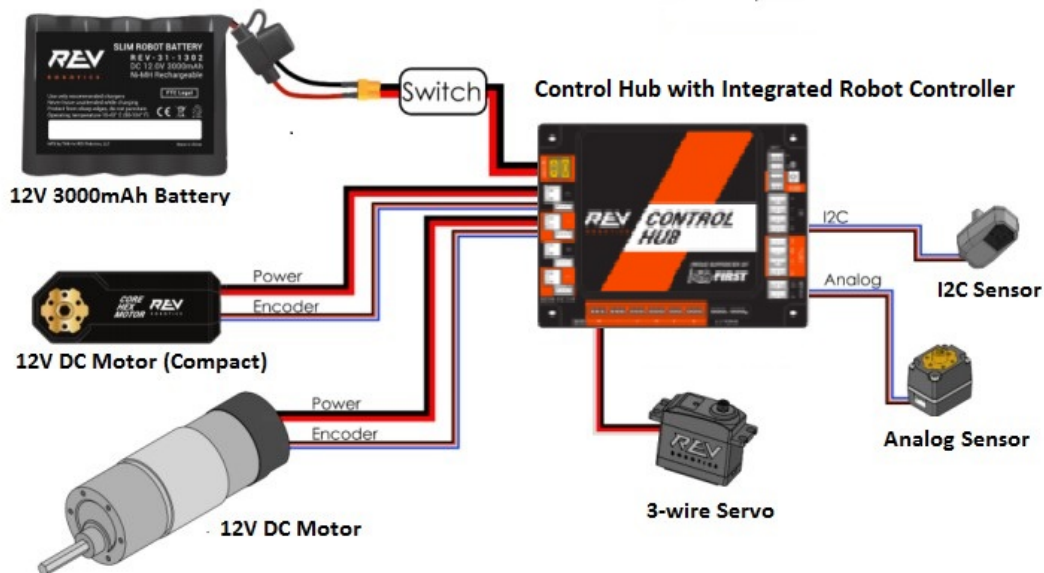


#### 4.5. REV Robotics Control Hub

The Control Hub is an integrated version of the Robot Controller. It combines an Android device built into the same case as a REV Robotics Expansion Hub.



The Control Hub, which has its built-in Android device connected directly to the Expansion Hub using an internal serial bus, eliminates the need for an external USB connection between the Android Robot Controller and the I/O module.



#### 4.6. What's an Op Mode?



During a typical FIRST Tech Challenge match, a team's robot has to perform a variety of tasks in an effort to score points. For example, a team might want their robot to follow a white line on the competition floor and then score a game element (such as a ball) into a goal autonomously during a match. Teams write "op modes" (which stand for "operational modes") to specify the behavior for their robot.

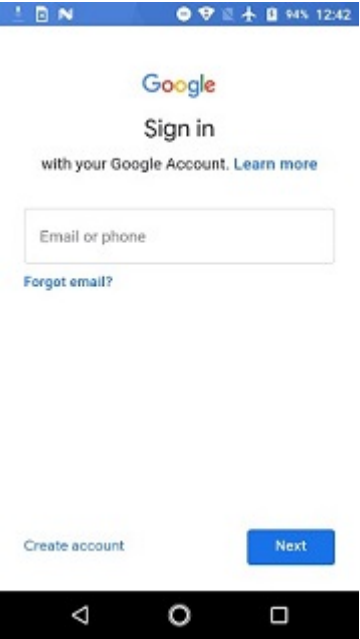


*Op modes* are computer programs that are used to customize the behavior of a competition robot. The Robot Controller can *execute* a selected *op mode* to perform certain tasks during a match.

Teams who are participating in the *FIRST* Tech Challenge have a variety of programming tools that they can use to create their own *op modes*. Teams can use a visual ("drag and drop") programming tool called the *FTC Blocks Programming Tool* to create their *op modes*. Teams can also use a text-based Java tool known as the *FTC OnBot Java Programming Tool* or Google's *Android Studio* integrated development environment (also known as an "IDE") to create their *op modes*.

## 5. Required Materials



This wiki contains tutorials that demonstrate how to configure, program, and operate the FTC control system. In order to complete the tutorials, you will need to have the following materials available:

Required Item(s)	Image
<p>Two (2) <i>FIRST</i>-approved* Android smartphones.</p> <p>Or...</p> <p>One (1) Control Hub and one (1) <i>FIRST</i>-approved* Android smartphone.</p> <p><small>*For a list of <i>FIRST</i>-approved Android smartphones, refer to the current FTC Game Manual Part 1, rule &lt;RE06&gt;.</small></p>	 <p>Or...</p> 





Required Item(s)	Image
<p>A Google account (available for free from Google) to access the Google Play store.</p>	
<p>Wireless Internet access.</p>	
<p>Laptop with Microsoft Windows 7, 8 or 10 and Wi-Fi capability.</p> <p>Note that your laptop should have the most current service packs and system updates from Microsoft.</p> <p>If you are using a different type of machine (such as a Chromebook, Android Tablet, etc.) as your programming device, the steps might differ slightly on how to access the Programming Server on the Robot Controller. Refer to your device's user documentation for details on how to connect to a Wi-Fi network.</p>	



Required Item(s)	Image
<p>Javascript-enabled web browser (Google Chrome is the recommended browser).</p>	
<p>If you are using a smartphone as your Robot Controller, you will also need a REV Robotics Expansion Hub (REV-31-1153) to connect to the motors, servos, and sensors. Control Hub users will use the integrated Expansion Hub.</p>	
<p>REV Robotics Switch, Cable, &amp; Bracket (REV-31-1387).</p>	

Required Item(s)	Image
<p>If you are using an approved 12V battery that has a Tamiya connector (like the Tetrax W39057 battery) you will need a &gt;REV Robotics Tamiya to XT30 Adapter Cable (REV-31-1382).</p> <p>If you have a REV Robotics Slim Battery (REV-31-1302) then you will not need this adapter since the REV battery already has an XT30 connector.</p>	
<p><i>FIRST</i>-approved* 12V Battery (such as Tetrax W39057 or REV Robotics REV-31-1302).</p> <p><small>*For a list of <i>FIRST</i>-approved 12V batteries, refer to the current FTC Game Manual Part 1, rule &lt;RE03&gt;.</small></p>	 <p>Or...</p>

Required Item(s)	Image
	 <p>A black rectangular battery with a yellow connector on one end and a black Anderson-style connector on the other. The label on the battery reads: "REV ROBOTICS SLIM ROBOT BATTERY REV-31-1302 DC 12.0V 3000mAh Ni-MH Rechargeable". It also includes safety warnings and regulatory marks like CE and FTC Legal.</p>
<p>FIRST-approved* 12V DC Motor (such as Tetrax W39530, with power cable W41352).</p> <p><small>*For a list of FIRST-approved 12V motors, refer to the current FTC Game Manual Part 1, rule &lt;RE09&gt;.</small></p>	 <p>A silver cylindrical motor with a brass shaft and a red braided power cable with a green PCB and red/black wires.</p>
<p>REV Robotics Anderson to JST VH Cable (REV-31-1381).</p>	 <p>A red braided cable with a black Anderson connector on one end and a silver JST VH connector on the other.</p>

Required Item(s)	Image
180-Degree Standard Scale Servo (such as Hitec HS-485HB).	
REV Robotics Color Sensor with 4-Pin Cable (REV-31-1154).	
REV Robotics Touch Sensor with 4-Pin Cable (REV-31-1425).	
<p>If you are using a smartphone as your Robot Controller, you will need a USB Type A male to type mini-B male cable.</p> <p>Control Hub users do not need this cable.</p>	

Required Item(s)	Image
<p>If you are using a smartphone as your Robot Controller, you will need two (2) micro USB OTG adapters.</p> <p>If you are using a Control Hub as your Robot Controller, you will need one (1) micro USB OTG adapter.</p>	
<p>Logitech F310 USB Gamepad.</p>	

## 6. Using Your Android Device

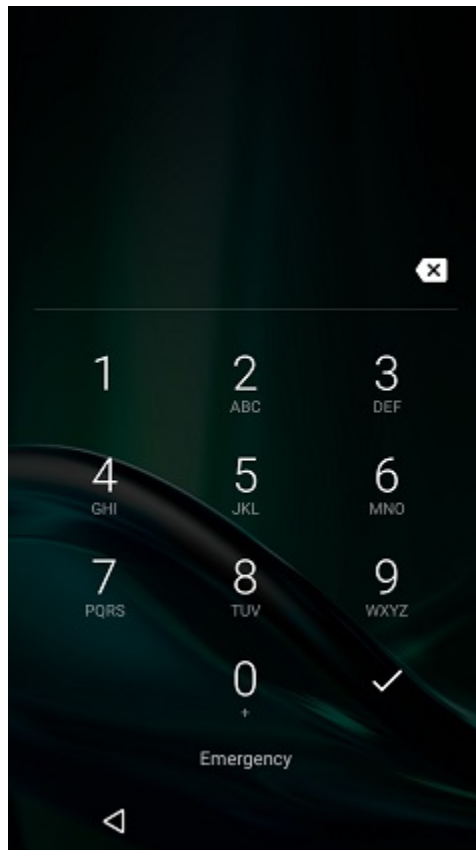
Before you get started with your control system, it is helpful if you familiarize yourself with the basic operation of your Android device.

### 6.1. Unlocking Your Screen

When you first power on an Android phone, it usually starts off with the screen in a "locked" state. For the Motorola smartphones that are used in the *FIRST* Tech Challenge, you must touch the locked screen and then slide your finger upwards along the screen to unlock the phone. Note that different devices might require a slightly different procedure to unlock the screen.



Depending on your security settings, you might be challenged for a pass code or PIN number. Use the touch screen to enter in the pass code or PIN value and tap on the check mark to log into the device.



## 6.2. Navigating in Android

Your phone should display its home screen if you just powered it on and unlocked it. Note that the actual screens on your smartphone might differ slightly from the screens depicted in this tutorial.



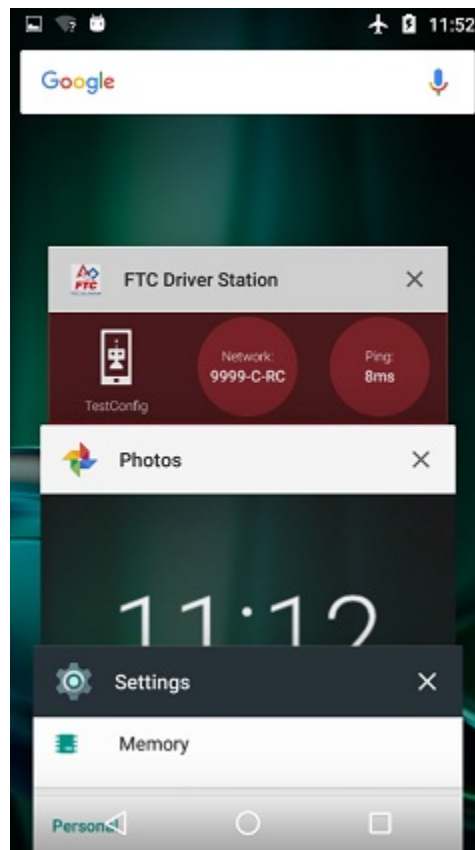
At the bottom of the screen there should be some buttons that you can use to navigate the screens on your Android device.



The leftmost button (see image above) is the "Back" button. You can use this button to return to the previous screen on your Android device.

The center button is the "Home" button. Pressing this button should take you back to the home or opening screen of your Android device.

The rightmost button is the "Recent Apps" button. If you click on this button it will display the apps that were recently run and are dormant in the background. You can close a recent app by tapping the "X" button on the app's listing.



Note that some Android smartphones have an auto-hide feature which automatically hides the bottom navigation buttons. If your smartphone has this feature, you might need to swipe up from the bottom of the screen to display the navigation buttons.



## 6.3. Displaying Available Apps on your Android Phone

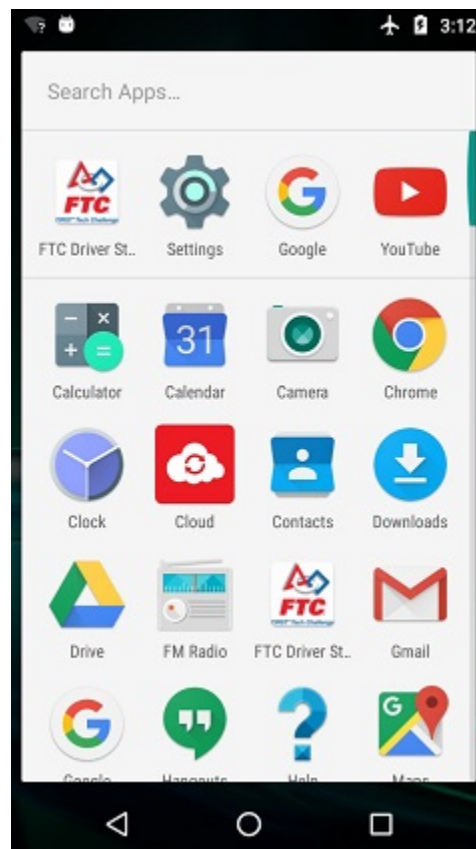
### 6.3.1. Android Marshmallow Users

If you are using a device with Android Marshmallow (6.x) or earlier, you can display the available apps using the *Android App Drawer* button that is available on the home screen.

There should be another row of buttons visible above the "Back", "Home" and "Recent Apps" buttons. In the center of this row of buttons is a button that has an array of dots or squares.

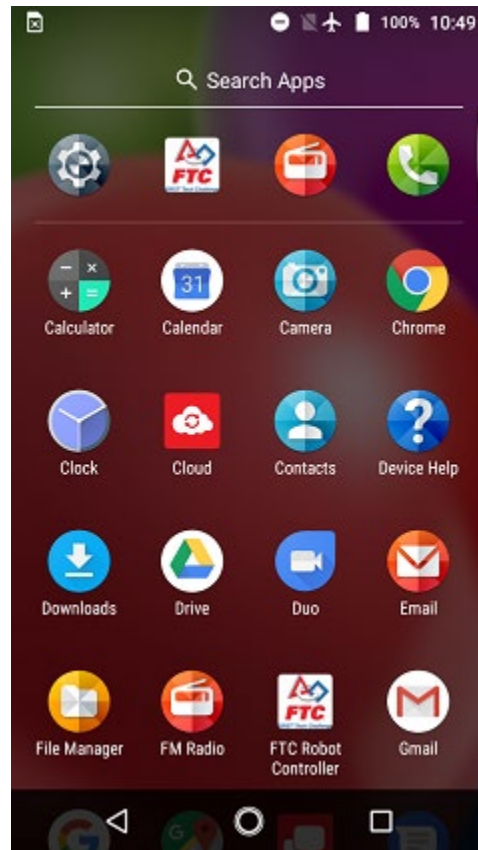


Tapping on this button will launch the *Android App Drawer*. The App Drawer displays a list of all of the apps that are available on your Android device. You can scroll through the App Drawer screens to find and launch an app.



### 6.3.2. Android Nougat Users

If you are using a device with Android Nougat (7.x) or newer, you can display the available apps by simply swiping upwards from the bottom of the touchscreen. Newer versions of Android no longer have the *App Drawer* feature.



## 7. Configuring Your Android Devices

---

### 7.1. What Needs to Be Configured for My Control System?

#### 7.1.1. Control Hub Users

Teams who are using a Control Hub with the integrated Robot Controller will only need to configure a single smartphone for use as a Driver Station. The process is as follows:

- Rename the smartphone to "<TEAM NUMBER>-DS" (where <TEAM NUMBER> is replaced by your team number).
- Install the Driver Station app onto the Driver Station phone.
- Put your phone into Airplane Mode (with the WiFi radio still on).
- Pair (i.e., wirelessly connect) the Driver Station to the Control Hub.



**IMPORTANT NOTE:** Eventually the Control Hub will need to be renamed so that its name complies with Game Manual rule < RS01>, but for now we will use the Control Hub with its default name. You can learn how to manage a Control Hub (and modify its name, password, etc.) in [this tutorial](#).

### 7.1.2. Users with Two Android Smartphones

Teams who have two smartphones and are not using a Control Hub will need to configure one smartphone for use as a Robot Controller and a second smartphone for use as a Driver Station. The process is as follows,

- Rename one smartphone to "<TEAM NUMBER>-RC" (replace <TEAM NUMBER> with your team number).
- Install the Robot Controller app onto the Robot Controller phone.
- Rename a second smartphone to "<TEAM NUMBER>-DS" (where <TEAM NUMBER> is replaced by your team number).
- Install the Driver Station app onto the Driver Station phone.
- Put your phones into Airplane Mode (with the WiFi radios still on).
- Pair (i.e., wirelessly connect) the Driver Station to the Robot Controller.



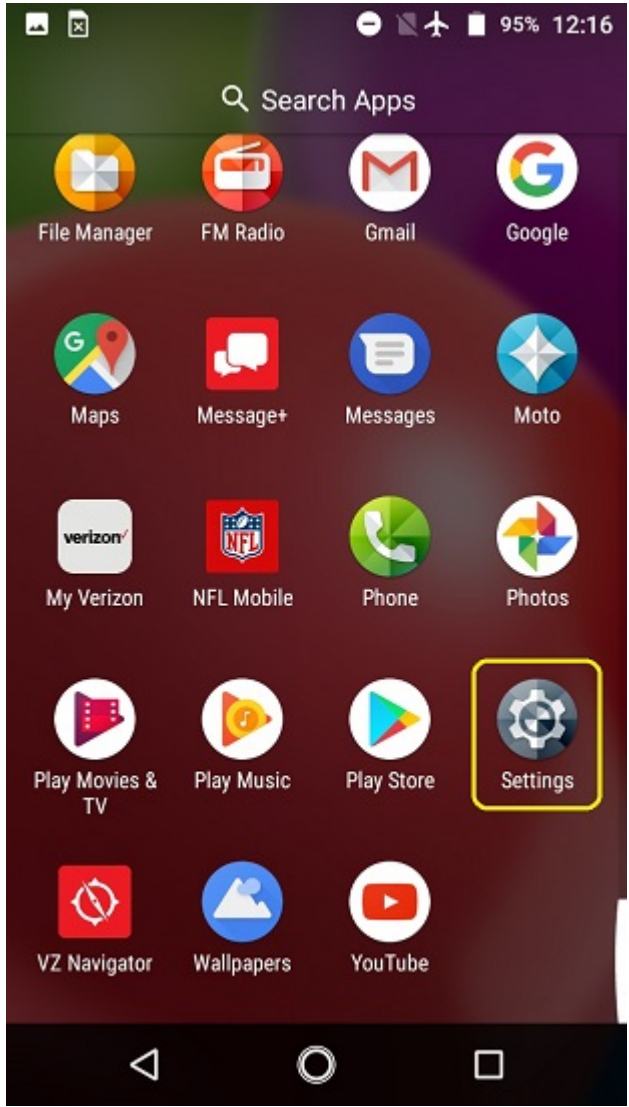
## 7.2. Renaming Your Smartphones

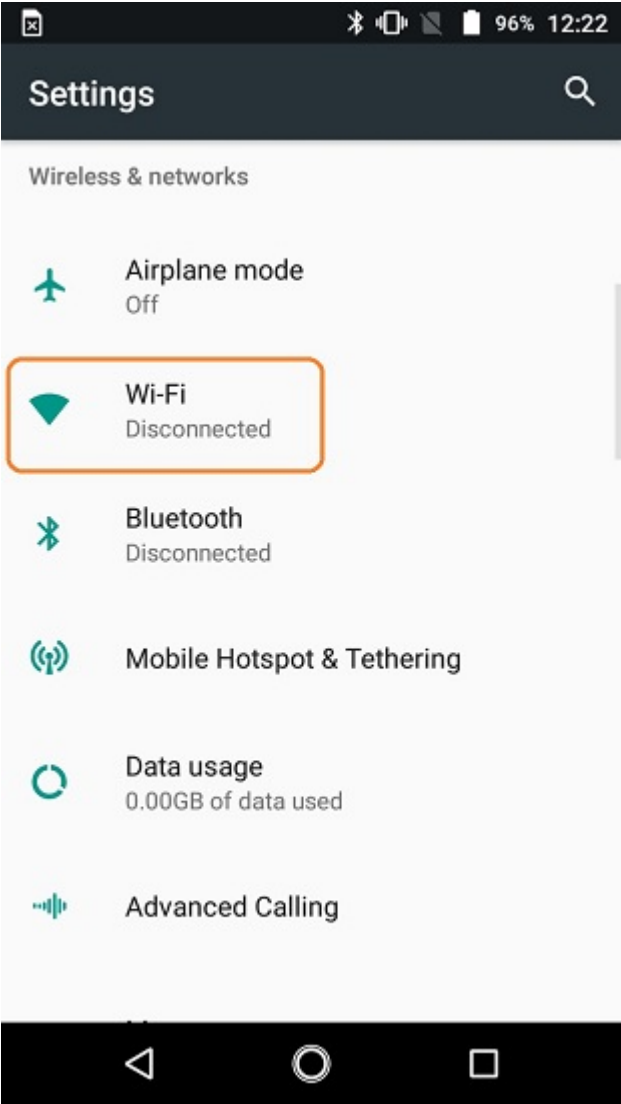
The official rules of the FIRST Tech Challenge (see <RS01>) require that you change the Wi-Fi name of your smartphones to include your team number and “-RC” if the phone is a Robot Controller or “-DS” if it is a Driver

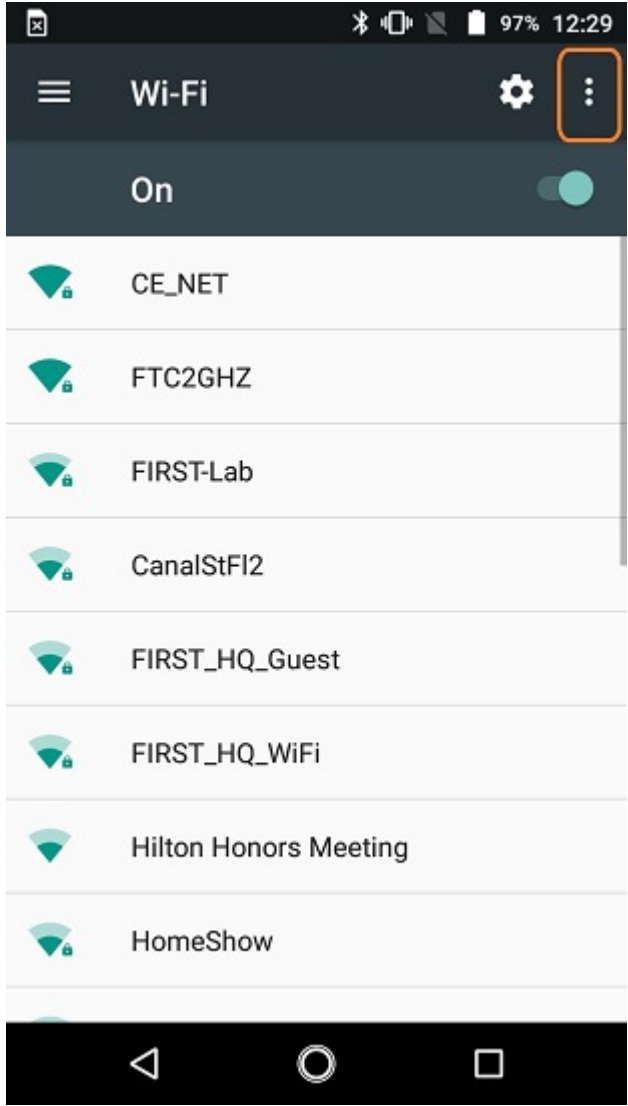
Station. A team can insert an additional dash and a letter (“A”, “B”, “C”, etc.) if the team has more than one set of Android phones.

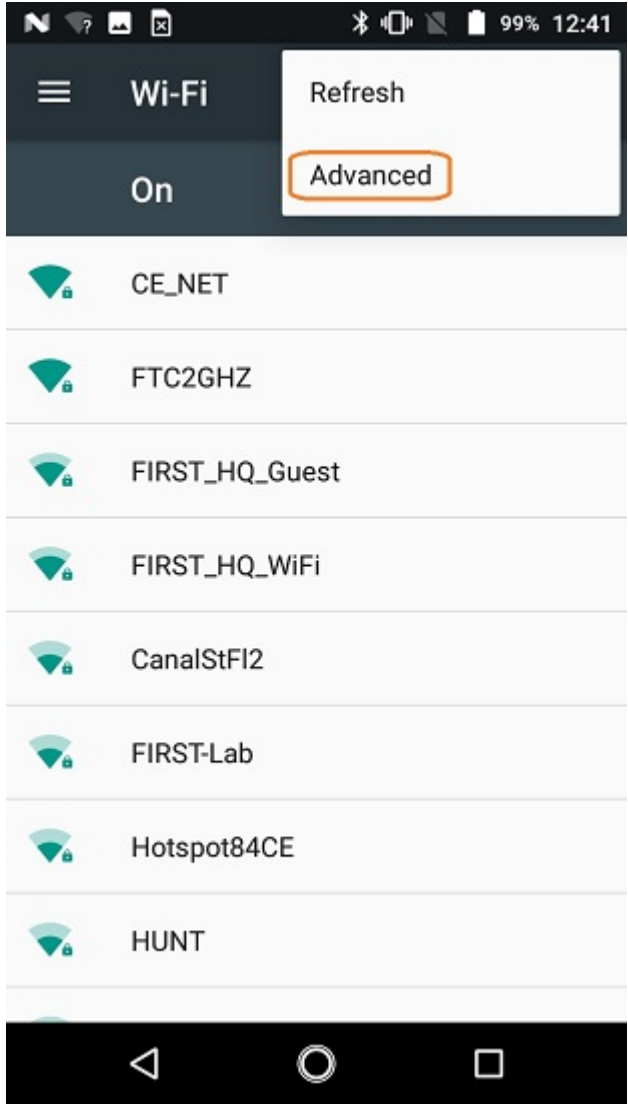
If, for example, a team has a team number of 9999 and the team has multiple sets of phones, the team might decide to name one phone “9999-C-RC” for the Robot Controller and the other phone “9999-C-DS” for the Driver Station. The “-C” indicates that these devices belong to the third set of phones for this team.


**NOTE:** it will take an estimated 5 minutes per phone to complete this task.

Step	Image
<p>1. Browse the list of available apps on the smartphone and locate the <b>Settings</b> icon. Click on <b>Settings</b> icon to display the Settings screen.</p>	 <p>The image shows a smartphone home screen with a grid of app icons. The icons include File Manager, FM Radio, Gmail, Google, Maps, Message+, Messages, Moto, My Verizon, NFL Mobile, Phone, Photos, Play Movies &amp; TV, Play Music, Play Store, and Settings. The Settings icon, which is a grey gear, is highlighted with a yellow rectangular box. The status bar at the top shows 95% battery and the time 12:16. The navigation bar at the bottom shows the back, home, and recent apps buttons.</p>

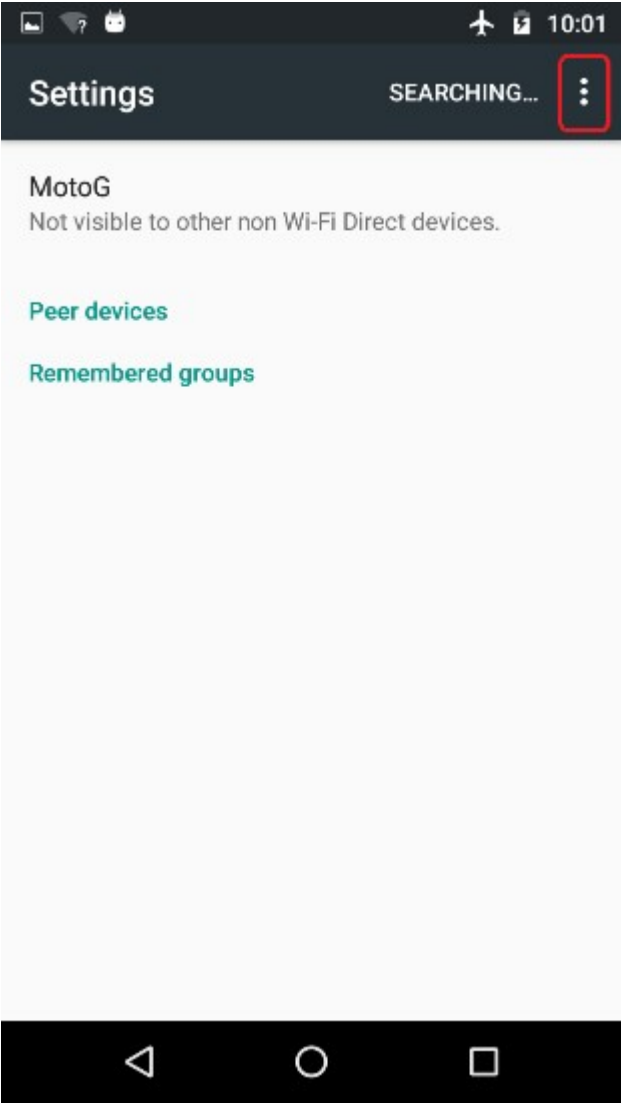
Step	Image
<p>2. Click on <b>Wi-Fi</b> to launch the Wi-Fi screen.</p>	 <p>The image shows a screenshot of the Android Settings application. At the top, the status bar displays the time as 12:22 and the battery level at 96%. The main header of the settings page is 'Settings' with a search icon on the right. Below this, the section 'Wireless &amp; networks' is visible. It contains several options: 'Airplane mode' (Off), 'Wi-Fi' (Disconnected), 'Bluetooth' (Disconnected), 'Mobile Hotspot &amp; Tethering', 'Data usage' (0.00GB of data used), and 'Advanced Calling'. The 'Wi-Fi' option is highlighted with an orange rectangular box.</p>

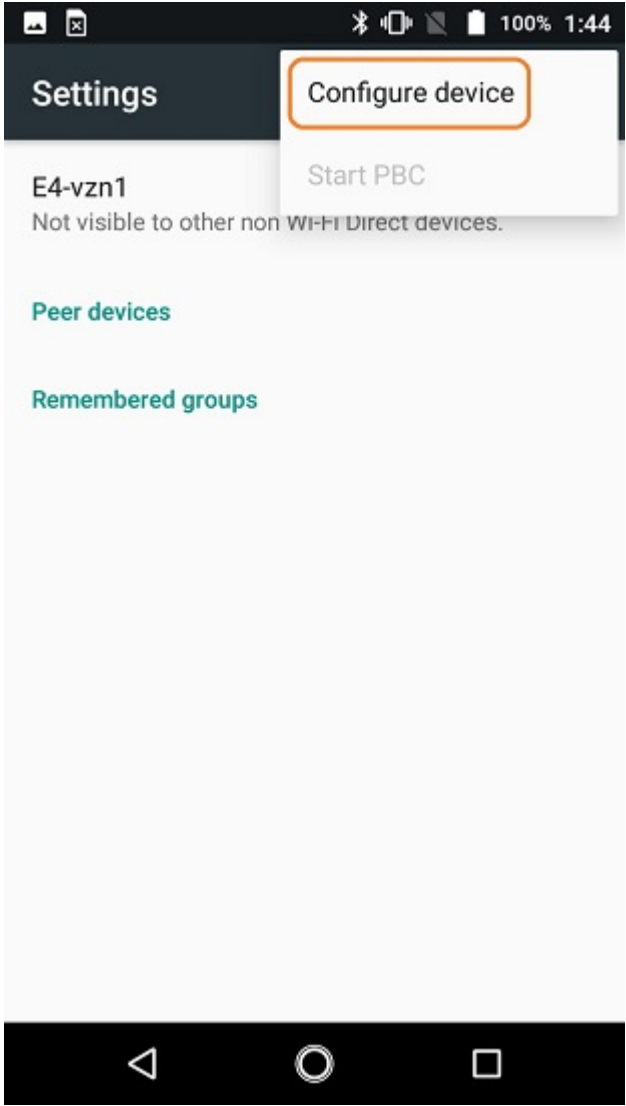
Step	Image
<p>3. Touch the three vertical dots to display a pop-up menu.</p>	

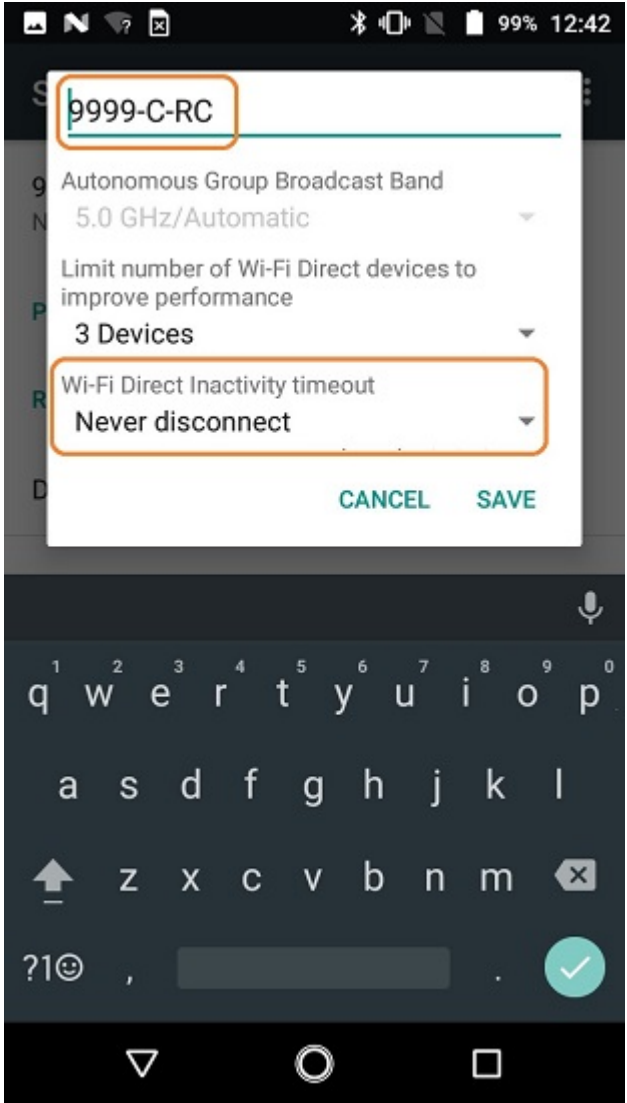
Step	Image
4. Select <b>Advanced</b> from the pop-up menu.	

Step	Image
<p>5. Select <b>Wi-Fi Direct</b> from the <b>Advanced Wi-Fi</b> screen.</p>	 <p>The image shows the 'Advanced Wi-Fi' settings screen on an Android device. At the top, there is a back arrow and the title 'Advanced Wi-Fi'. Below the title, there are several menu items: 'Install certificates', 'Wi-Fi Direct' (which is highlighted with an orange rectangular box), 'WPS Push Button', and 'WPS Pin Entry'. Underneath these are three toggle switches: 'Avoid bad Wi-Fi connections' (turned off), 'Wi-Fi notifications' (turned on), and 'Show Wi-Fi pop-up' (turned off). The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.</p>



Step	Image
<p>6. Touch the three vertical dots to display a pop-up menu.</p>	 <p>The image shows a screenshot of an Android phone's Settings application. The title bar at the top is dark and contains the word "Settings" on the left, "SEARCHING..." in the center, and three vertical dots on the right. The three vertical dots are enclosed in a red rectangular box. Below the title bar, the screen displays "MotoG" and the text "Not visible to other non Wi-Fi Direct devices." Underneath, there are two sections: "Peer devices" and "Remembered groups", both in teal text. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.</p>

Step	Image
<p>7. Select <b>Configure Device</b> from the pop-up menu.</p>	 <p>The image shows a screenshot of an Android phone's Settings application. The top status bar displays icons for notifications, Bluetooth, Wi-Fi, and battery, along with the time 1:44 and 100% battery. The main screen shows the 'Settings' app interface with a dark header. A pop-up menu is visible over the 'E4-vzn1' section, with the 'Configure device' option highlighted by an orange border. Below the pop-up, the text 'Not visible to other non-Wi-Fi Direct devices.' is visible. Further down, there are sections for 'Peer devices' and 'Remembered groups'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.</p>

Step	Image
<p>8. Use touch pad to enter new name of device. If the device will be a Robot Controller, specify your team number and "-RC". If the device will be a Driver Station, specify your team number and "-DS".</p> <p>You can also set the Wi-Fi Direct inactivity timeout to "Never disconnect" and then hit the <b>SAVE</b> button to save your changes.</p> <p>Note that in the screenshot shown to the right, the team number is "9999". The "-C" indicates that this is from the third pair of smartphones for this team. The "-RC" indicates that this phone will be a Robot Controller.</p>	
<p>9. After renaming phone, power cycle the device.</p>	

### 7.3. Installing the FTC Apps

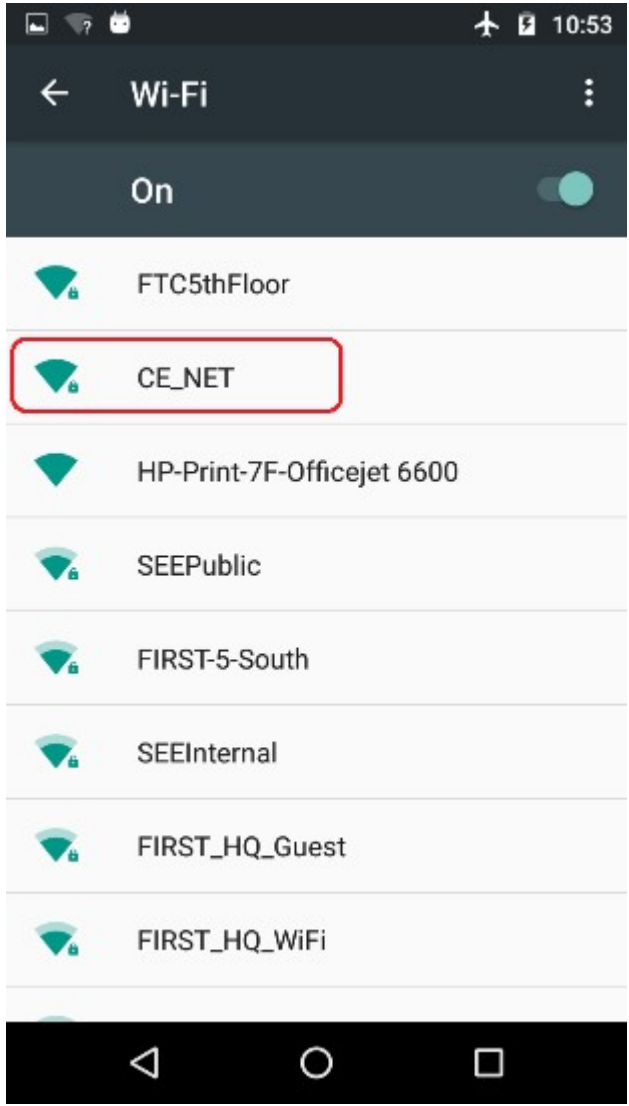
The FTC apps are available to download for free from the Google Play store. You will need to have your Android phones connected to a Wi-Fi network that has Internet access before you can access the Google Play store. You will also need a Google account to be able to download the apps from the Google Play store.

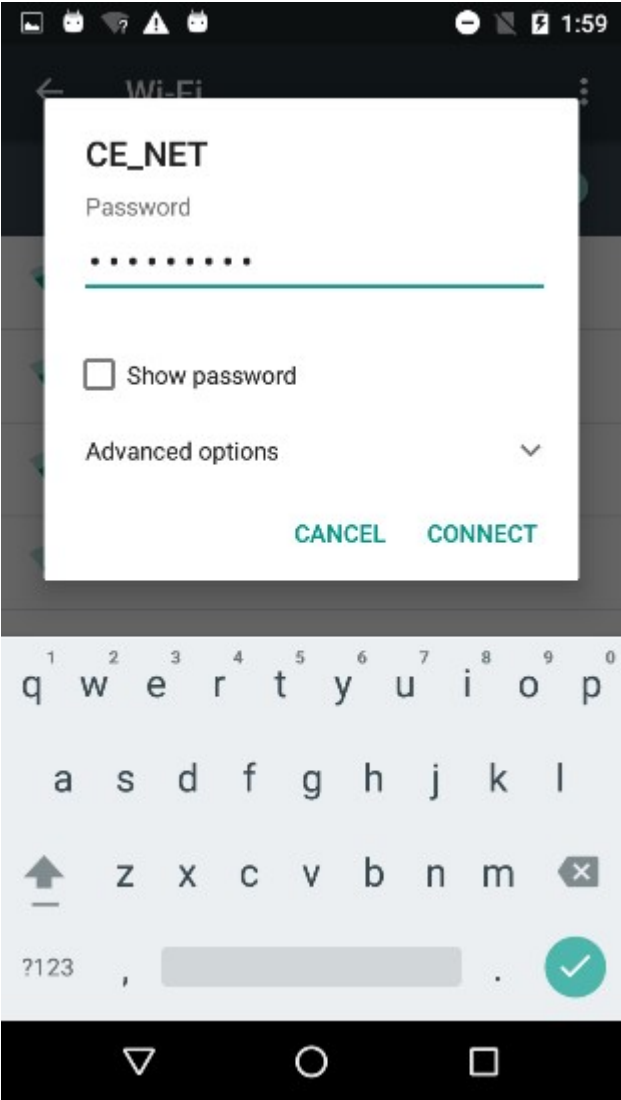
It is also possible to "side-load" the FTC Android Apps onto the Robot Controller and Driver Station phones. The GitHub repository contains the release versions of the Android apps:

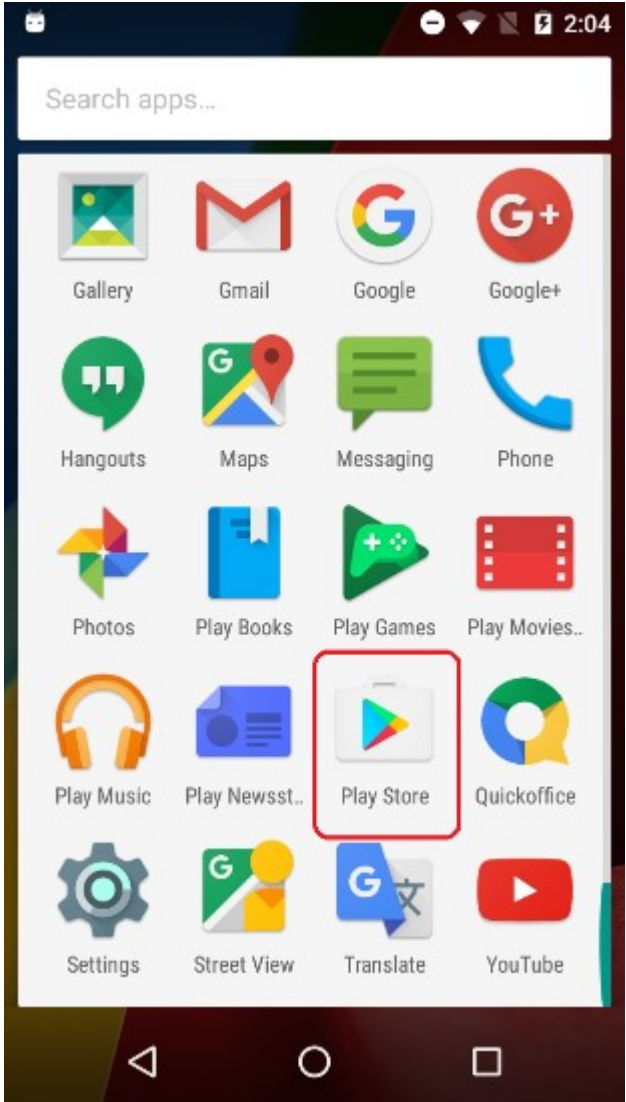
[https://github.com/ftctechnh/ftc\\_app/tree/master/doc/apk](https://github.com/ftctechnh/ftc_app/tree/master/doc/apk)

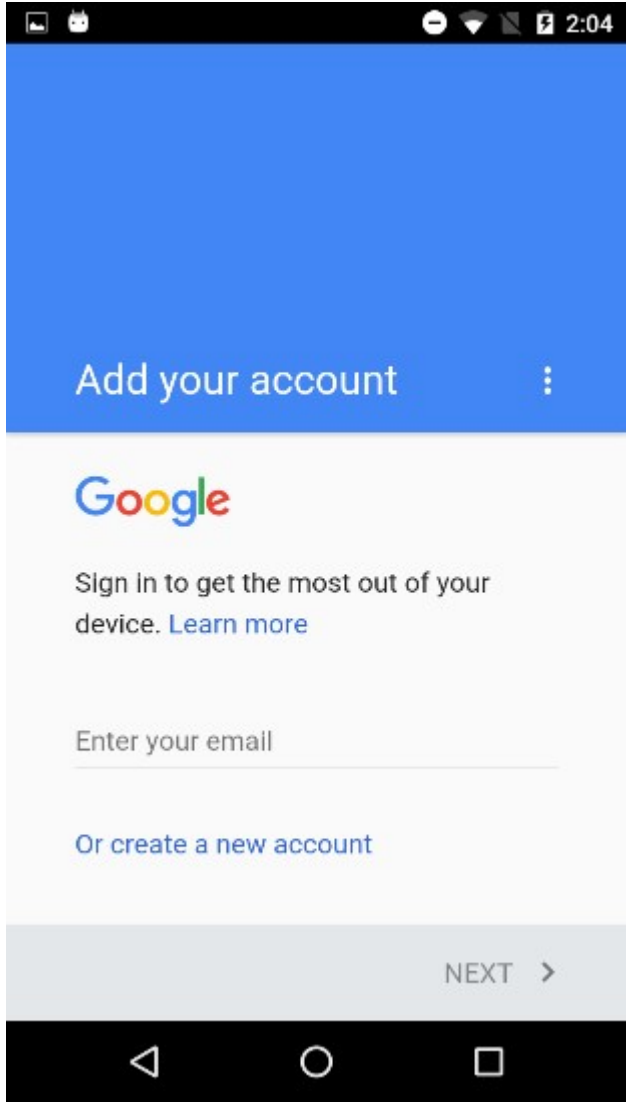
However, this section of the wiki does **not** include instructions on how to side-load the apps. It only explains how to use Google Play to install the FTC apps.

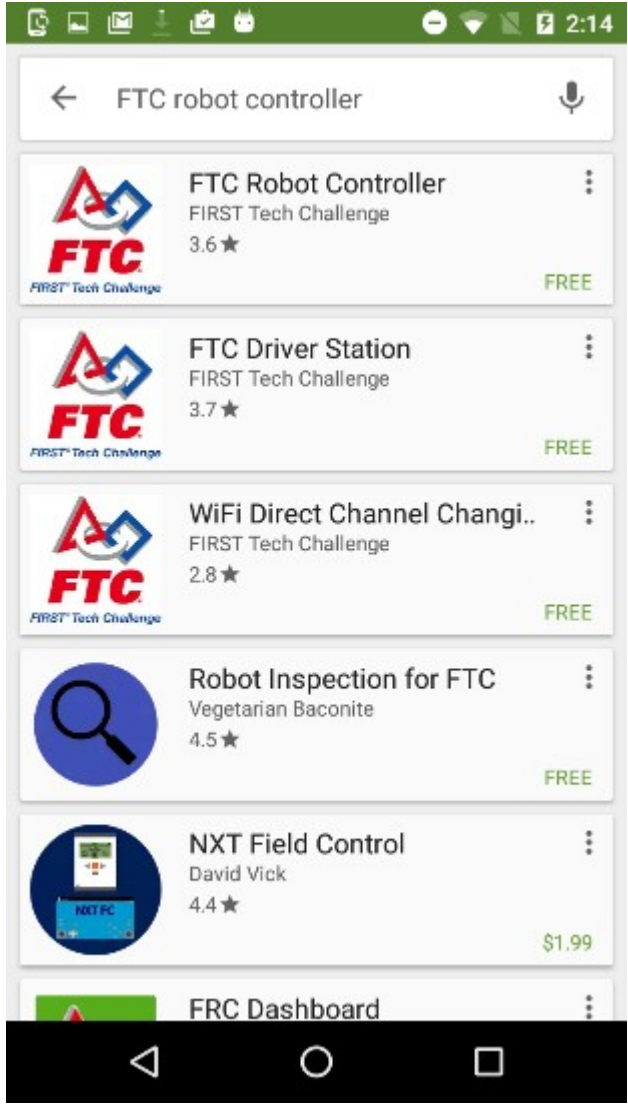
**NOTE:** it will take an estimated 7.5 minutes per phone to complete this task.

Step	Image
<p>1. From the Android Wi-Fi screen look for the name of your wireless network (“CE_NET” in this example) and touch the wireless network name to log into the network.</p>	


Step	Image
<p>2. Specify the password using the touch keypad and hit <b>CONNECT</b> to connect to this wireless network.</p>	

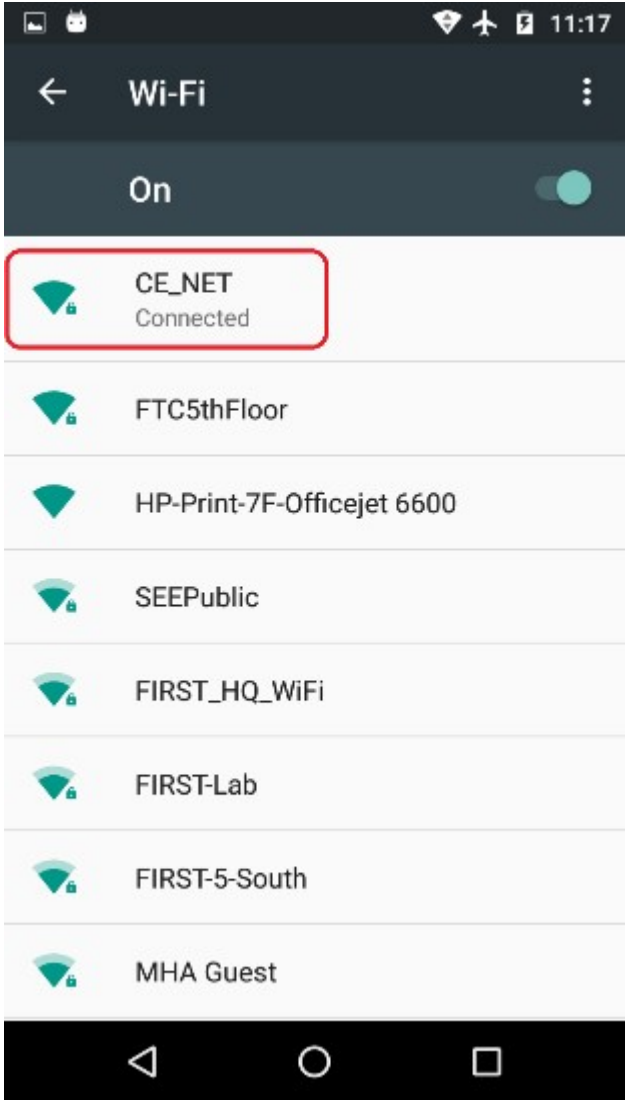
Step	Image
<p>3. Find the Google Play Store icon on your phone and click it to launch the Google Play Store app.</p>	 <p>The image shows a screenshot of an Android phone's app drawer. At the top, there is a search bar labeled "Search apps...". Below it, a grid of app icons is displayed. The icons include Gallery, Gmail, Google, Google+, Hangouts, Maps, Messaging, Phone, Photos, Play Books, Play Games, Play Movies.., Play Music, Play Newsst.., Play Store, Quickoffice, Settings, Street View, Translate, and YouTube. The Play Store icon, which features a colorful play button symbol, is highlighted with a red rectangular box.</p>

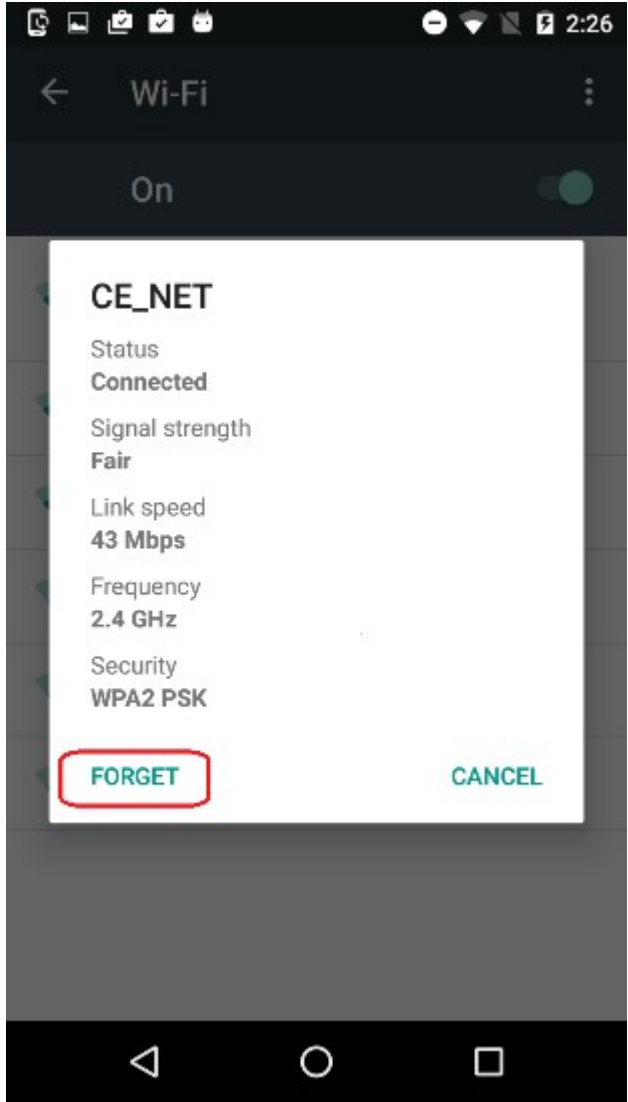
Step	Image
<p>4. If you haven't signed into your Google account yet, follow the onscreen instructions to log into your Google account.</p> <p>If you don't have a Google account, follow the onscreen instructions to create a new account.</p>	

Step	Image
<p>5. In the search window of the Google Play app, type in the words “FTC Robot Controller” to find the Robot Controller or “FTC Driver Station” to find the Driver Station.</p>	 <p>The screenshot shows the Google Play Store search results for the query "FTC robot controller". The search bar at the top contains the text "FTC robot controller" and a microphone icon. Below the search bar, there is a list of search results. The first three results are from the "FIRST Tech Challenge" developer and all are free. The first result is "FTC Robot Controller" with a 3.6 star rating. The second result is "FTC Driver Station" with a 3.7 star rating. The third result is "WiFi Direct Channel Changi.." with a 2.8 star rating. The fourth result is "Robot Inspection for FTC" by Vegetarian Baconite with a 4.5 star rating and is free. The fifth result is "NXT Field Control" by David Vick with a 4.4 star rating and a price of \$1.99. The sixth result is "FRC Dashboard". The bottom of the screen shows the Android navigation bar.</p>



Step	Image
<p>6. Tap on the app in the Google Play listing to bring up the installation screen. Follow the onscreen instructions to install the appropriate app for your phone.</p> <p><b>Important note:</b> When you install the FTC apps, <i>only install one FTC app</i> (FTC Robot Controller or FTC Driver Station) <i>per phone</i>.</p> <p>You should avoid installing both apps onto the same phone. Doing so can cause Wi-Fi connection problems. You should only install the FTC Robot Controller app onto the phone that will be the Robot Controller and the FTC Driver Station app onto the phone that will be the Driver Station.</p>	

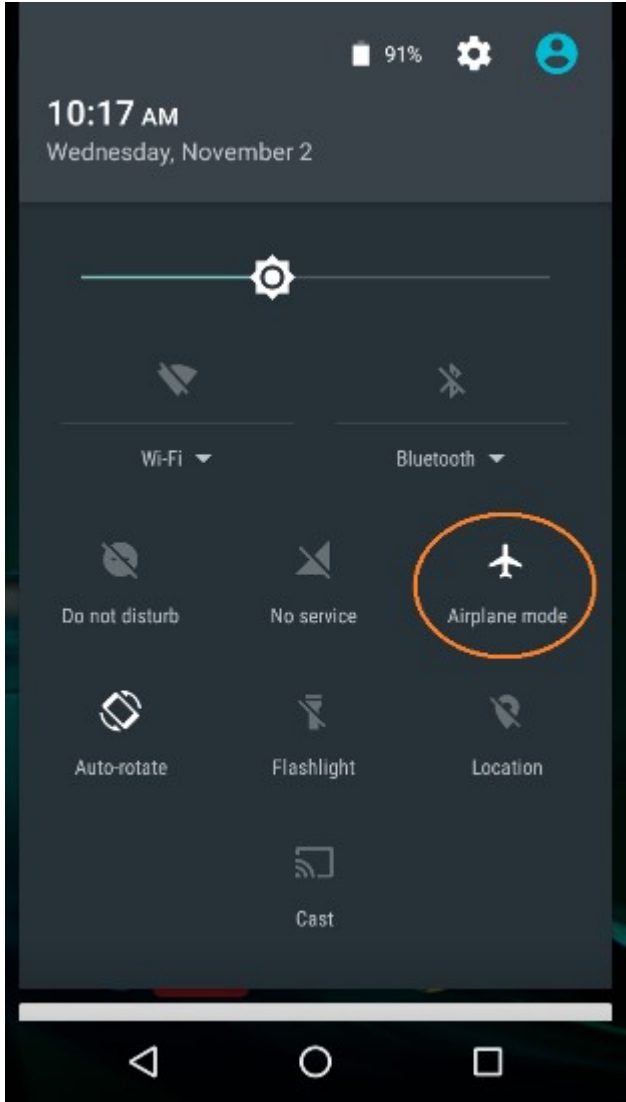
Step	Image
<p>7. After you have successfully installed the app, you should forget the external wireless network on your phone.</p> <p>Go to the Android Wi-Fi screen, find the name of the currently connected network, and tap on the network name to bring up a pop-up box with info about the network.</p>	 <p>The image shows the Android Wi-Fi settings interface. At the top, the Wi-Fi toggle is turned on. Below it, a list of available networks is shown. The network 'CE_NET' is highlighted with a red box and labeled as 'Connected'. Other networks listed include 'FTC5thFloor', 'HP-Print-7F-Officejet 6600', 'SEEPublic', 'FIRST_HQ_WiFi', 'FIRST-Lab', 'FIRST-5-South', and 'MHA Guest'. The time 11:17 is visible in the top right corner.</p>

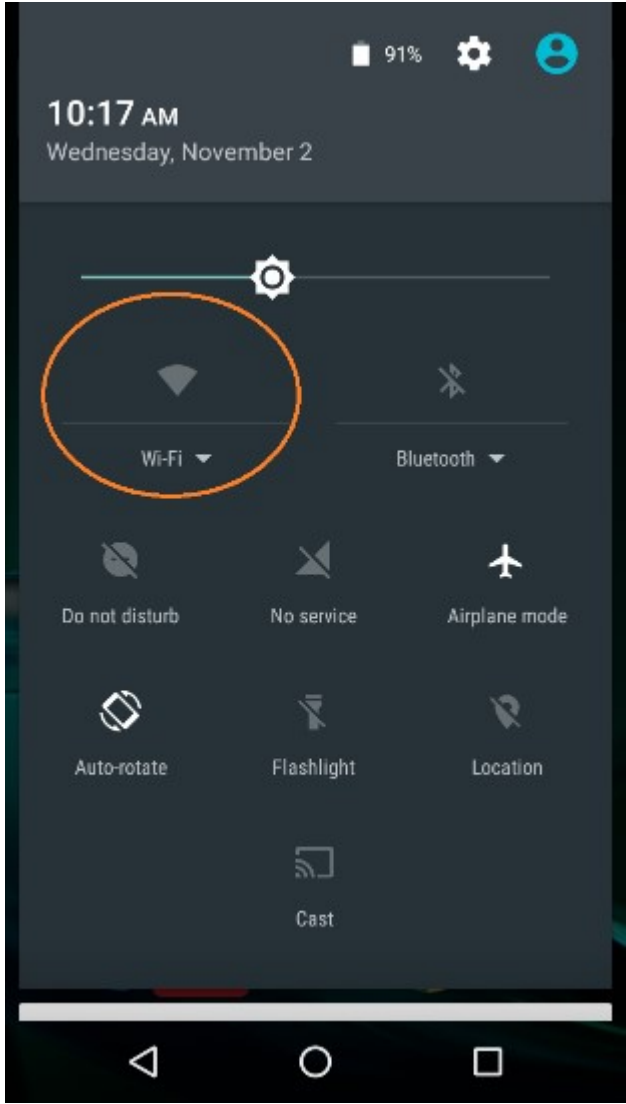
Step	Image
8. Click on the <b>FORGET</b> button to forget the wireless network.	

#### 7.4. Placing Phones into Airplane Mode with Wi-Fi On

For the FIRST Tech Challenge competitions, it is important that you place your Robot Controller and Driver Station phones into Airplane mode but keep their Wi-Fi radios turned on. This is important because you do not want any of the cellular telephone functions to be enabled during a match. The cellular telephone functions could disrupt the function of the robot during a match.

**NOTE:** it will take an estimated 2.5 minutes per phone to complete this task. Also note that the screens displayed on your Android devices might differ slightly from the images contained in this wiki.

Step	Image
<p>1. On the main Android screen of each smartphone, use your finger to slide from the top of the screen down towards the bottom of the screen to display the quick configuration screen.</p> <p>Note that for some smartphones you might have to swipe down more than once to display the quick configuration screen, particularly if there are messages or notifications displayed at the top of your screen.</p> <p>Look for the Airplane mode icon (which is shaped like an airplane) and if the icon is not activated, touch the icon to put the phone into airplane mode.</p>	

Step	Image
<p>2. Placing the phone into airplane mode will turn off the Wi-Fi radio. If the Wi-Fi icon has a diagonal line through it (see Step 1 above), then the Wi-Fi radio is disabled. You will need to touch the <b>Wi-Fi</b> icon on the quick configuration screen to turn the Wi-Fi radio back on.</p>	

## 7.5. Pairing the Driver Station to the Robot Controller



### 7.5.1. Control Hub Users

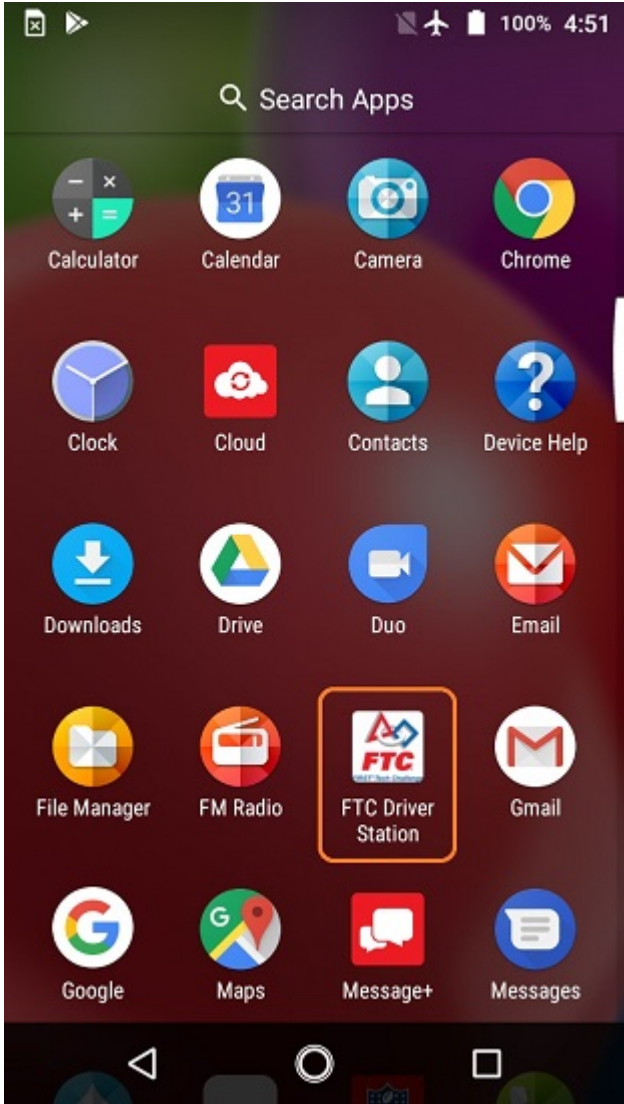
The REV Robotics Control Hub should come with the Robot Controller app pre-installed. Once you have successfully installed the FTC Driver Station on an Android phone, you will want to establish a secure wireless connection between the Control Hub and the Driver Station. This connection will allow your Driver Station phone to select op modes on your Robot Controller and send gamepad input to these programs. Likewise, it will allow your op modes running on your Robot Controller to send telemetry data to your Driver Station phone where it can be displayed for your drivers. The process to connect the two devices is known as “pairing.”

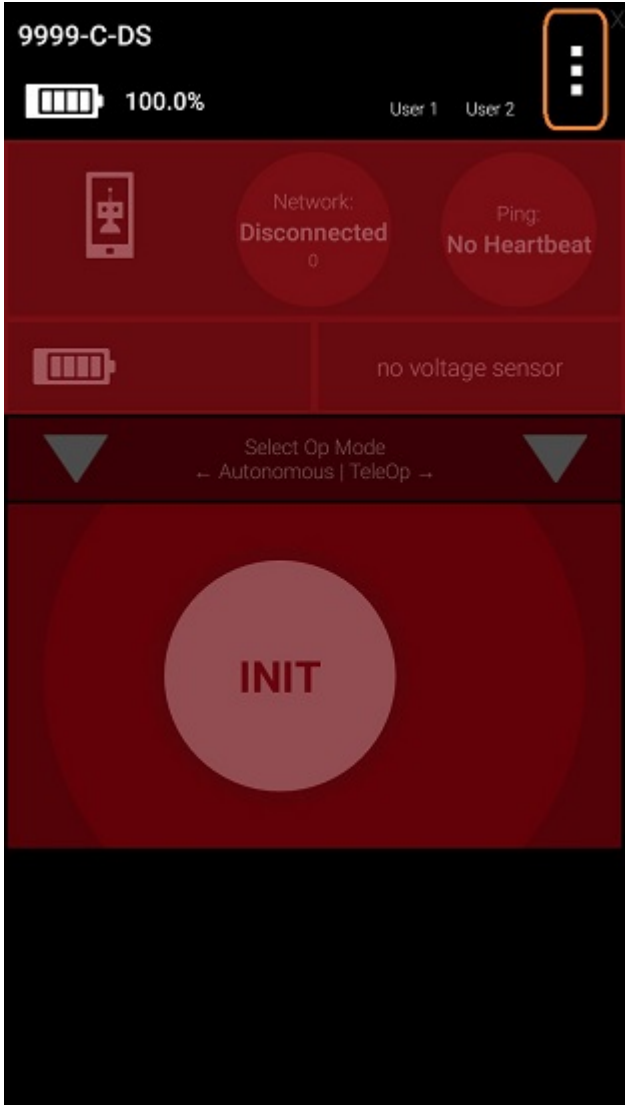
**NOTE:** the Control Hub does not have its own internal battery. Before you can connect a Driver Station to the Control Hub, you must connect the Control Hub to a 12V battery.

*Gracious Professionalism®* - “Doing your best work while treating others with respect and kindness - It’s what makes FIRST, first.”

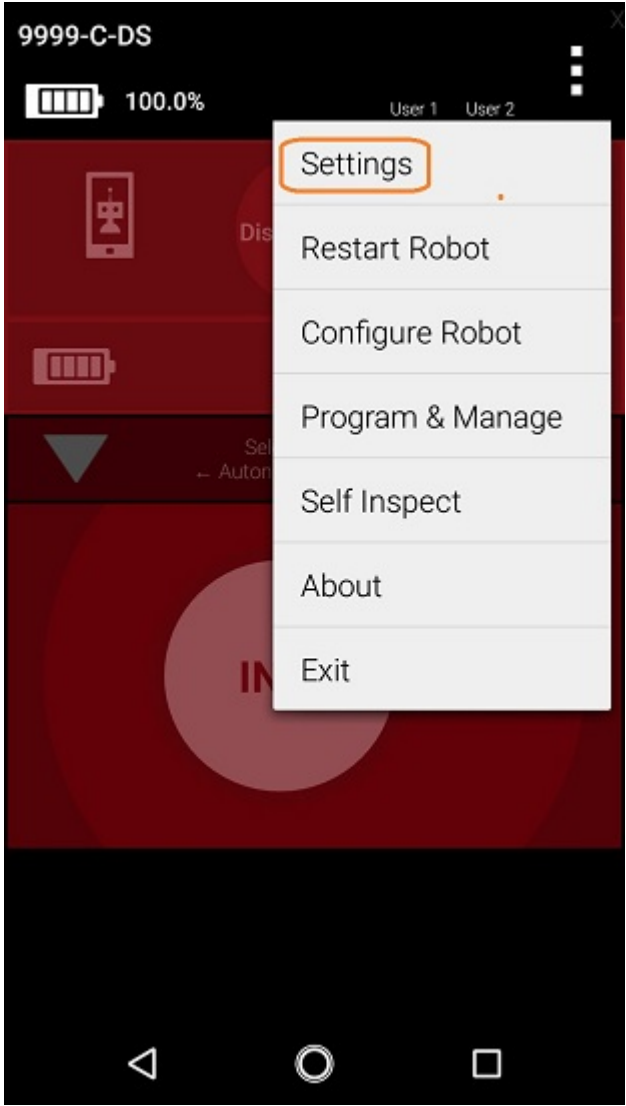
Also note that it will take an estimated 10 minutes to complete this task.

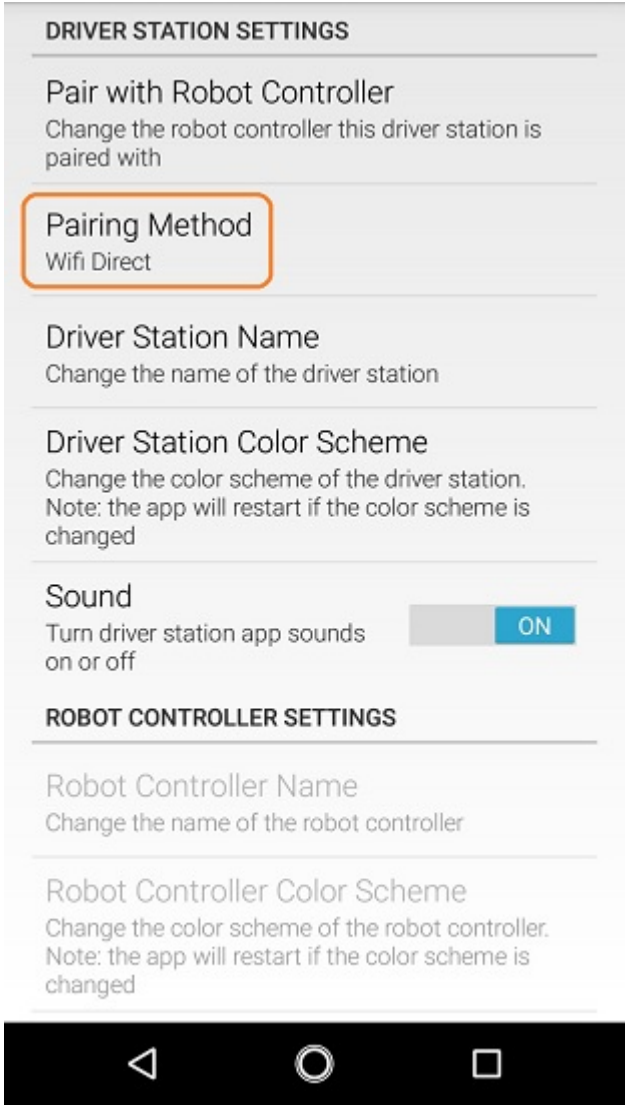
Step	Image
<p>1. Connect an approved 12V battery to the power switch (REV-31-1387) and make sure the switch is in the off position. Connect the switch to an XT30 port on the Control Hub and turn the switch on.</p> <p>The LED should initially be blue on the Control Hub.</p>	
<p>2. It takes approximately 18 seconds for the Control Hub to power on. The Control Hub is ready to pair with the Driver Station when the LED turns green.</p> <p>Note: the light blinks blue every ~5 seconds to indicate that the Control Hub is healthy.</p>	

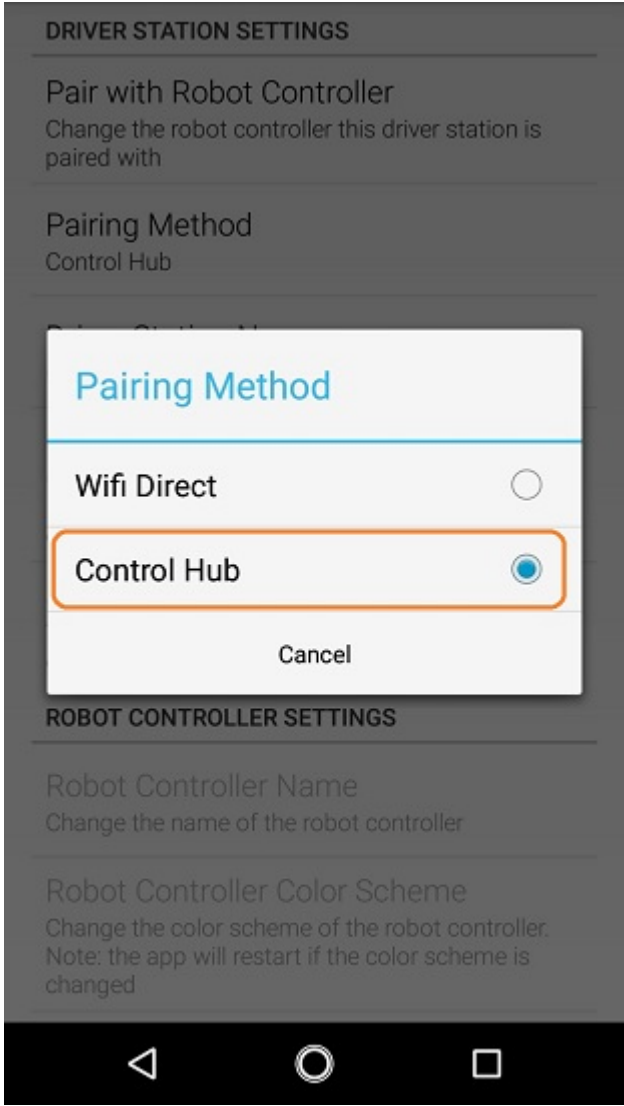
Step	Image
<p>3. On the Driver Station device, browse the available apps and locate the <b>FTC Driver Station</b> icon. Tap on the icon to launch the Driver Station app.</p> <p>Note that the first time you launch the app your Android device might prompt you for permissions that the app will need to run properly. Whenever prompted, press <b>Allow</b> to grant the requested permission.</p>	 <p>The image shows a screenshot of an Android phone's app drawer. The screen displays a grid of application icons. At the top, there is a search bar labeled "Search Apps". Below the search bar, the icons are arranged in a grid. The "FTC Driver Station" icon, which features a red and white logo with the letters "FTC" and "Driver Station" below it, is highlighted with a red rectangular box. Other visible icons include Calculator, Calendar, Camera, Chrome, Clock, Cloud, Contacts, Device Help, Downloads, Drive, Duo, Email, File Manager, FM Radio, Gmail, Google, Maps, Message+, and Messages. The status bar at the top shows the time as 4:51 and 100% battery.</p>

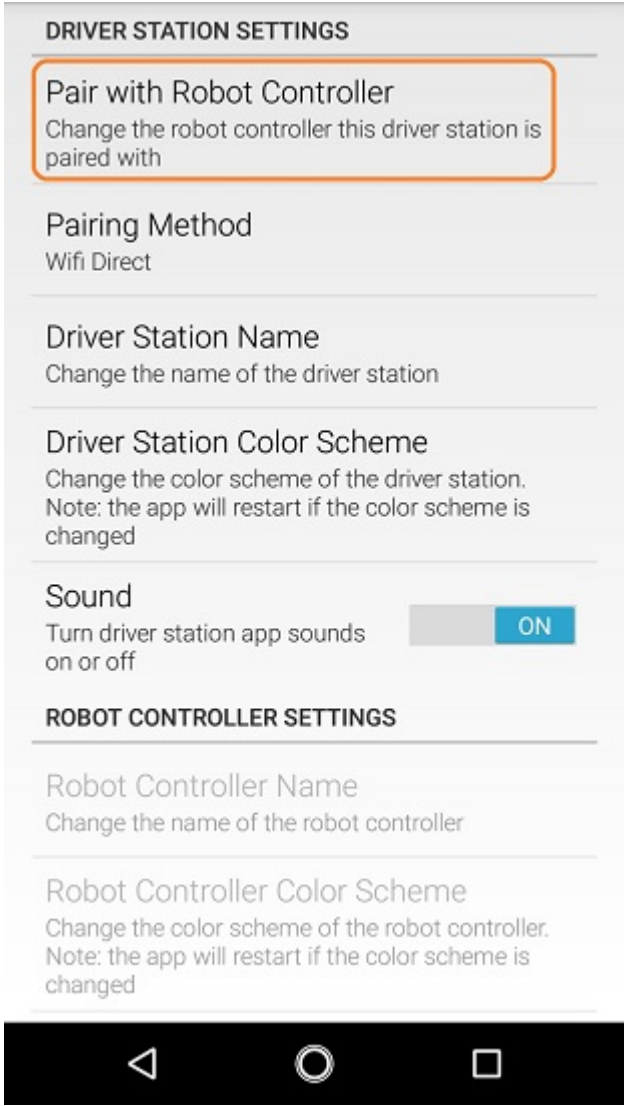
Step	Image
<p>4. Touch the three vertical dots on the upper right-hand corner of the main screen of the FTC Driver Station app. This will launch a pop-up menu.</p>	

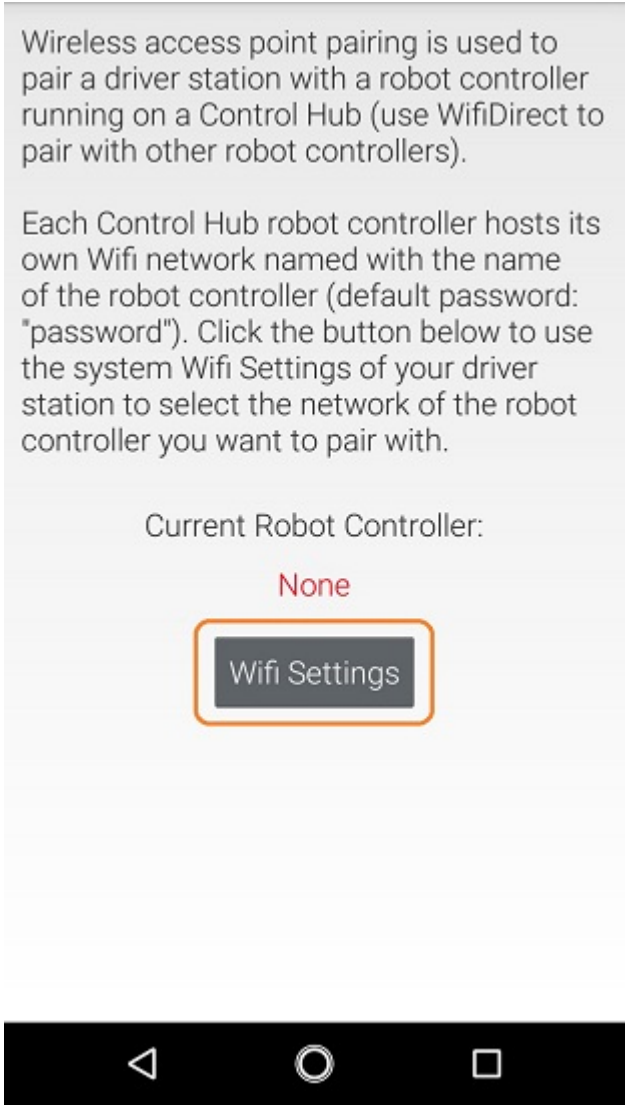


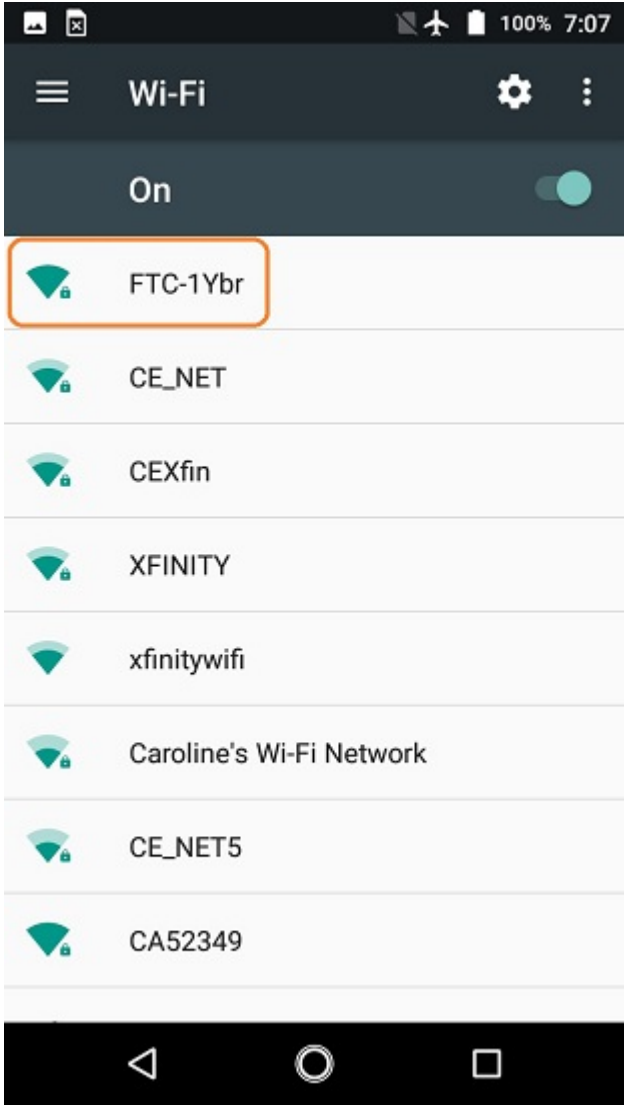
Step	Image
<p>5. Select <b>Settings</b> from the pop-up menu.</p>	 <p>The image shows a screenshot of the FIRST Tech Challenge Android Studio interface. The top status bar displays '9999-C-DS', a battery icon at 100.0%, and user names 'User 1' and 'User 2'. The main interface has a dark red background with various icons and text. A white pop-up menu is overlaid on the right side, listing several options: 'Settings' (highlighted with an orange border), 'Restart Robot', 'Configure Robot', 'Program &amp; Manage', 'Self Inspect', 'About', and 'Exit'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.</p>

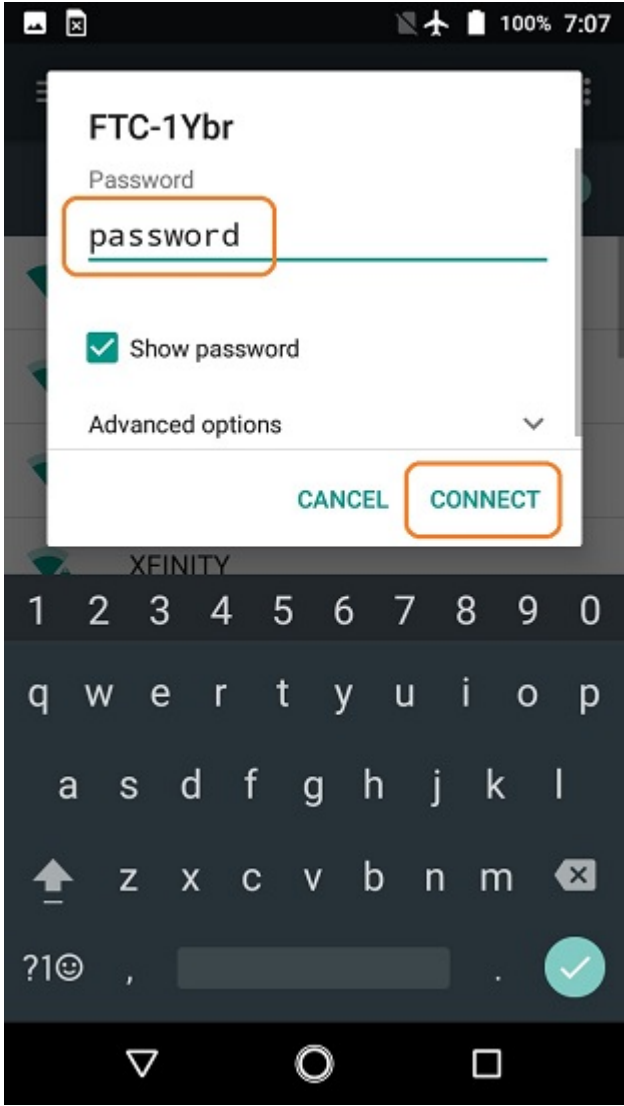
Step	Image
<p>6. From the <b>Settings</b> screen, select <b>Pairing Method</b></p>	 <p><b>DRIVER STATION SETTINGS</b></p> <p><b>Pair with Robot Controller</b> Change the robot controller this driver station is paired with</p> <p><b>Pairing Method</b> Wifi Direct</p> <p><b>Driver Station Name</b> Change the name of the driver station</p> <p><b>Driver Station Color Scheme</b> Change the color scheme of the driver station. Note: the app will restart if the color scheme is changed</p> <p><b>Sound</b> Turn driver station app sounds on or off <input checked="" type="checkbox"/> ON</p> <p><b>ROBOT CONTROLLER SETTINGS</b></p> <p><b>Robot Controller Name</b> Change the name of the robot controller</p> <p><b>Robot Controller Color Scheme</b> Change the color scheme of the robot controller. Note: the app will restart if the color scheme is changed</p>

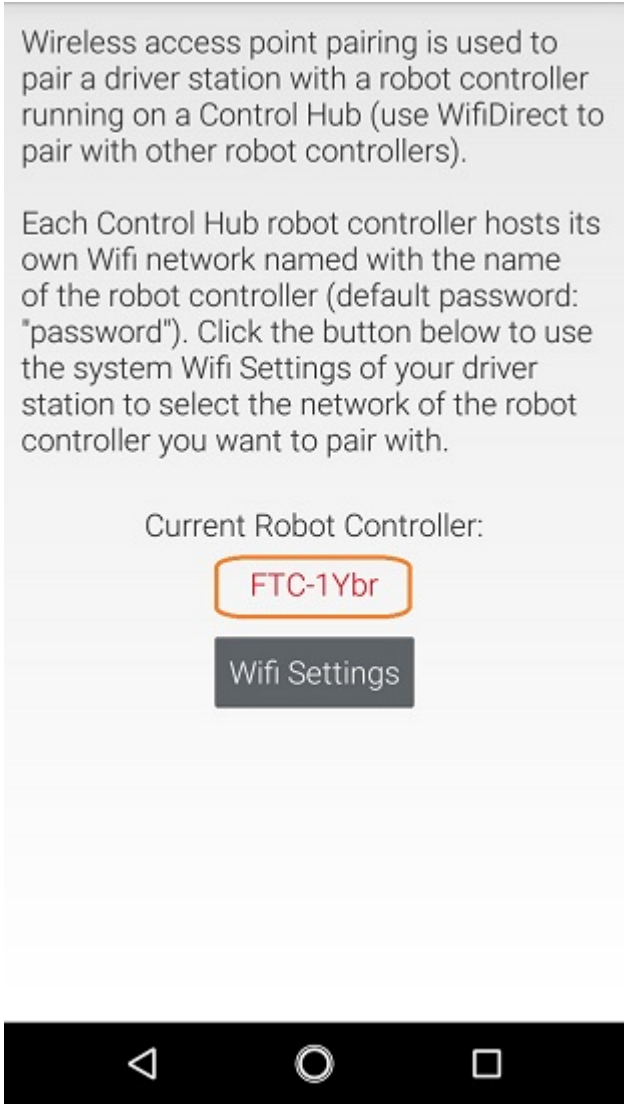
Step	Image
<p>7. Touch the words <b>Control Hub</b> to pair with the Driver Station.</p>	 <p>The image shows a screenshot of the 'DRIVER STATION SETTINGS' app. The main screen is dimmed, showing options like 'Pair with Robot Controller', 'Pairing Method' (set to 'Control Hub'), 'ROBOT CONTROLLER SETTINGS', 'Robot Controller Name', and 'Robot Controller Color Scheme'. A white dialog box titled 'Pairing Method' is overlaid on the screen. It contains two radio button options: 'Wifi Direct' (unselected) and 'Control Hub' (selected). The 'Control Hub' option is highlighted with an orange rectangular border. A 'Cancel' button is located at the bottom of the dialog box. The Android navigation bar is visible at the bottom of the screen.</p>

Step	Image
<p>8. From the <b>Settings</b> screen, select <b>Pair with Robot Controller</b>.</p>	 <p><b>DRIVER STATION SETTINGS</b></p> <p><b>Pair with Robot Controller</b> Change the robot controller this driver station is paired with</p> <p><b>Pairing Method</b> Wifi Direct</p> <p><b>Driver Station Name</b> Change the name of the driver station</p> <p><b>Driver Station Color Scheme</b> Change the color scheme of the driver station. Note: the app will restart if the color scheme is changed</p> <p><b>Sound</b> Turn driver station app sounds on or off <input checked="" type="checkbox"/> ON</p> <p><b>ROBOT CONTROLLER SETTINGS</b></p> <p><b>Robot Controller Name</b> Change the name of the robot controller</p> <p><b>Robot Controller Color Scheme</b> Change the color scheme of the robot controller. Note: the app will restart if the color scheme is changed</p>

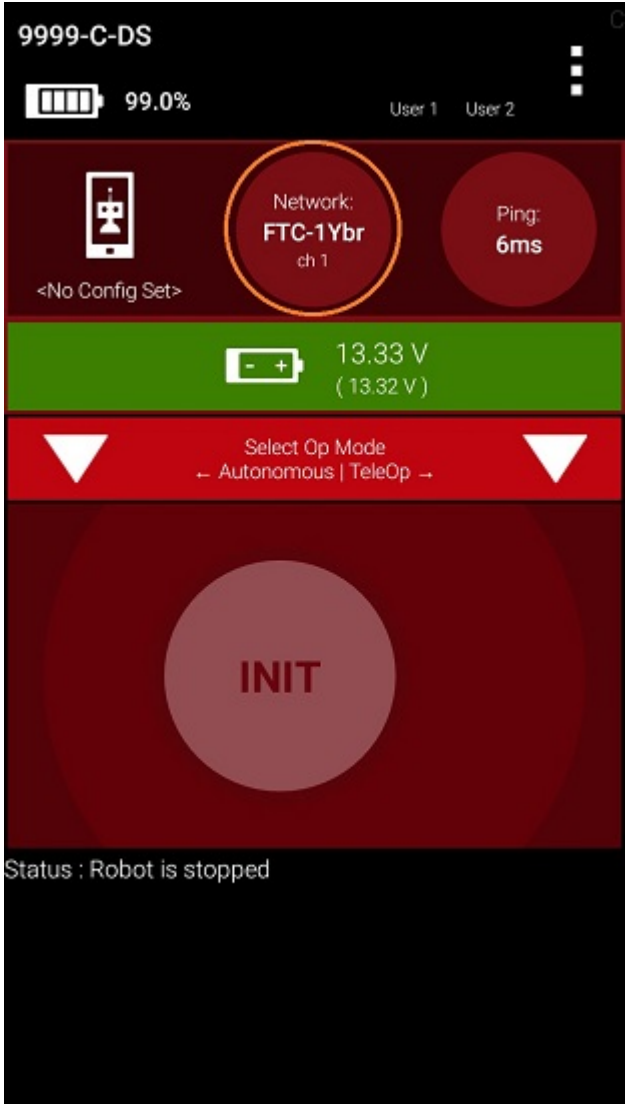
Step	Image
<p>9. From <b>Pair with Robot Controller</b> screen, press the <b>Wifi Settings</b> to launch the device's Android Wifi Settings screen.</p>	<p>Wireless access point pairing is used to pair a driver station with a robot controller running on a Control Hub (use WifiDirect to pair with other robot controllers).</p> <p>Each Control Hub robot controller hosts its own Wifi network named with the name of the robot controller (default password: "password"). Click the button below to use the system Wifi Settings of your driver station to select the network of the robot controller you want to pair with.</p> <p>Current Robot Controller:</p> <p>None</p> <p>Wifi Settings</p> 

Step	Image
<p>10. Select WiFi, then select the correct network.</p> <p>If this is the first time you are connecting to the Control Hub, then the default network name should begin with the prefix "FTC-" ("FTC-1Ybr" in this example).</p> <p>The default network name should be listed on a sticker attached to the bottom side of the Control Hub.</p>	

Step	Image
<p>11. When prompted, specify the password for the Control Hub's WiFi network and press <b>Connect</b> to the Hub.</p> <p><b>Note:</b> Default password for the Control Hub network is "password".</p> <p>Also note that when you connect to the Control Hub's WiFi network successfully, the Driver Station will not have access to the Internet.</p>	

Step	Image
<p>12. After you successfully connected to the Hub, use the back arrow to navigate to the previous screen. You should see the name of the WiFi network listed under "Current Robot Controller:". Use the back-arrow key to return to the Settings screen. Then press the back-arrow key one more time to return to the main Driver Station screen.</p>	<p>Wireless access point pairing is used to pair a driver station with a robot controller running on a Control Hub (use WifiDirect to pair with other robot controllers).</p> <p>Each Control Hub robot controller hosts its own Wifi network named with the name of the robot controller (default password: "password"). Click the button below to use the system Wifi Settings of your driver station to select the network of the robot controller you want to pair with.</p> <p>Current Robot Controller:</p> 



Step	Image
<p>13. Verify that the Driver Station screen has changed and that it now indicates that it is connected to the Control Hub.</p> <p>The name of the Control Hub's WiFi network ("FTC-1Ybr" in this example) should be displayed in the Network field on the Driver Station.</p>	 <p>The screenshot shows the Driver Station app interface. At the top, the title is "9999-C-DS". Below the title, there is a battery icon showing 99.0% and two user profiles labeled "User 1" and "User 2". The main display area is divided into several sections: a dark red section with a robot icon and "&lt;No Config Set&gt;", a green section showing "Network: FTC-1Ybr ch 1" and "Ping: 6ms", a red section showing "13.33 V (13.32 V)", a red section with "Select Op Mode" and "Autonomous   TeleOp", and a large dark red circle with "INIT" in the center. At the bottom, the status bar reads "Status : Robot is stopped".</p>

### 7.5.2. Users with Two Android Smartphones

Important Note: If your Driver Station was previously paired to a Control Hub, and you currently would like to connect to an Android smartphone Robot Controller, then before attempting to pair to the Robot Controller, you should forget the Wi-Fi network for the previous Control Hub (using the Android Wifi Settings screen on the Driver Station) and then power cycle the Driver Station phone. If the previous Control Hub is powered on and if you haven't forgotten this network, then the Driver Station might try and connect to the Control Hub and might be unable to connect to the Robot Controller smartphone.

Once you have successfully installed the FTC apps onto your Android phones, you will want to establish a secure wireless connection between the two devices. This connection will allow your Driver Station phone to select op modes on your Robot Controller phone and send gamepad input to these programs. Likewise, it will

allow your op modes running on your Robot Controller phone to send telemetry data to your Driver Station phone where it can be displayed for your drivers. The process to connect the two phones is known as “pairing.”

**Note:** It will take an estimated 10 minutes to complete this task.

## 8. Connecting Devices to a Control or Expansion Hub

This section explains how to connect a motor, a servo, and some sensors to your REV Robotics Control Hub or REV Robotics Expansion Hub. While the Control Hub differs from the Expansion Hub because of its built in Android device, the layout of the external motor, servo, and sensor ports are identical for the Control Hub and Expansion Hub.

The images in this section use an Expansion Hub to demonstrate how to connect the devices. The process, however, is identical for a Control Hub.

When the instructions in this section use the word "Hub", they are referring to a Control Hub or Expansion Hub.

### 8.1. Connecting 12V Power to the Hub

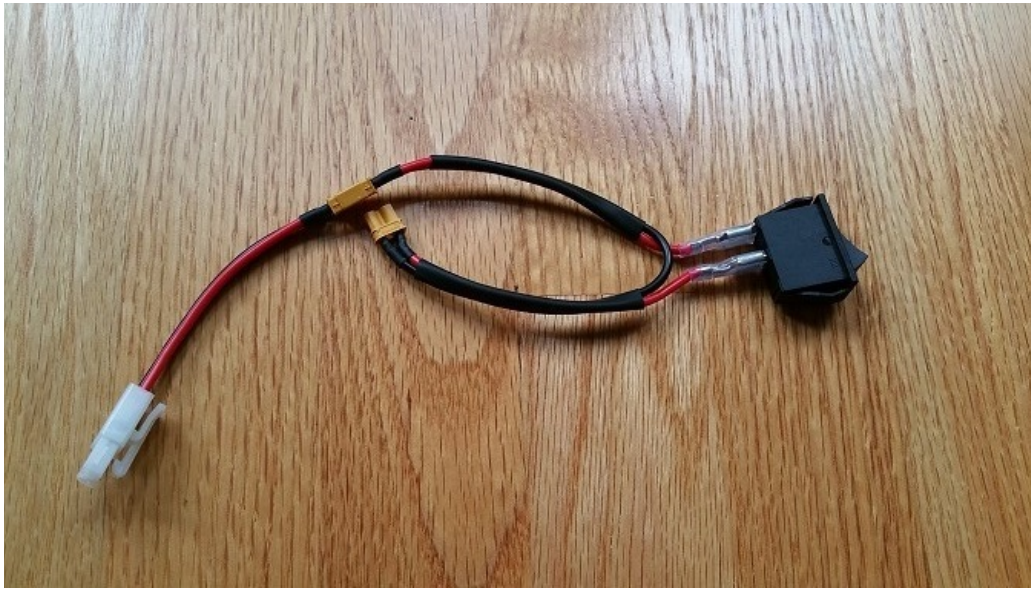
The Hub draws power from a 12V rechargeable battery. For safety reasons, the battery has a 20A fuse built in. A mechanical switch is used to turn on/turn off the power.

**Note:** that it will take an estimated 5 minutes to complete this task.

#### Connecting 12V Power to the Hub

1. If your 12V battery has a Tamiya style connector, connect the Tamiya to XT30 adapter cable to the matching end of the switch cable.

## Connecting 12V Power to the Hub



**Important Note:** Do not connect the 12V battery to the Tamiya adapter yet. We will connect the battery during a later step.

2. Connect the other end of the switch cable to a matching XT30 port on the Hub.



### Connecting 12V Power to the Hub

3. Verify that the switch is in the OFF position.



4. Connect the 12V battery to the Tamiya to XT30 cable.





## Connecting 12V Power to the Hub

5. Turn on the switch and verify that the Hub is drawing power from the battery. Note that the Hub's LED should be illuminated (notice the blue LED in upper right-hand corner of the Hub in the image below).

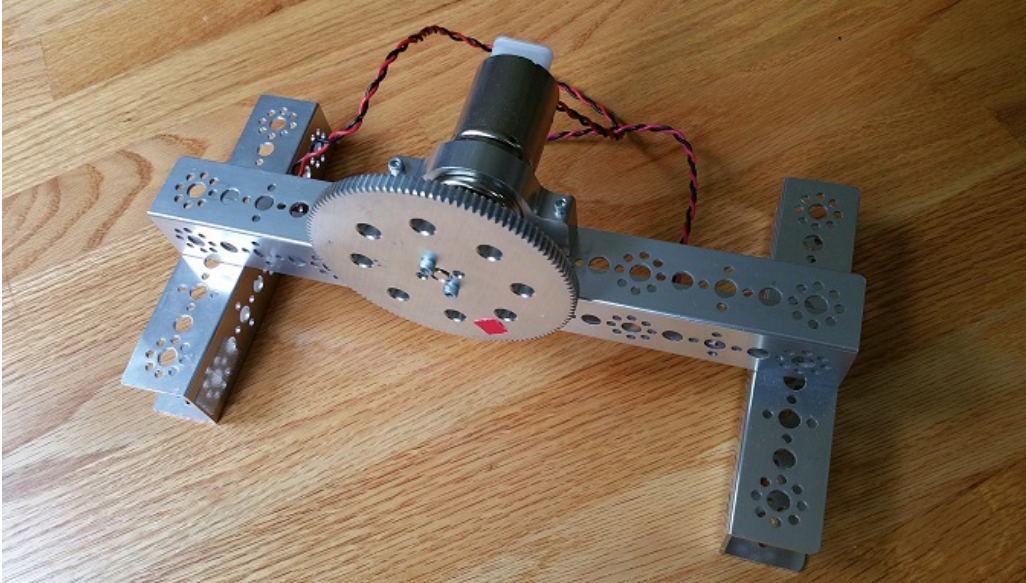


6. Turn off the switch and verify that the Hub is off. Note that the Hub's LED should not be illuminated.



## 8.2. Connecting a Motor to the Hub

The Hub can drive up to four (4) 12V DC motors per Hub. The Hub uses a type of electrical connector known as a 2-pin JST VH connector. Many of the FIRST-approved 12V DC motors are equipped with Anderson Powerpole connectors. An adapter cable can be used to connect the Anderson Powerpole connectors to the Hub motor port (see [FIRST Tech Challenge Robot Wiring Guide](#) for more information).



For the examples in this wiki, FIRST recommends that the user build a simple rig to secure the motor in place and prevent it from moving about during the test runs. The image above shows a Tetrax motor installed in a rig built with a Tetrax motor mount and some Tetrax C-channels. A gear was mounted on the motor shaft to make it easier for the user to see the rotation of the shaft.

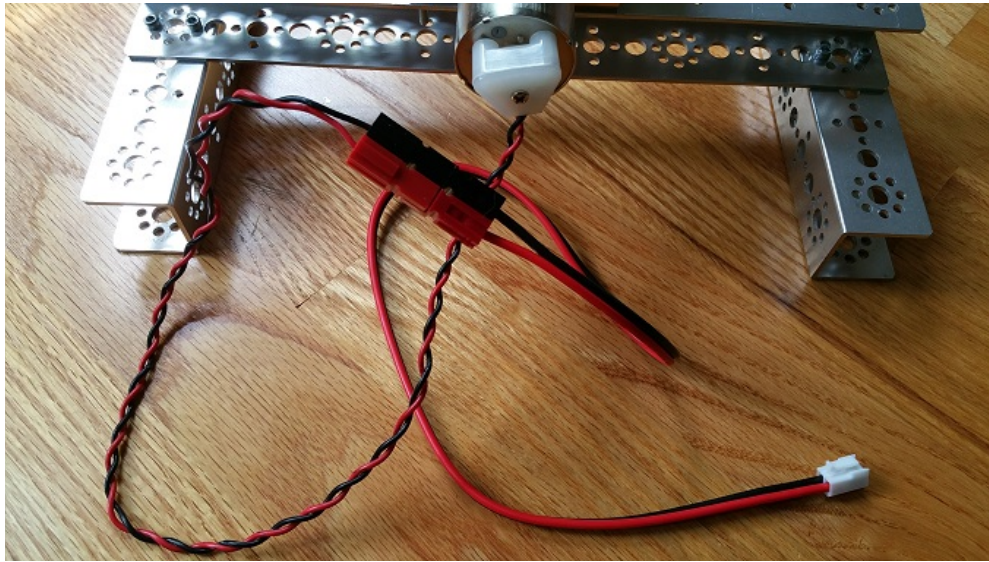
Note that it will take an estimated 2.5 minutes to complete this task.

### Connecting a 12V Motor to the Hub

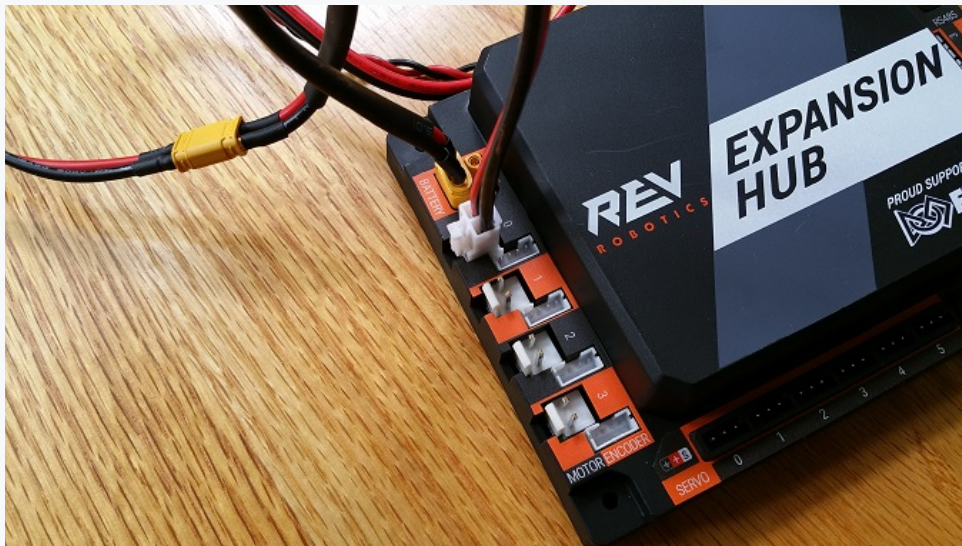
1. Connect the Anderson Powerpole end of the motor's power cable to the Powerpole end of the Anderson to JST VH adapter cable.



### Connecting a 12V Motor to the Hub



2. Connect the other end of the Anderson to JST VH adapter cable into the motor port labeled “0” on the Hub.



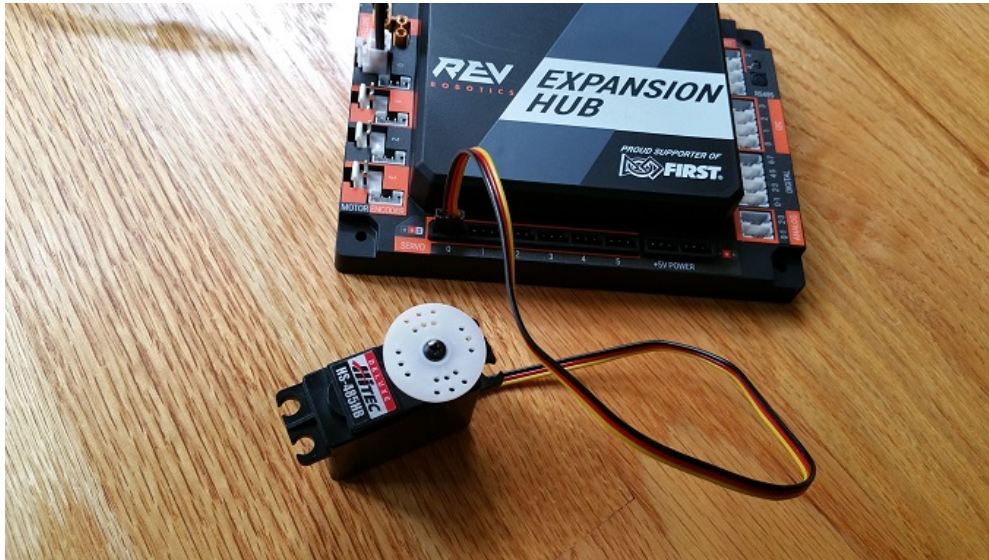
### 8.3. Connecting a Servo to the Hub

The Hub has 6 built-in servo ports. The servo ports accept the standard 3-wire header style connectors commonly found on servos. Note that ground pin is on the left side of the servo port.

Note that it will take an estimated 2.5 minutes to complete this task.

## Connecting a Servo to the Hub

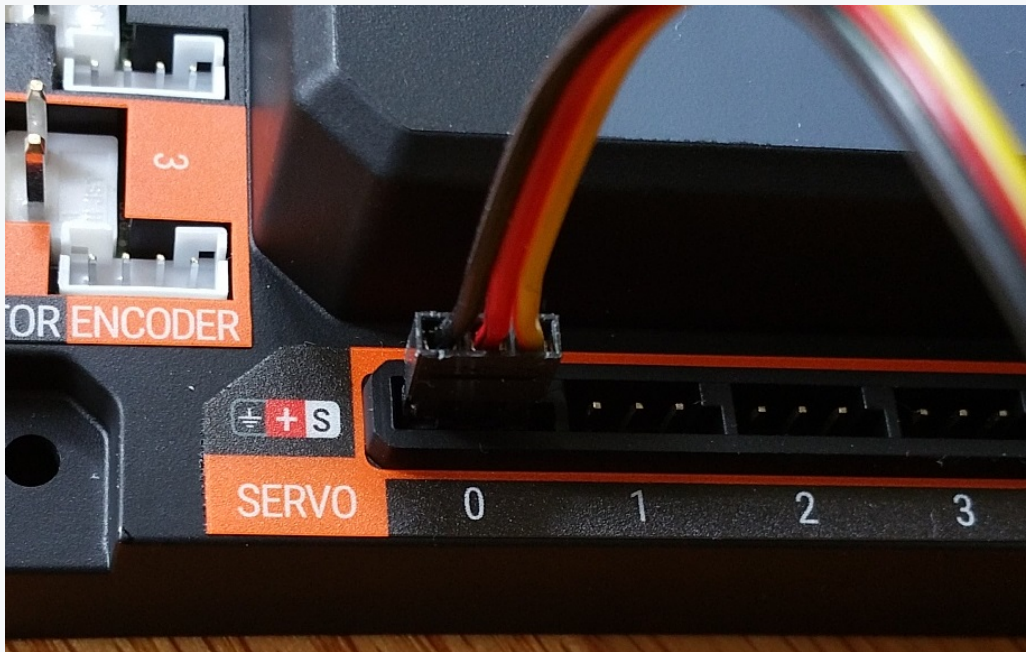
1. Connect the servo cable to the servo port labeled “0” on the Hub. Note that the ground pin is on the left side of the servo port.



2. Verify that the black ground wire of the servo cable matches the ground pin of the servo port (which is aligned on the left side of the port).



### Connecting a Servo to the Hub



#### 8.4. Connecting a Color-Distance Sensor to the Hub

The Hub has 4 independent I2C buses. Each bus has its own port on the Hub. We will connect a REV Robotics Color-Distance sensor to the I2C bus #0 on the Hub.

**Note:** that it will take an estimated 2.5 minutes to complete this task.

### Connecting a Color-Distance Sensor to the Hub

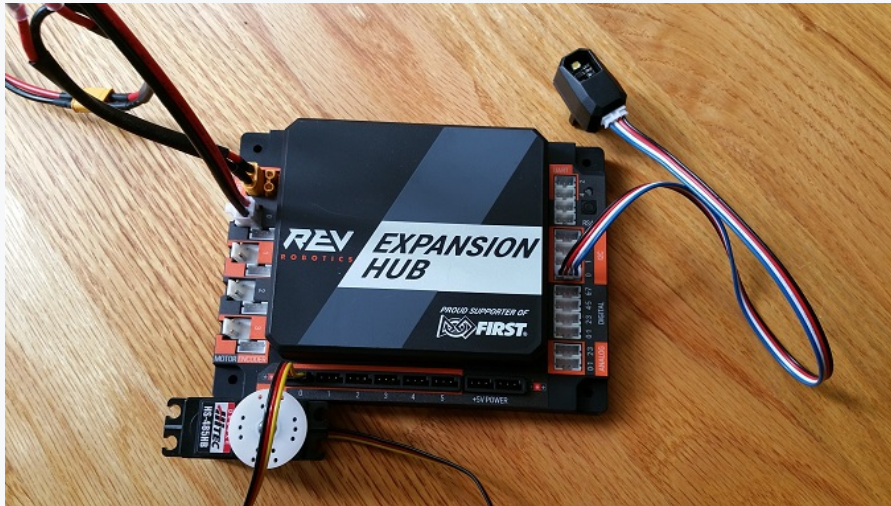
1. Connect the one end of the 4-pin JST PH cable to the REV Robotics Color-Distance sensor.

## Connecting a Color-Distance Sensor to the Hub



2. Plug the other end of the 4-pin JST PH cable to the I2C port labeled “0” on the Hub.

### Connecting a Color-Distance Sensor to the Hub



#### 8.5. Connecting a Touch Sensor to the Hub

The Hub has 4 independent digital input/output (I/O) ports. Each port has two digital I/O pins for a total of 8 digital I/O pins on a Hub. You will connect a REV Robotics Touch sensor to one of the digital I/O ports.

Note that in the case of the REV Robotics Touch Sensor, the device has a connector port for a 4-pin sensor cable. However, the device only needs to connect to one of the two available digital I/O pins. For the REV Robotics Touch Sensor, the second digital I/O pin in the port is the one that gets connected when a standard REV Robotics 4-pin JST PH cable is used. For the “0-1” port, it is the pin labeled “1” that gets connected through the 4-pin cable. Similarly, for the “2-3” port, it is the pin labeled “3” that gets connected through the 4-pin cable.

Note that it will take an estimated 2.5 minutes to complete this task.

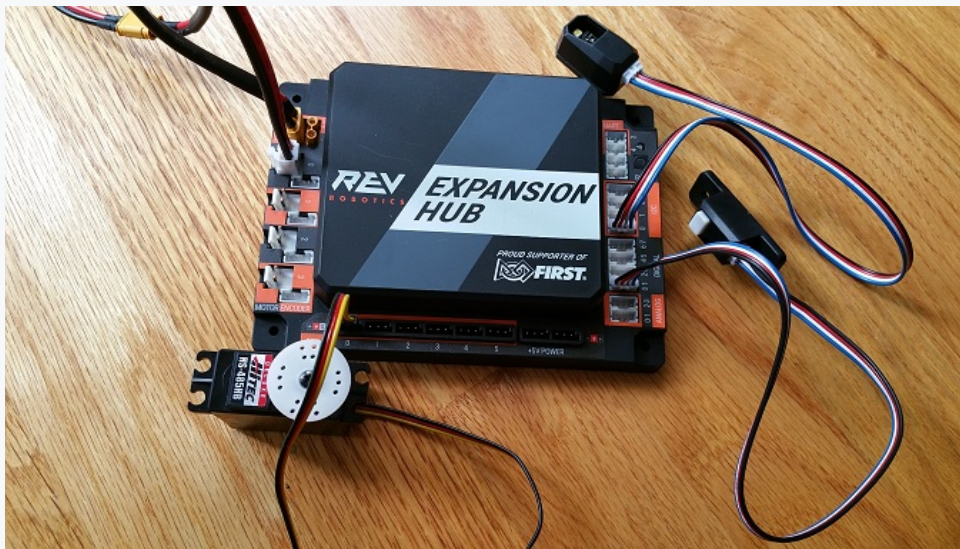
### Connecting a Touch Sensor to the Hub

1. Connect the one end of the 4-pin JST PH cable to the REV Robotics Touch sensor.

## Connecting a Touch Sensor to the Hub



2. Plug the other end of the 4-pin JST PH cable to digital I/O port labeled “0-1” on the Hub.





## 9. Configuring Your Hardware

### 9.1. Before You Begin...

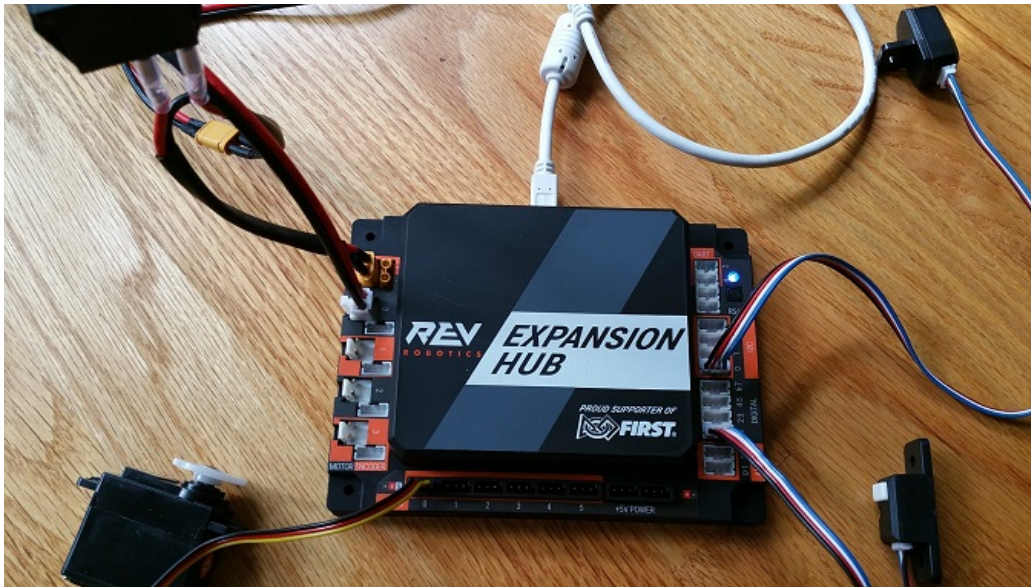
Before you can communicate with the motor, servo and sensors that are connected to the Control Hub or Expansion Hub, you first must create a configuration file on your Robot Controller, so that the Robot Controller will know what hardware is available on the Control Hub's or Expansion Hub's external ports.

### 9.2. Connecting an Android Smartphone to an Expansion Hub

If you are using an Android smartphone as a Robot Controller, you must physically connect the Robot Controller smartphone to the Expansion Hub using a USB cable and an On-The-Go (OTG) adapter. Also, you should verify that the Driver Station is currently paired to the Robot Controller.

#### Connecting an Android Smartphone to an Expansion Hub

1. Power on the Expansion Hub by turning on the power switch.



2. Plug the Type B Mini end of the USB cable into the USB mini port on the Expansion Hub.

### Connecting an Android Smartphone to an Expansion Hub



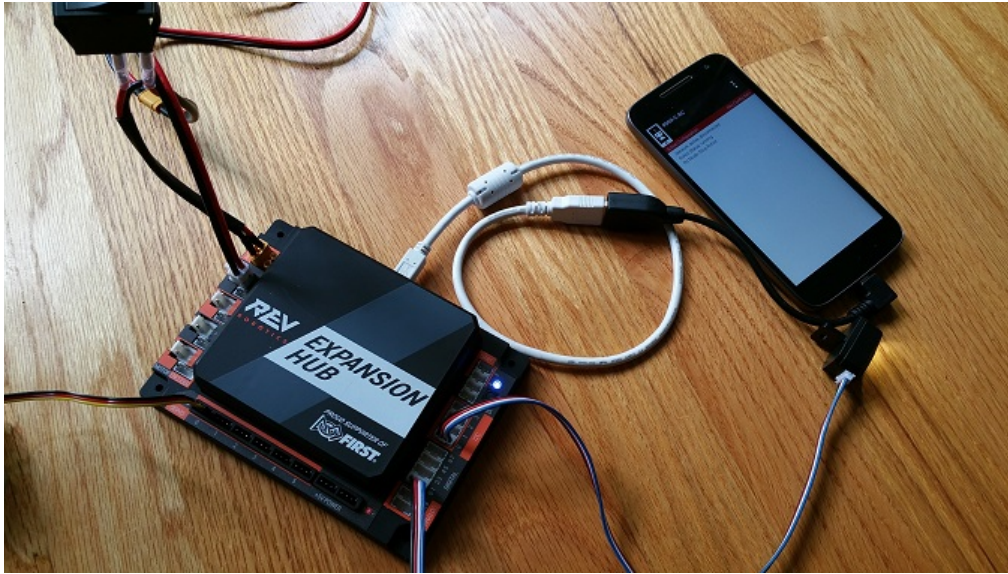
3. Plug the Type A end of the USB cable into the OTG adapter.



4. Verify that your Robot Controller smartphone is powered on and unlocked. Plug in the USB Micro OTG adapter into the OTG port of the Robot Controller phone.

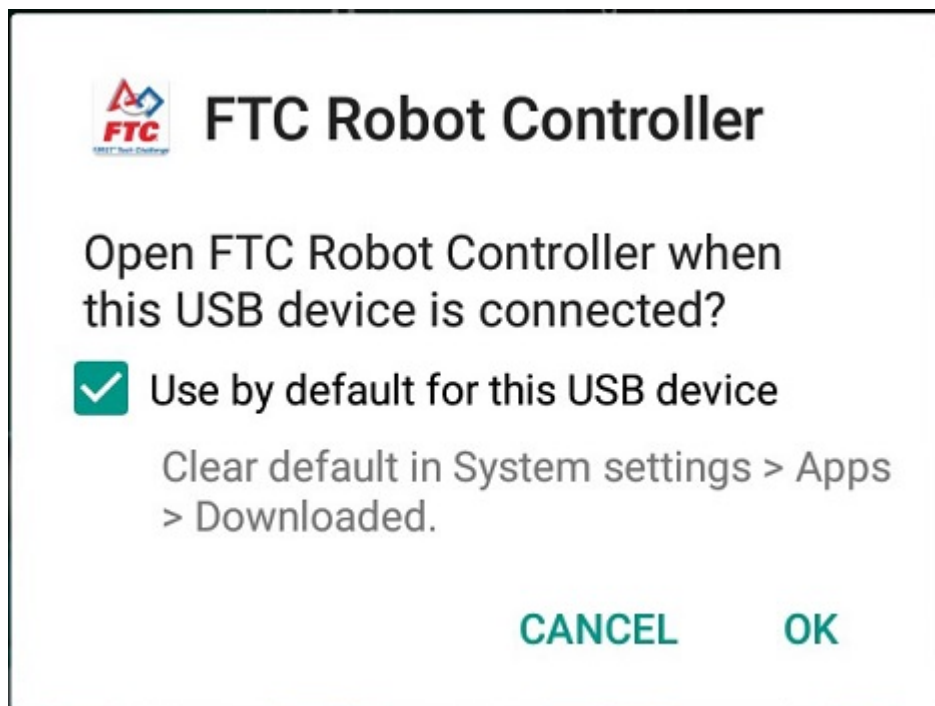


## Connecting an Android Smartphone to an Expansion Hub



Note that when the OTG adapter is plugged into the smartphone, the phone will detect the presence of the Expansion Hub and launch the Robot Controller app.

5. The first time you connect the Robot Controller smartphone to the Expansion Hub, the Android operating system should prompt you to ask if it is OK to associate the newly detected USB device (which is the Expansion Hub) with the FTC Robot Controller app.



## Connecting an Android Smartphone to an Expansion Hub

### Important Information!

You might be prompted multiple times to associate the USB hardware with the FTC Robot Controller. Whenever you are prompted by your phone with this message, you should always select the “Use by default for this USB device” option and hit the “OK” button to associate the USB device with the FTC Robot Controller app.

If you fail to make this association, then the Robot Controller app might not reliably connect to this Expansion Hub the next time you turn your system on.

### 9.3. Getting the Control Hub Ready

If you are using a Control Hub, you do not need to make any additional connections. You simply need to make sure that the Control Hub is powered on and paired to the Driver Station.

### 9.4. Creating a Configuration File Using the Driver Station

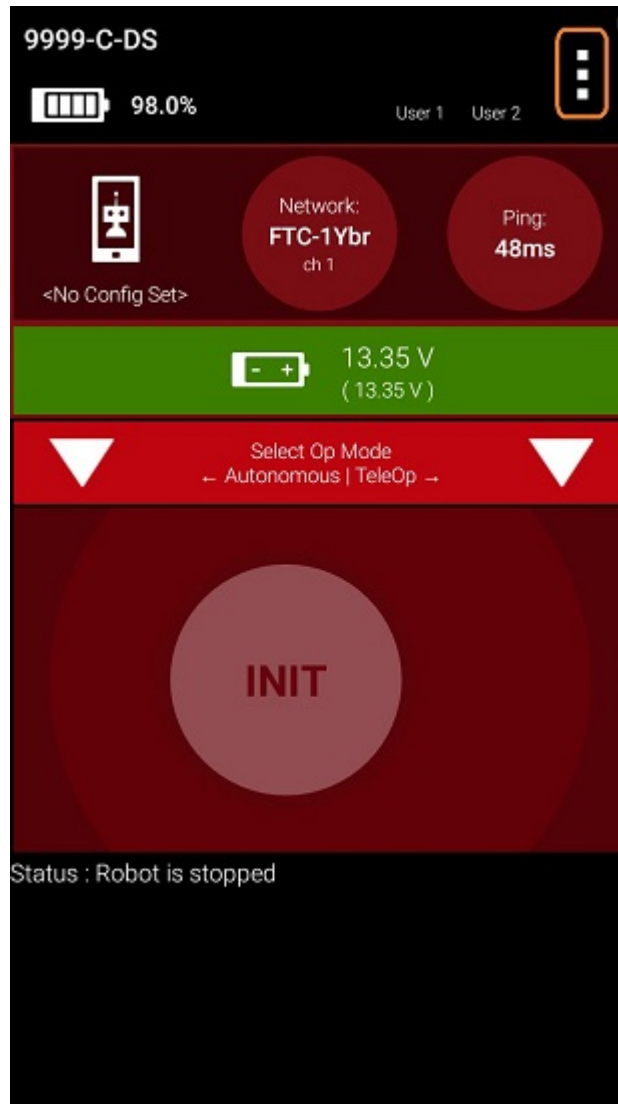
Although the configuration file needs to reside on the Robot Controller, for this tutorial we will use the Driver Station app to create the configuration file remotely. The Driver Station can be used to create a configuration file for a Control Hub or for an Android smartphone Robot Controller.

#### Creating a Configuration File on the Robot Controller using the Driver Station

1. Touch the three vertical dots in the upper right hand corner of the Driver Station app. This will launch a pop-up menu.

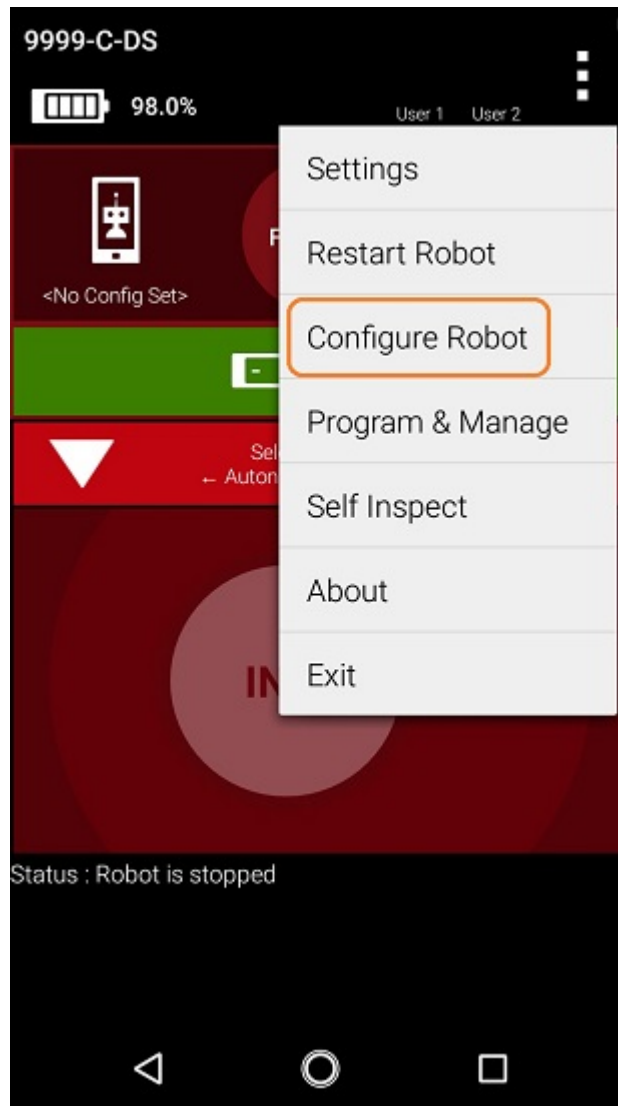


## Creating a Configuration File on the Robot Controller using the Driver Station



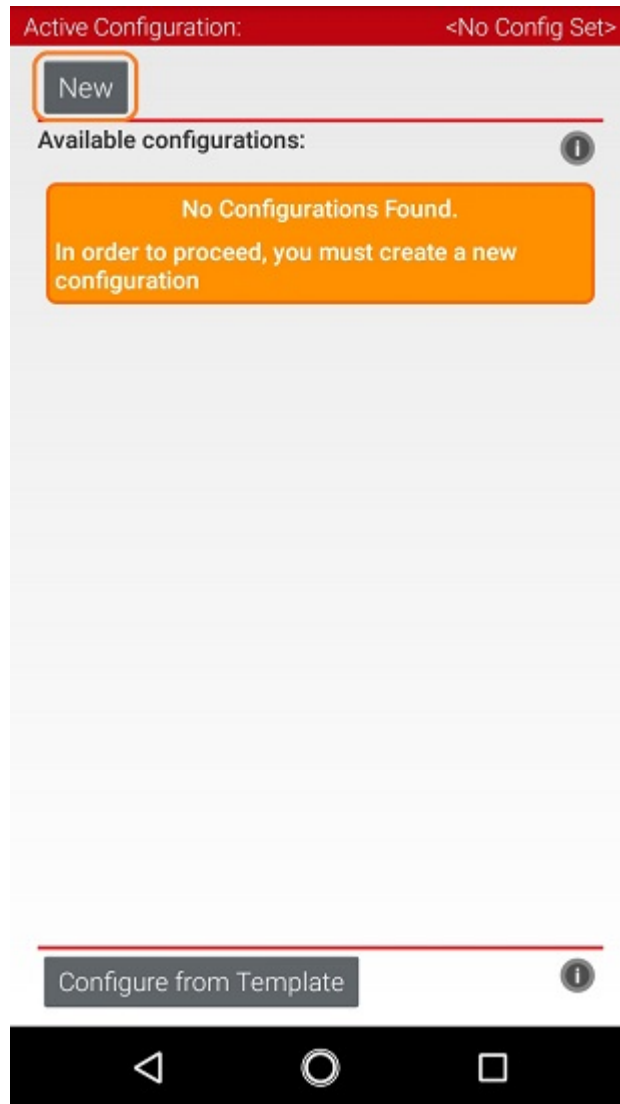
2. Select **Configure Robot** from the pop up menu to display the **Configuration** screen.

### Creating a Configuration File on the Robot Controller using the Driver Station



3. If your Robot Controller does not have any existing configuration files, the screen will display a message indicating that you need to create a file before proceeding.

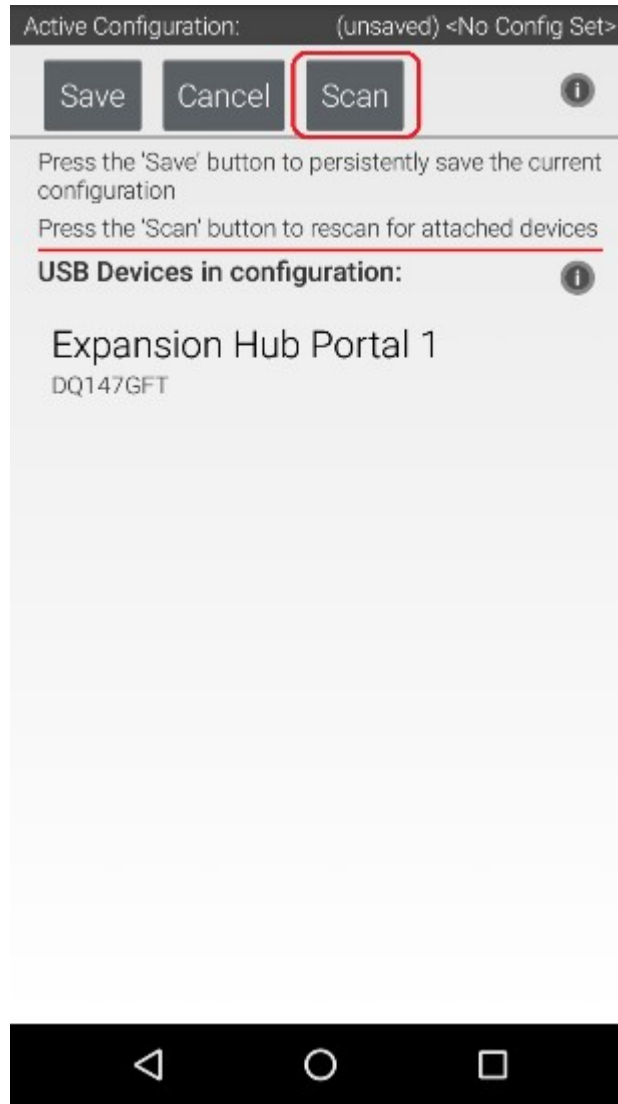
## Creating a Configuration File on the Robot Controller using the Driver Station



Hit the **New** button to create a new configuration file for your Robot Controller.

4. When the new configuration screen appears, the Robot Controller app will do a scan of the serial bus to see what devices are connected to the Robot Controller.

### Creating a Configuration File on the Robot Controller using the Driver Station



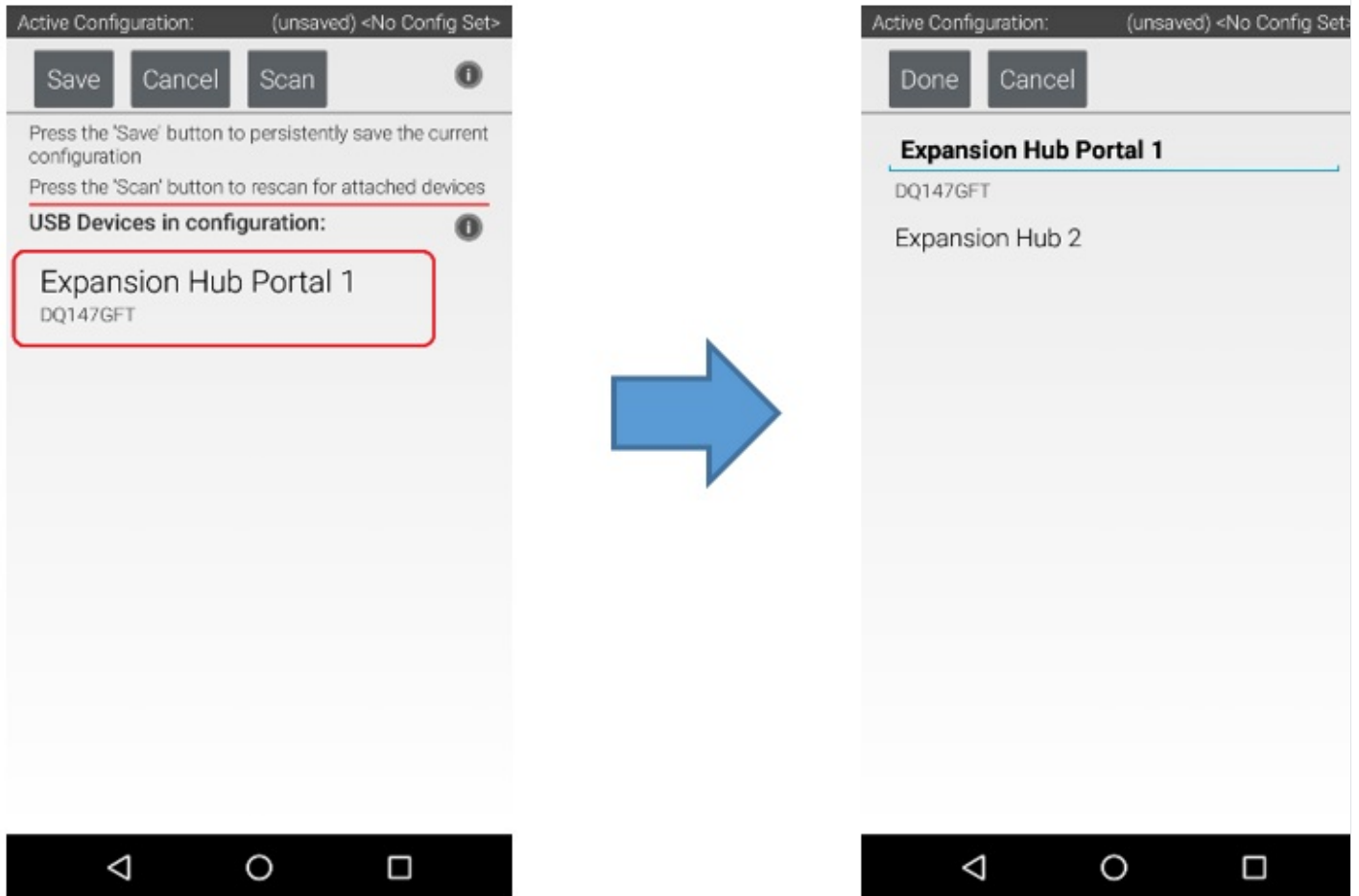
It will display the devices that it found in a list underneath the words “USB Devices in configuration.” You should see an entry that says something like “Expansion Hub Portal 1” in the list.

Your Expansion Hub is listed as a Portal because it is directly connected to the Robot Controller phone through the USB cable or in the case of the Control Hub through the internal serial bus.

If you do not see your Expansion Hub Portal listed and you are using a smartphone as a Robot Controller, check the wired connections to make sure they are secure and then press the Scan button one or two times more to see if the smartphone detects the device on a re-scan of the USB bus.

## Creating a Configuration File on the Robot Controller using the Driver Station

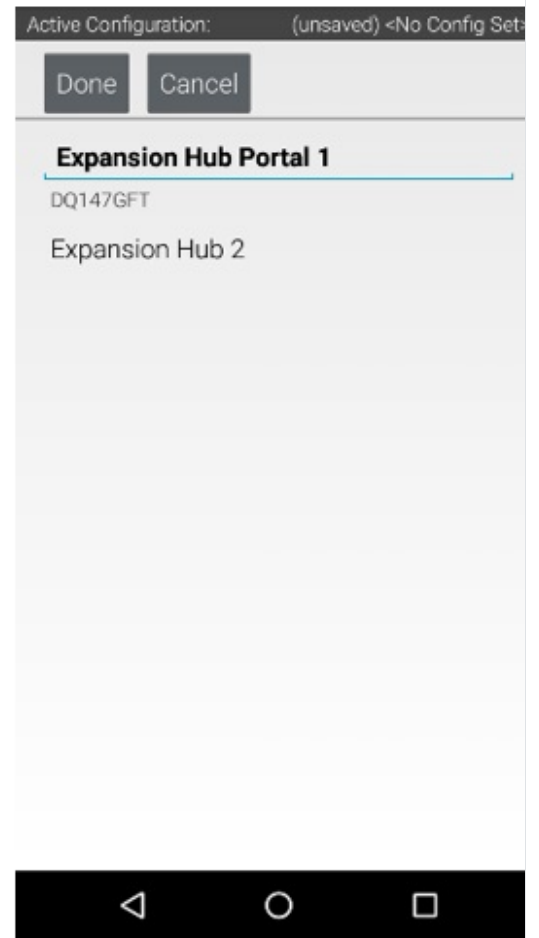
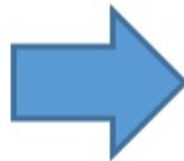
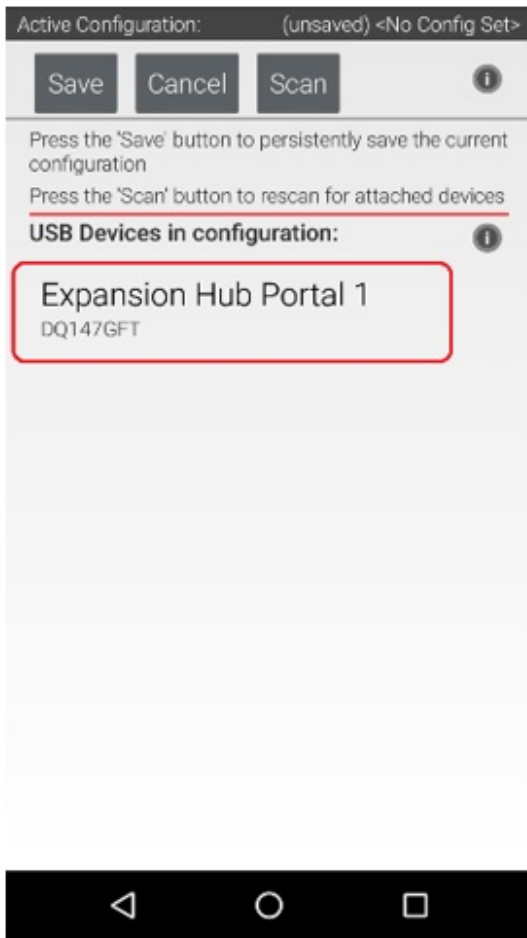
5. Touch the Portal listing (“Expansion Hub Portal 1” in this example) to display what Expansion Hubs are connected through this Portal.



Since we only have a single Expansion Hub connected, we should only see a single Expansion Hub configured (“Expansion Hub 2” in this example).

6. Touch the Expansion Hub listing (“Expansion Hub 2” in this example) to display the Input/Output ports for that device.

### Creating a Configuration File on the Robot Controller using the Driver Station



The screen should change and list all the motor, servo and sensor ports that are available on the selected Expansion Hub.

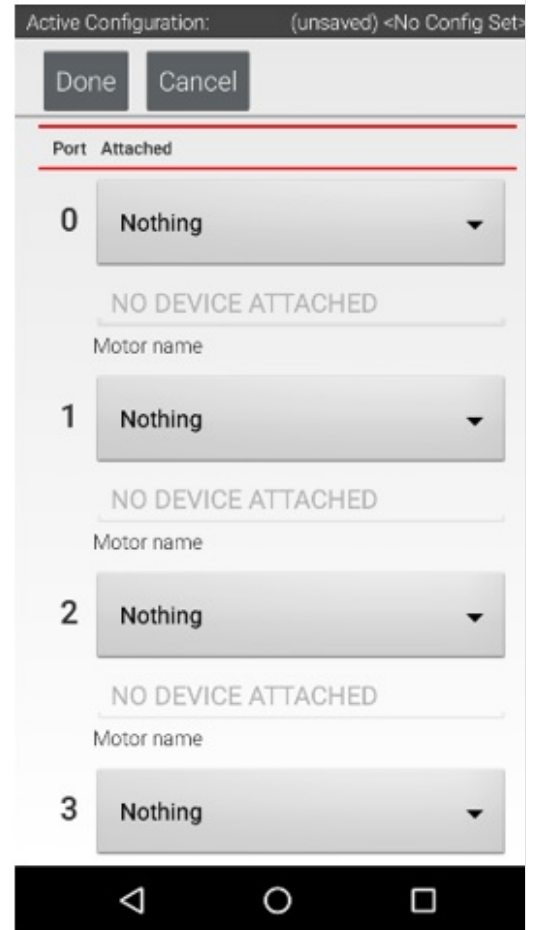
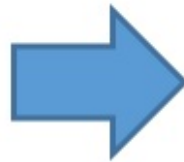
### 9.5. Configuring a DC Motor

Now that you've created a file, you will need to add a DC Motor to the configuration file.

**Important Note:** At this point, although you have created your configuration file, you have not yet saved its contents to the Robot Controller. You will save the configuration file in a [later step](#).

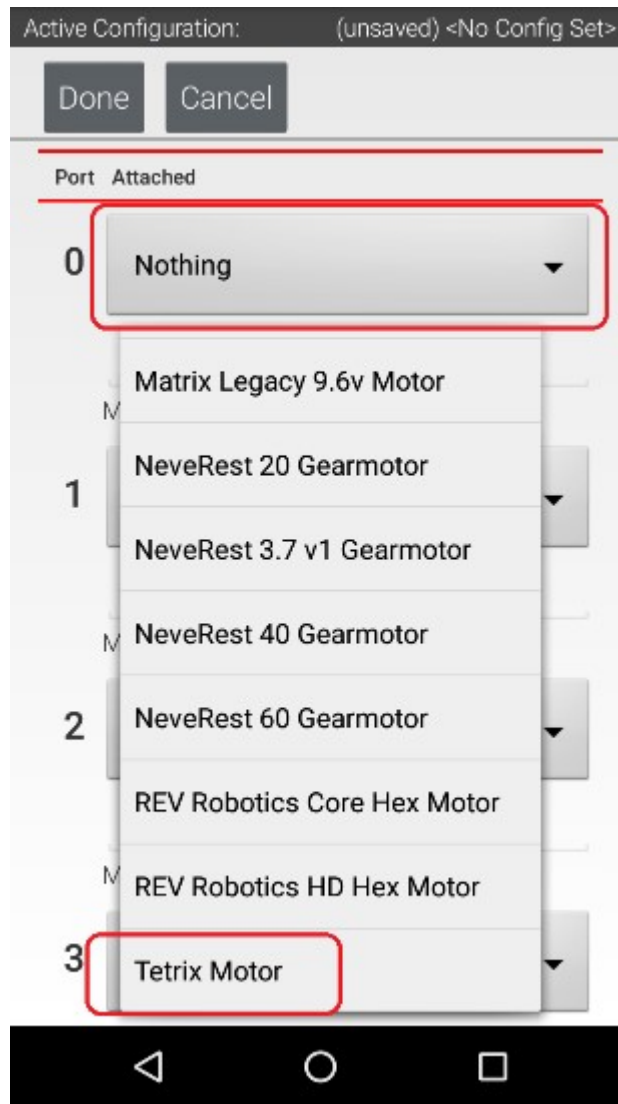
## Configuring a DC Motor

1. Touch the word **Motors** on the screen to display the Motor Configuration screen.



2. Since we installed our motor onto port #0 of the Expansion Hub, use the dropdown control for port 0 to select the motor type (Tetrix Motor for this example).

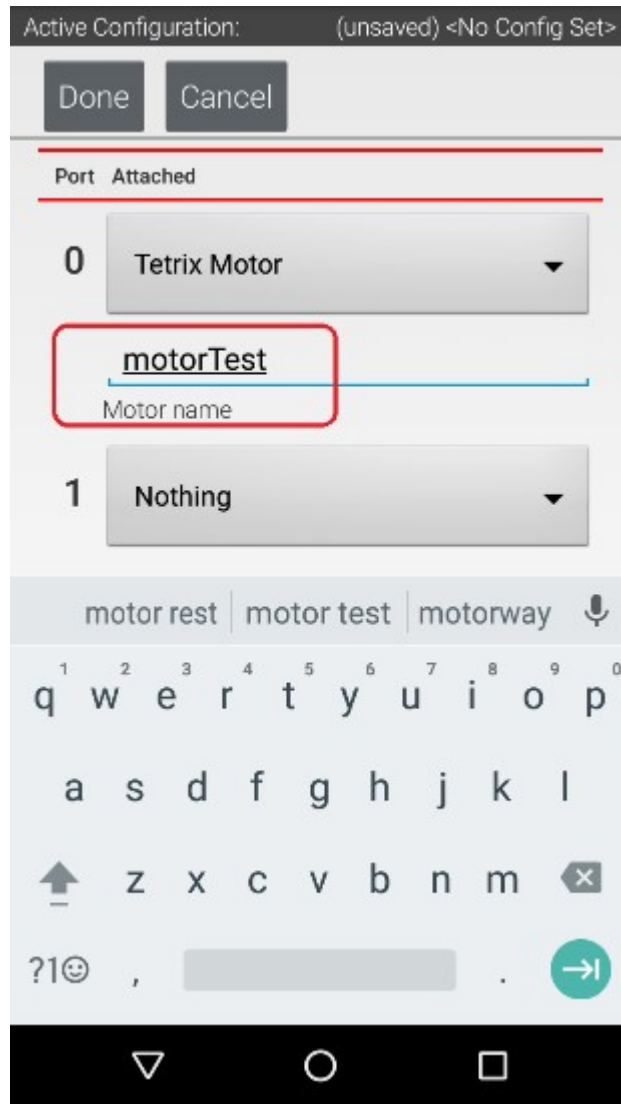
### Configuring a DC Motor



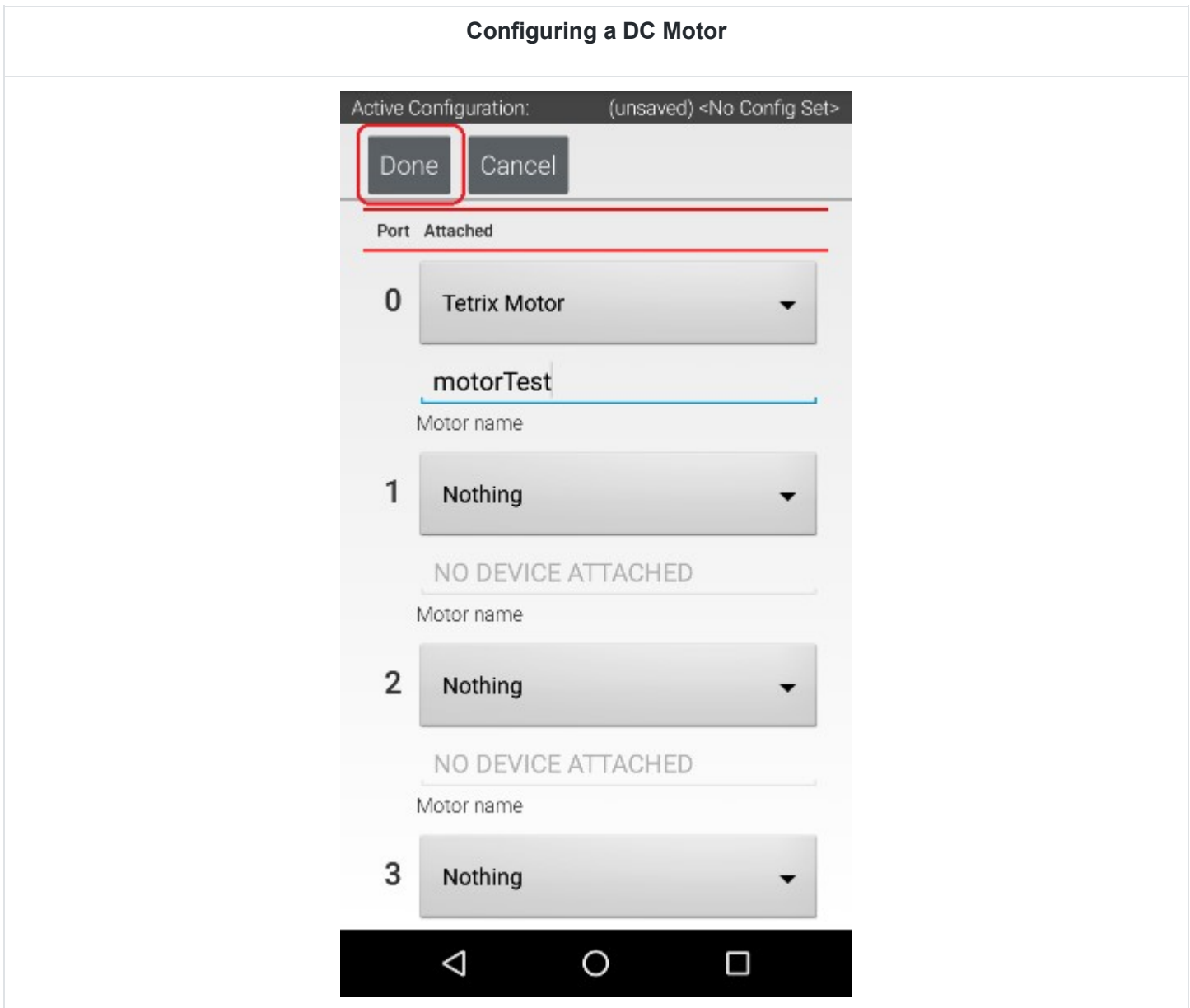
3. Use the touch screen keypad to specify a name for your motor ("motorTest" in this example).



### Configuring a DC Motor



4. Press the **Done** button to complete the motor configuration. The app should return to the previous screen.



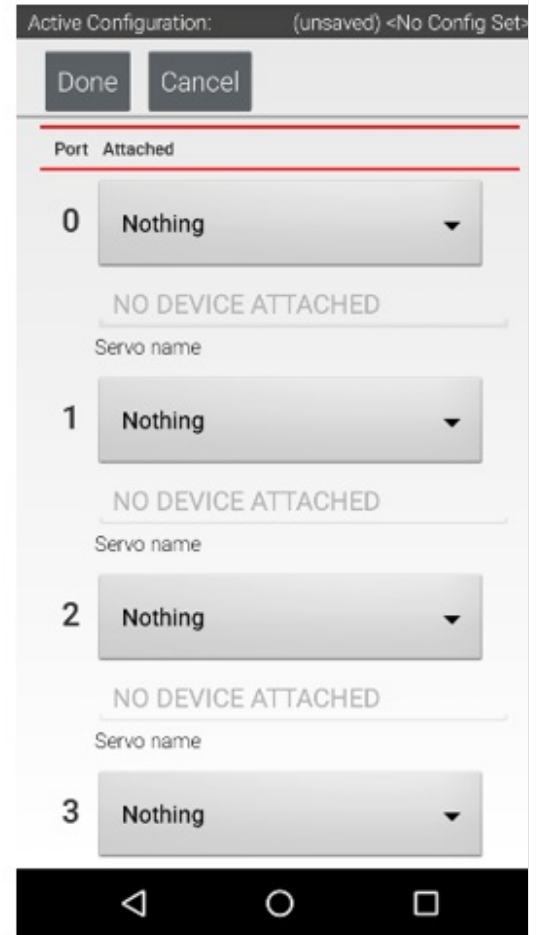
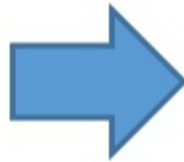
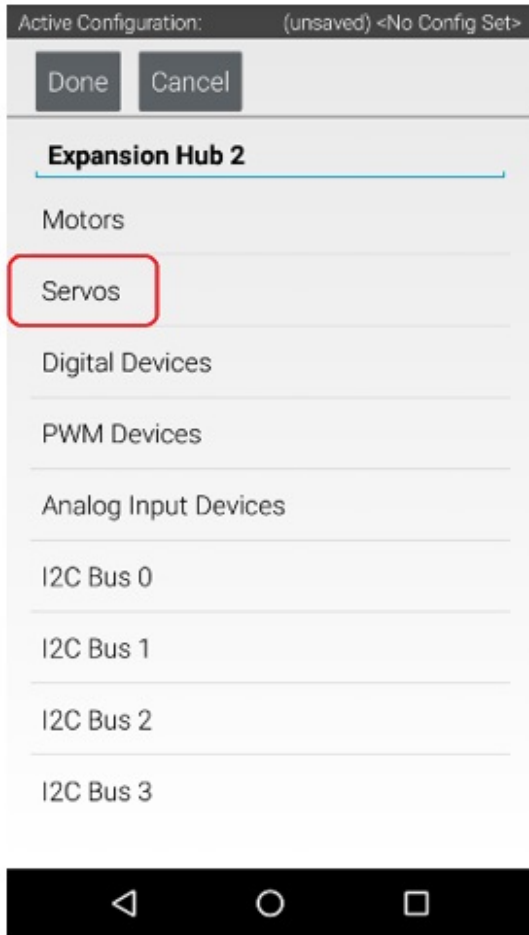
### 9.6. Configuring a Servo

You will also want to add a servo to the configuration file. In this example, you are using a standard 180-degree servo.

#### Configuring a Servo

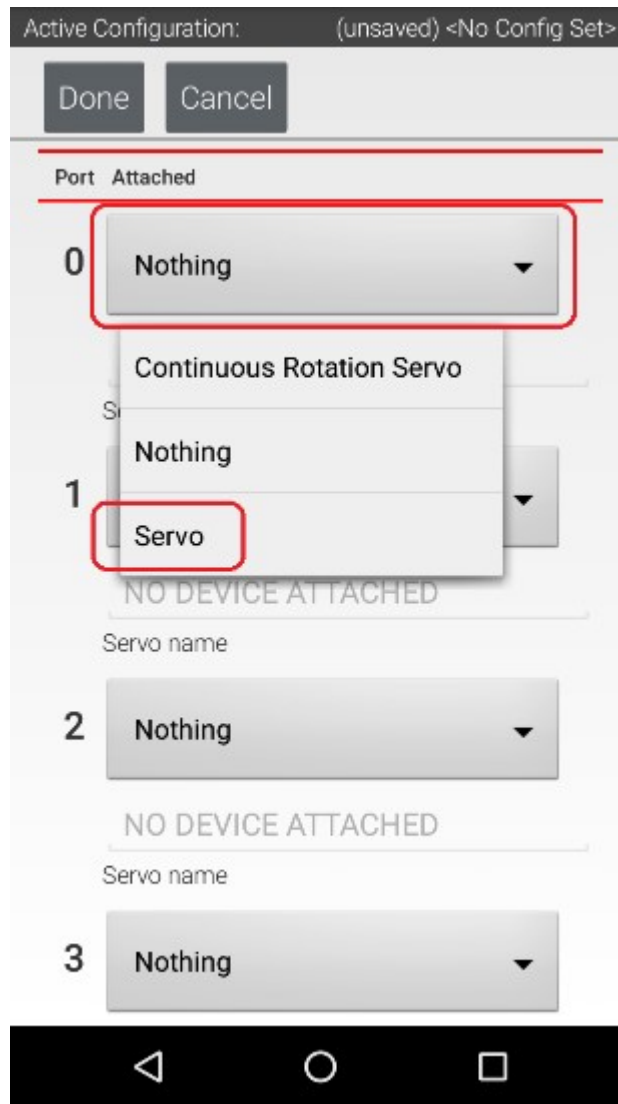
1. Touch on the word **Servos** on the screen to display the **Servo Configuration** screen.

## Configuring a Servo



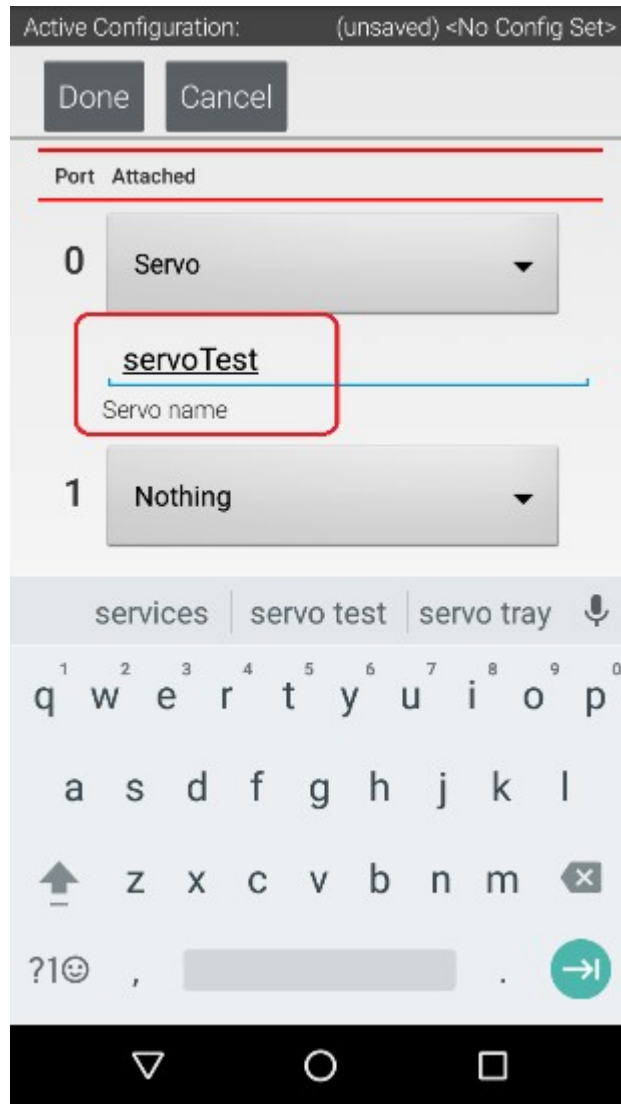
2. Use the dropdown control to select “Servo” as the servo type for port #0.

### Configuring a Servo



3. Use the touch pad to specify the name of the servo (“servoTest” for this example) for port #0.

### Configuring a Servo



4. Press the **Done** button to complete the servo configuration. The app should return to the previous screen.

### Configuring a Servo



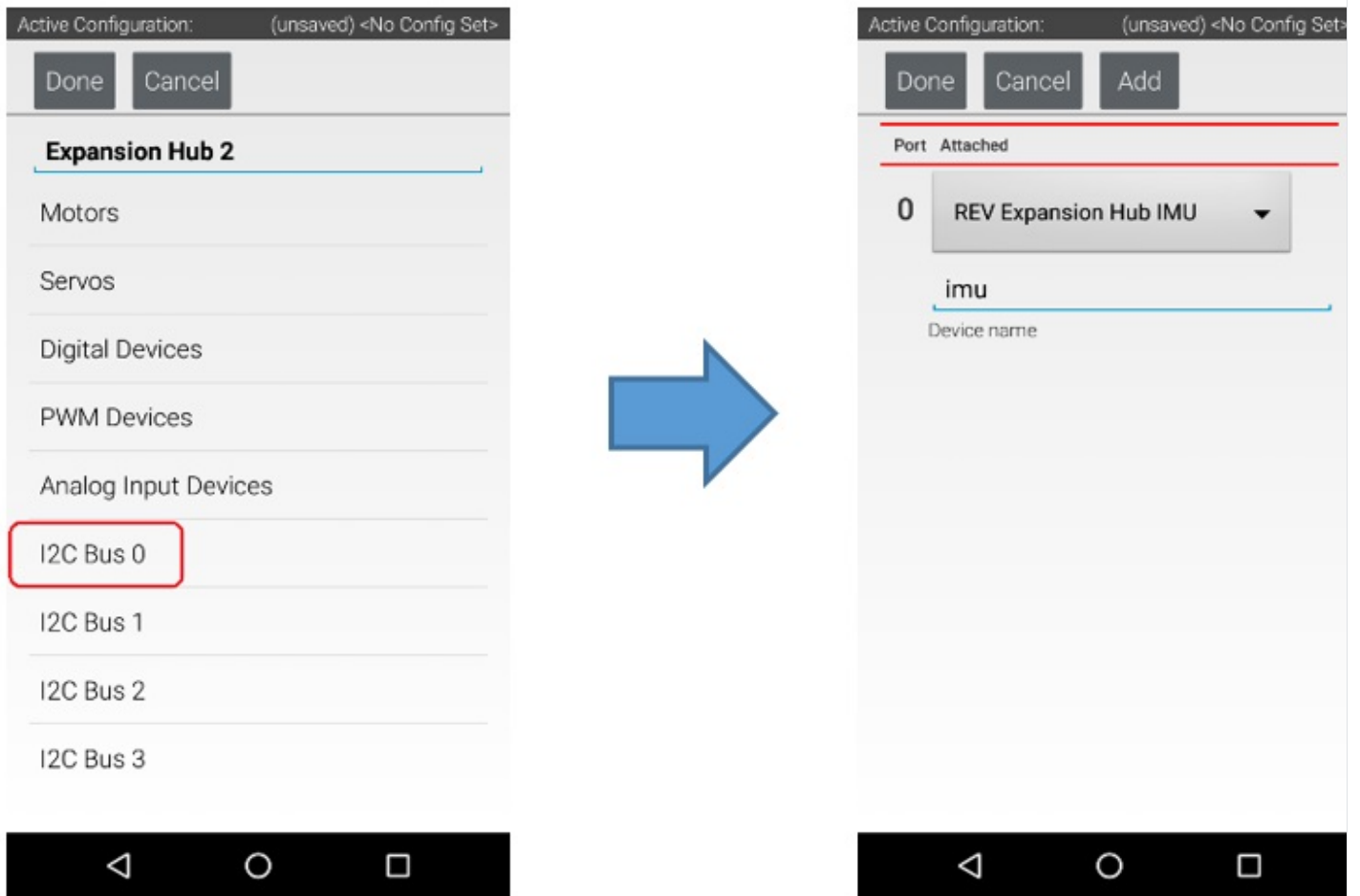
### 9.7. Configuring a Color Distance Sensor

The REV Robotics Color Distance Sensor is an I2C sensor. It actually combines two sensor functions into a single device. It is a color sensor, that can determine the color of an object. It is also a distance or range sensor, that can be used to measure short range distances. Note that in this tutorial, the word "distance" is used interchangeably with the word "range".

#### Configuring a Color Distance Sensor

1. Touch the words **I2C Bus 0** on the screen to launch the I2C configuration screen for this I2C bus.

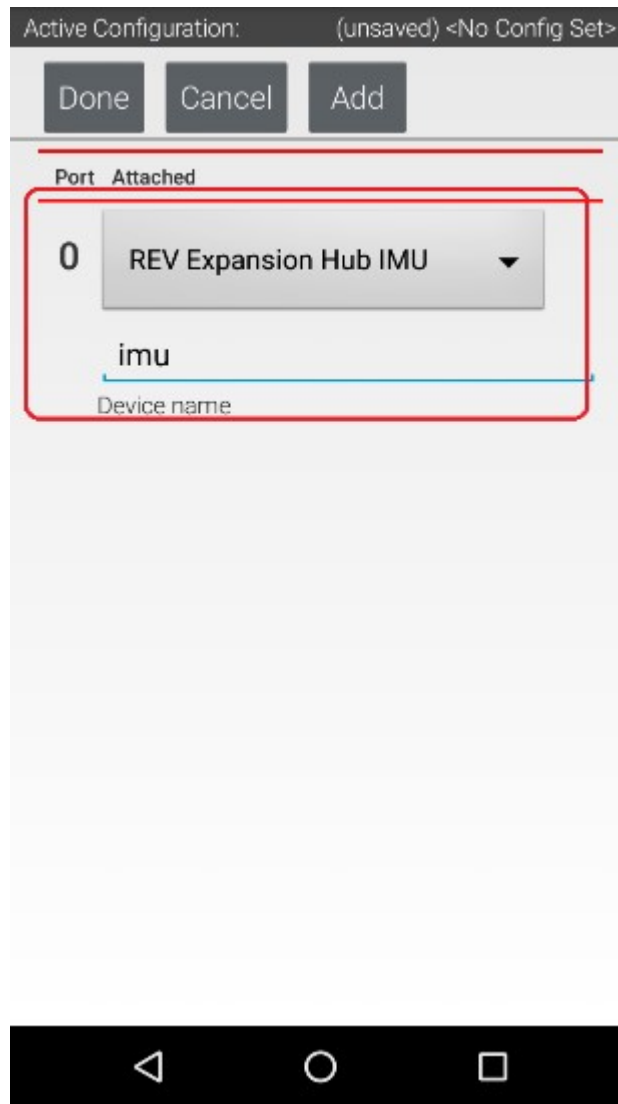
## Configuring a Color Distance Sensor



The Expansion Hub has four independent I2C buses, labeled “0” through “3”. In this example, since you connected the Color Sensor to the port labeled “0”, it resides on I2C Bus 0.

2. Look at the **I2C Bus 0** screen. There should already be a sensor configured for this bus. The Expansion Hub has its own built-in inertial measurement unit (IMU) sensor. This sensor can be used to determine the orientation of a robot, as well as measure the accelerations on a robot.

### Configuring a Color Distance Sensor

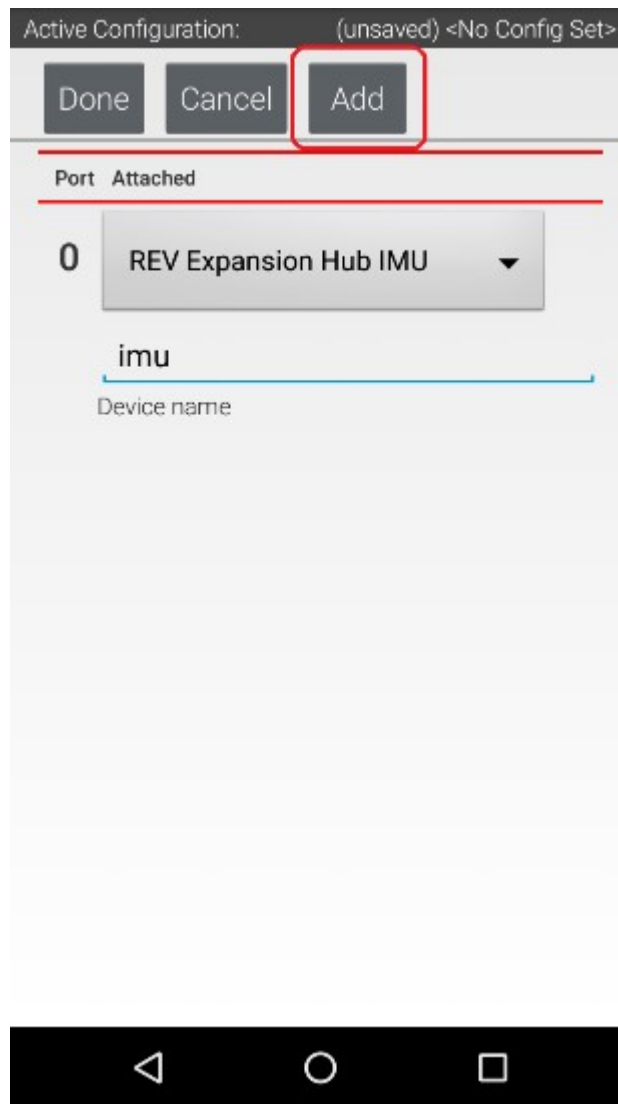


The built-in IMU is internally connected to I2C Bus 0 on each Expansion Hub. Whenever you configure an Expansion Hub using the Robot Controller, the app automatically configures the IMU for I2C Bus 0. You will need to add another I2C device for this bus to be able to configure the color sensor.

3. Press the **Add** button to add another I2C device to this bus.

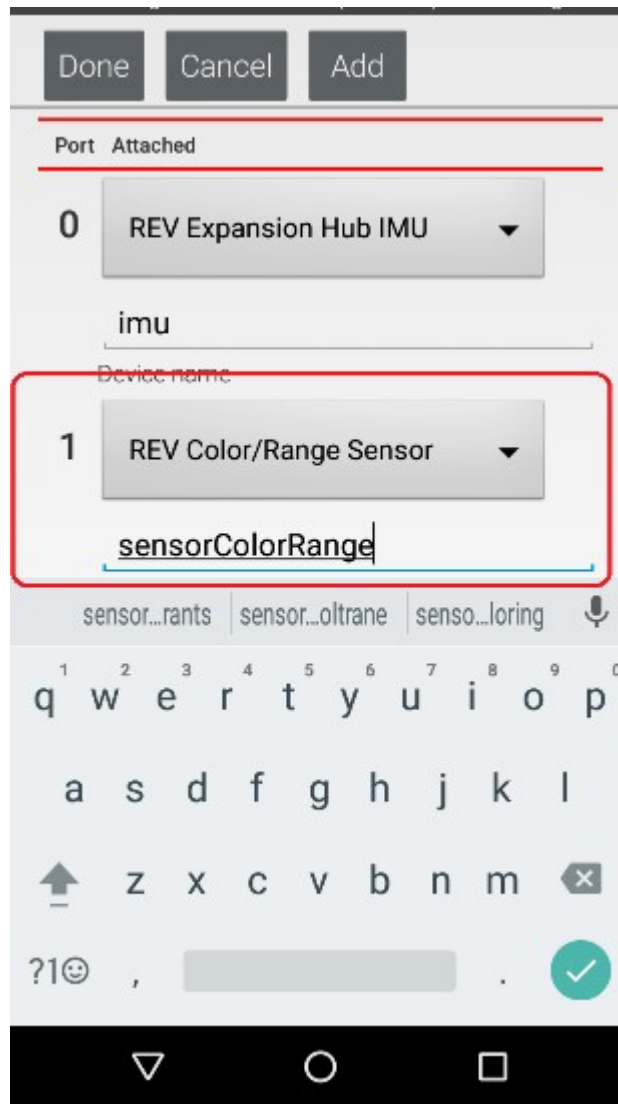


### Configuring a Color Distance Sensor



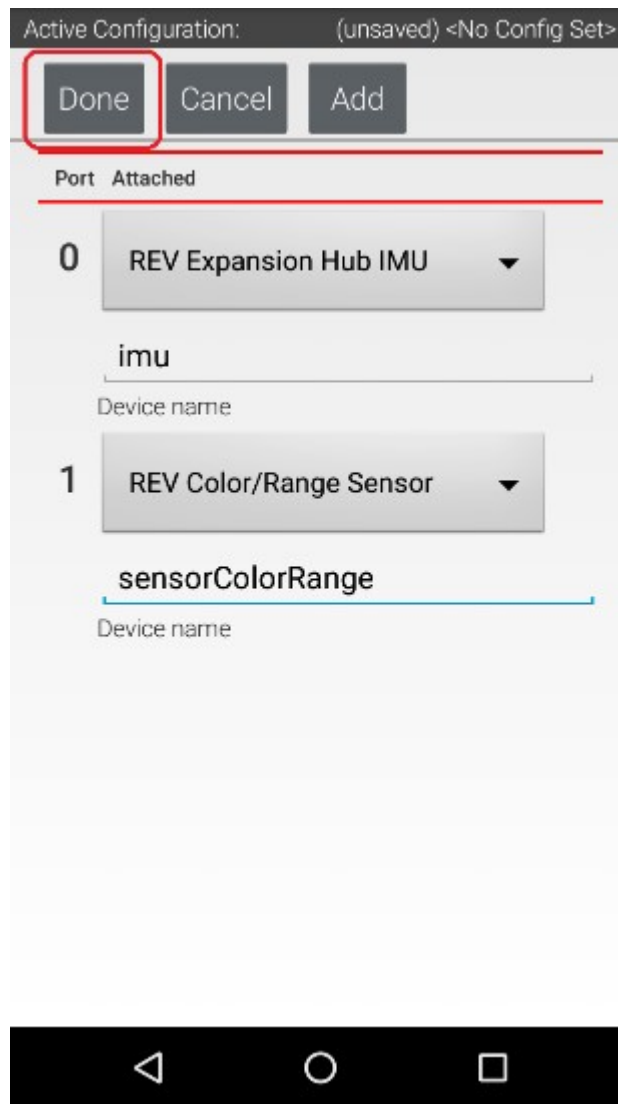
4. Select "REV Color/Range Sensor" from the dropdown selector for this new device. Use the touchscreen keyboard to name this device "sensorColorRange".

### Configuring a Color Distance Sensor



5. Press the **Done** button to complete the I2C sensor configuration. The app should return to the previous screen.

### Configuring a Color Distance Sensor



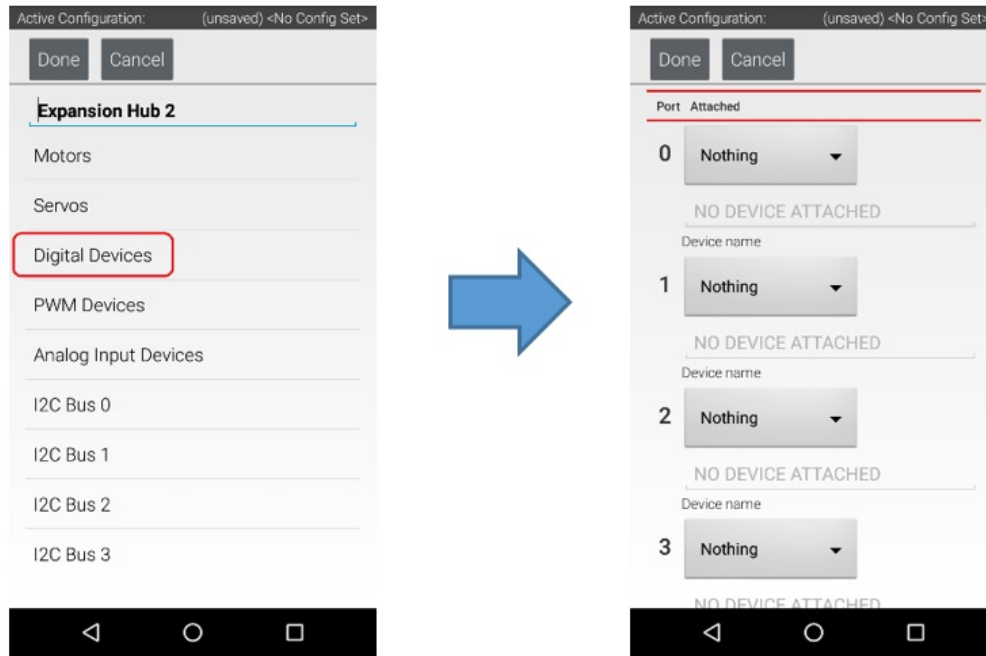
### 9.8. Configuring a Digital Touch Sensor

The REV Robotics Touch Sensor is a digital sensor. An Op Mode can query the Touch Sensor to see if its button is being pressed or not.

#### Configuring a Digital Touch Sensor

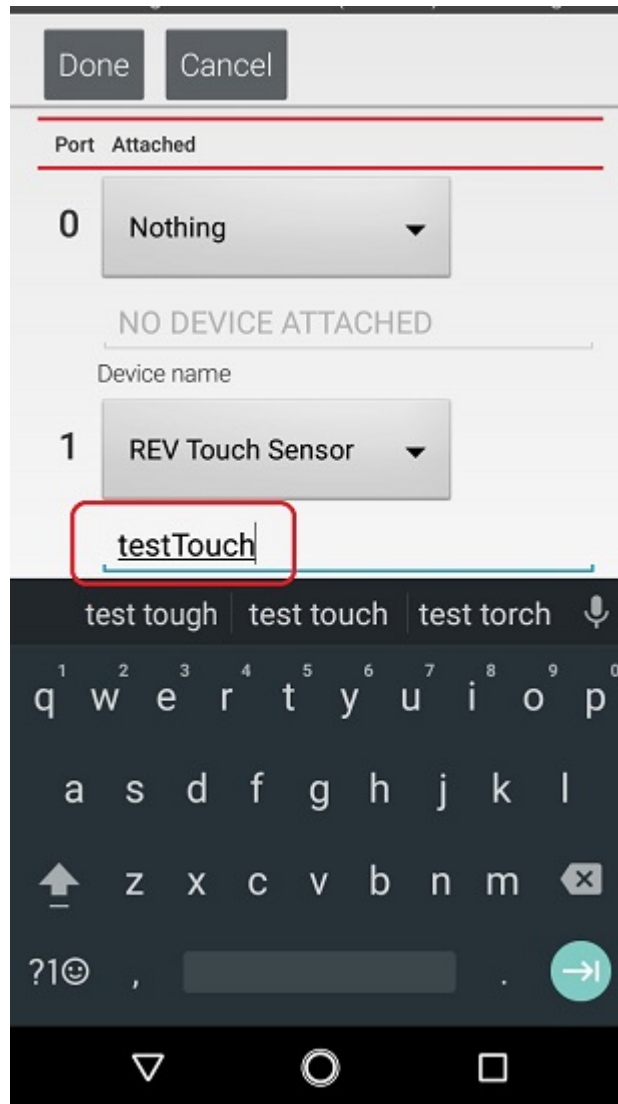
1. Touch the words **Digital Devices** on the screen to launch the Digital I/O configuration screen.

## Configuring a Digital Touch Sensor



2. Use the touch screen to add a “REV Touch Sensor” for port #1 and name the device “testTouch”.

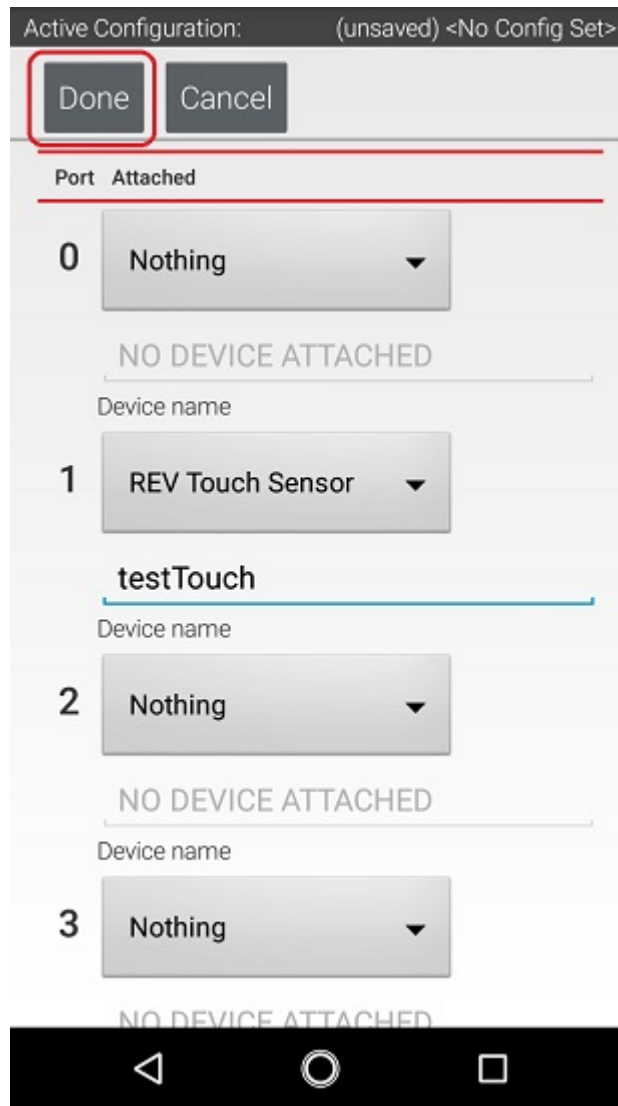
### Configuring a Digital Touch Sensor



Notice that we are configuring the Touch Sensor on port #1 instead of port #0. This is because when the REV Robotics Touch Sensor is connected to a digital port using a standard 4-wire JST sensor cable, it is the second digital pin that is connected. The first pin remains disconnected.

3. Press the **Done** button to return to the previous screen.

### Configuring a Digital Touch Sensor



### 9.9. Saving the Configuration Information

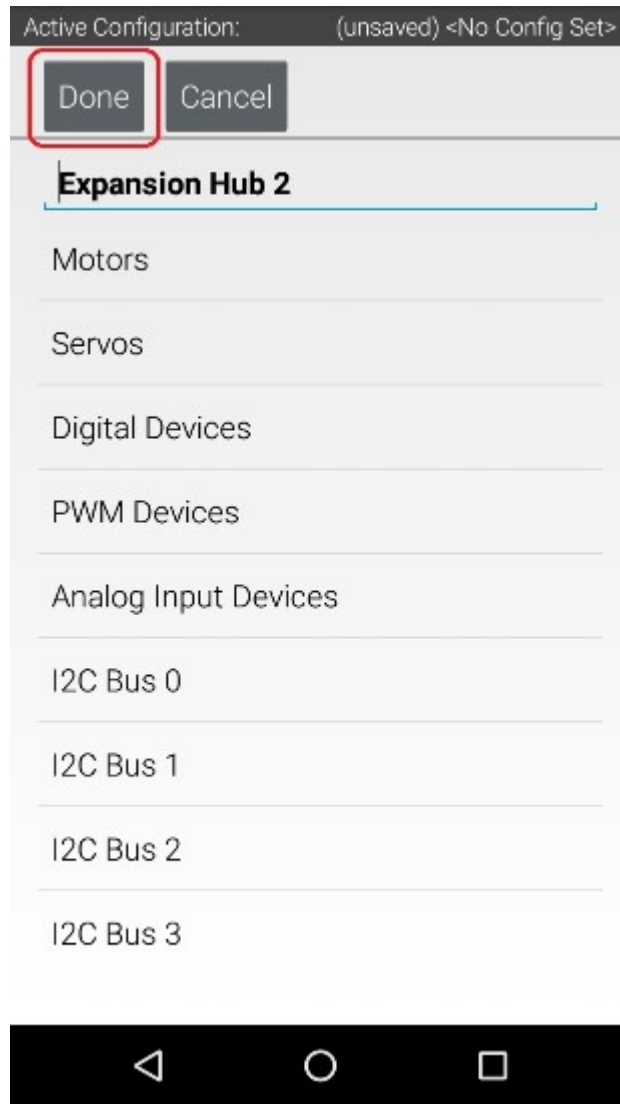
Once you have configured your hardware, you must save the information to the configuration file. If you do not save this information, it will be lost and the robot controller will be unable to communicate with your hardware.

#### Saving the Configuration Information

1. Press the **Done** button to go up one level in the configuration screens.

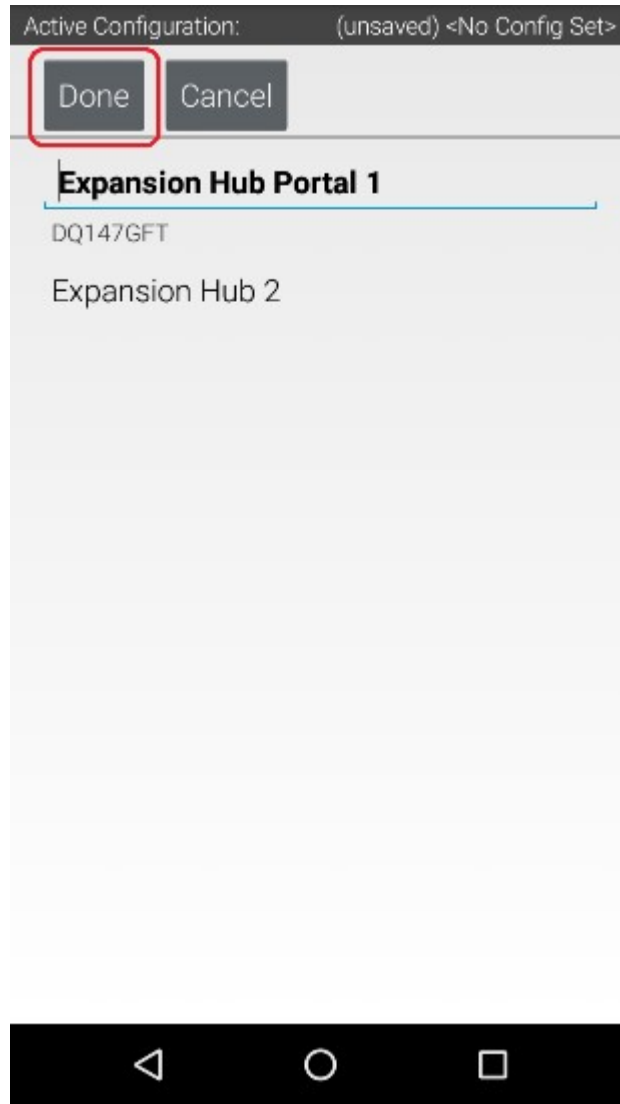


## Saving the Configuration Information



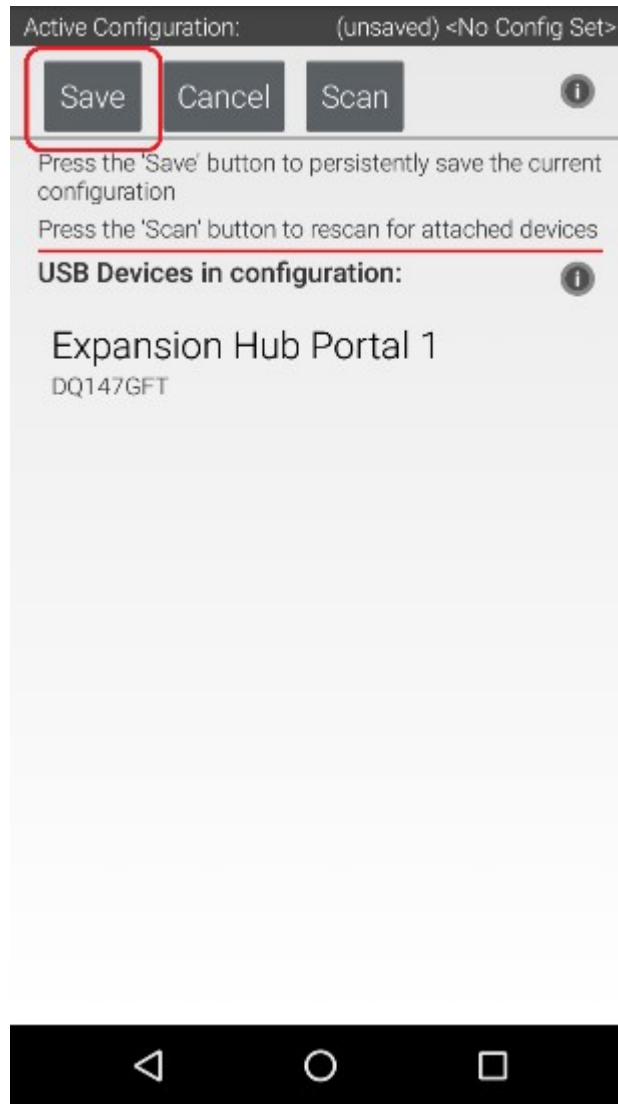
2. Press the **Done** button again to return to the highest level in the configuration screens.

### Saving the Configuration Information



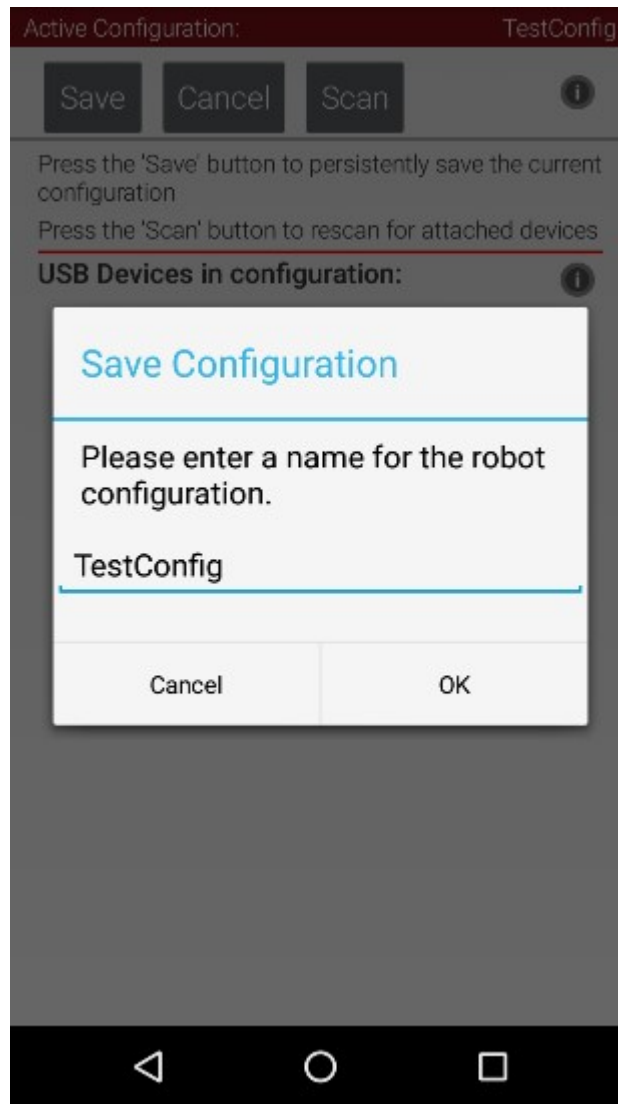
3. Press the **Save** button.

### Saving the Configuration Information



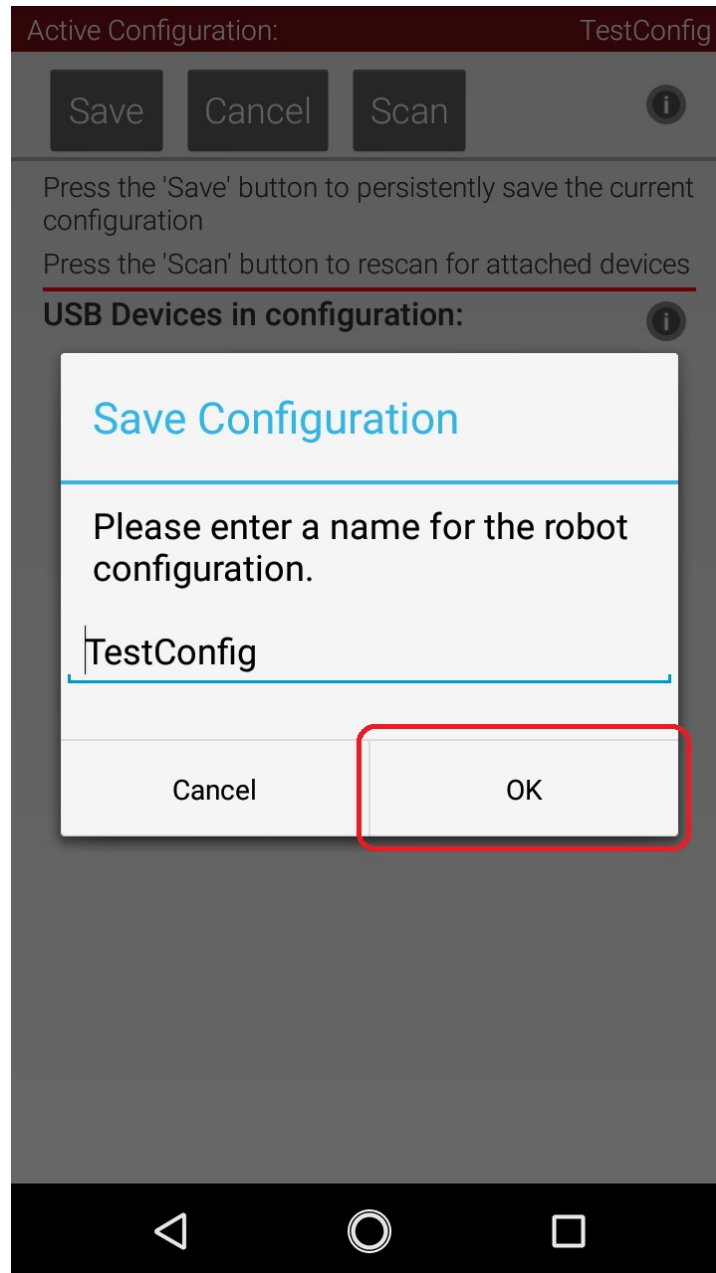
4. When prompted, specify a configuration file name using the touchscreen's keypad (use "TestConfig" for this example).

## Saving the Configuration Information



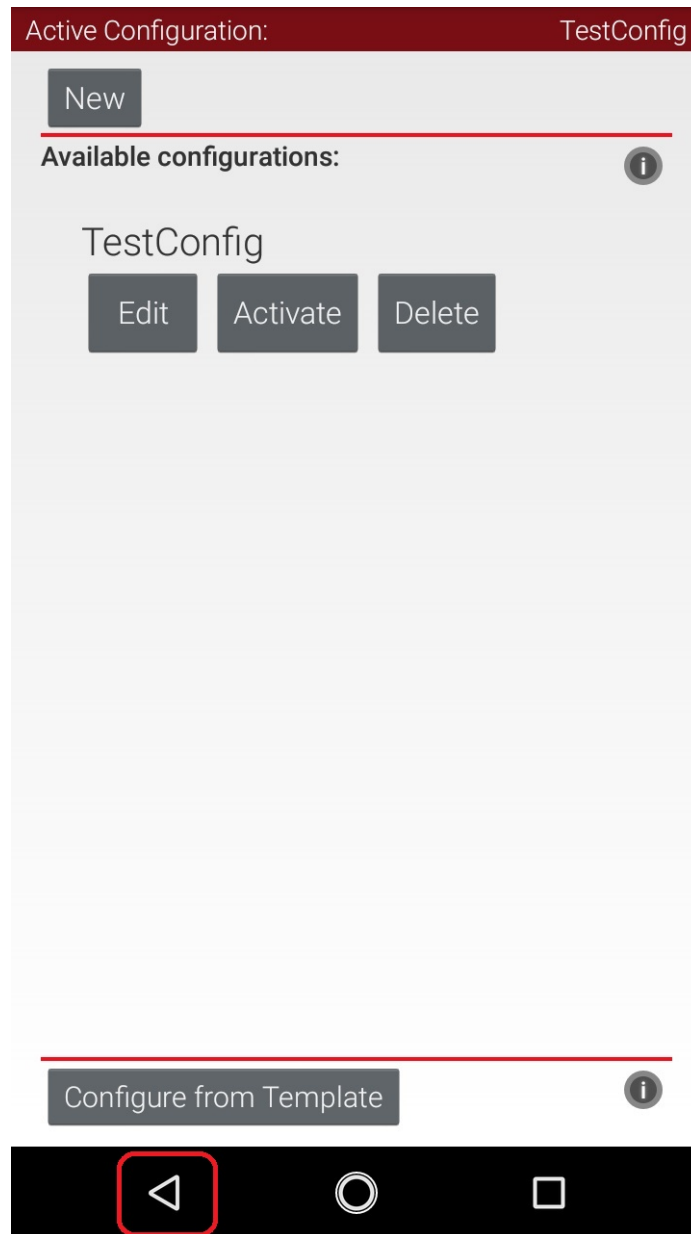
5. Press the **OK** button to save your configuration information using that file name.

### Saving the Configuration Information



6. After the configuration file has been saved, touch the Android back-arrow button to return to the main screen of the app.

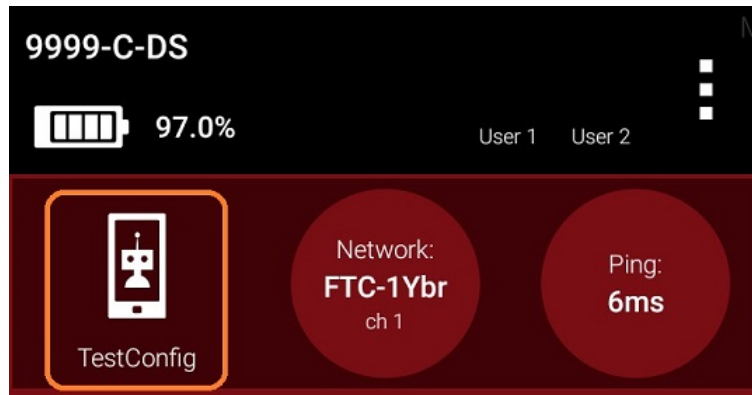
### Saving the Configuration Information



7. Verify that the configuration file is the active configuration file on the main Driver Station screen.



### Saving the Configuration Information



## 10. Installing Android Studio

### 10.1. Android Developer Website

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 available on the Windows, MacOS, and Linux operating systems.

### 10.2. 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>

### 10.3. Java Development Kit

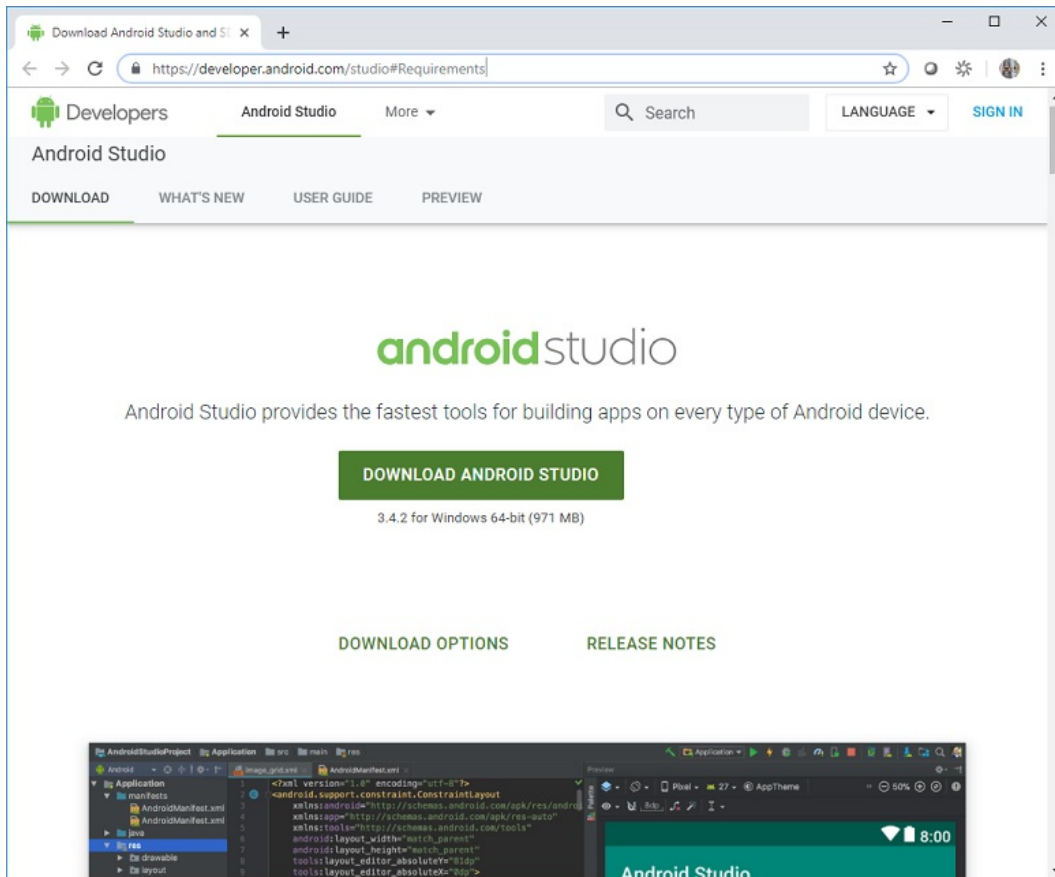
Earlier versions of Android Studio required that the user install the Java Development Kit software separately. Current versions of Android Studio incorporate the Java development software as part of the entire install package. It is no longer necessary (or recommended) to install the Java Development Kit separately. Instead, it is recommended that you use the Java Development Kit that is included with Android Studio.

### 10.4. Downloading and Installing Android Studio

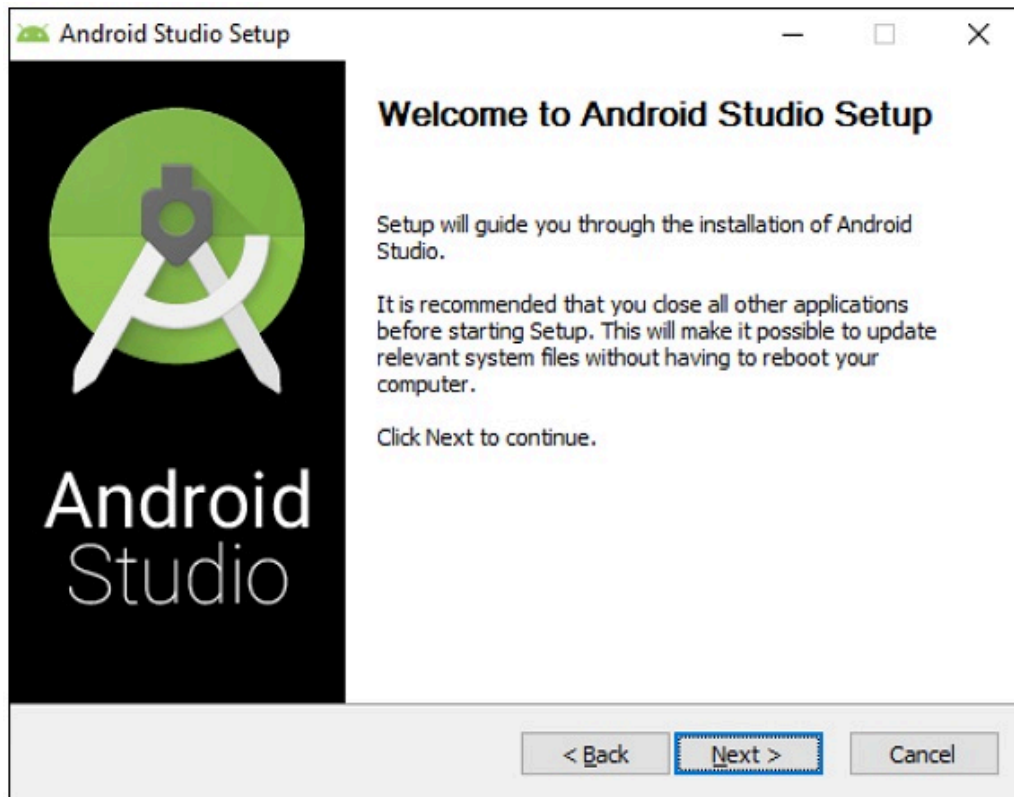
Once you have verified that your laptop satisfies the minimum system requirements, you can go to the Android developer's website to download and install Android Studio:

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

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 on the Android Developer webpage to download the software.



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

## 11. Disabling Android Studio Instant Run

---

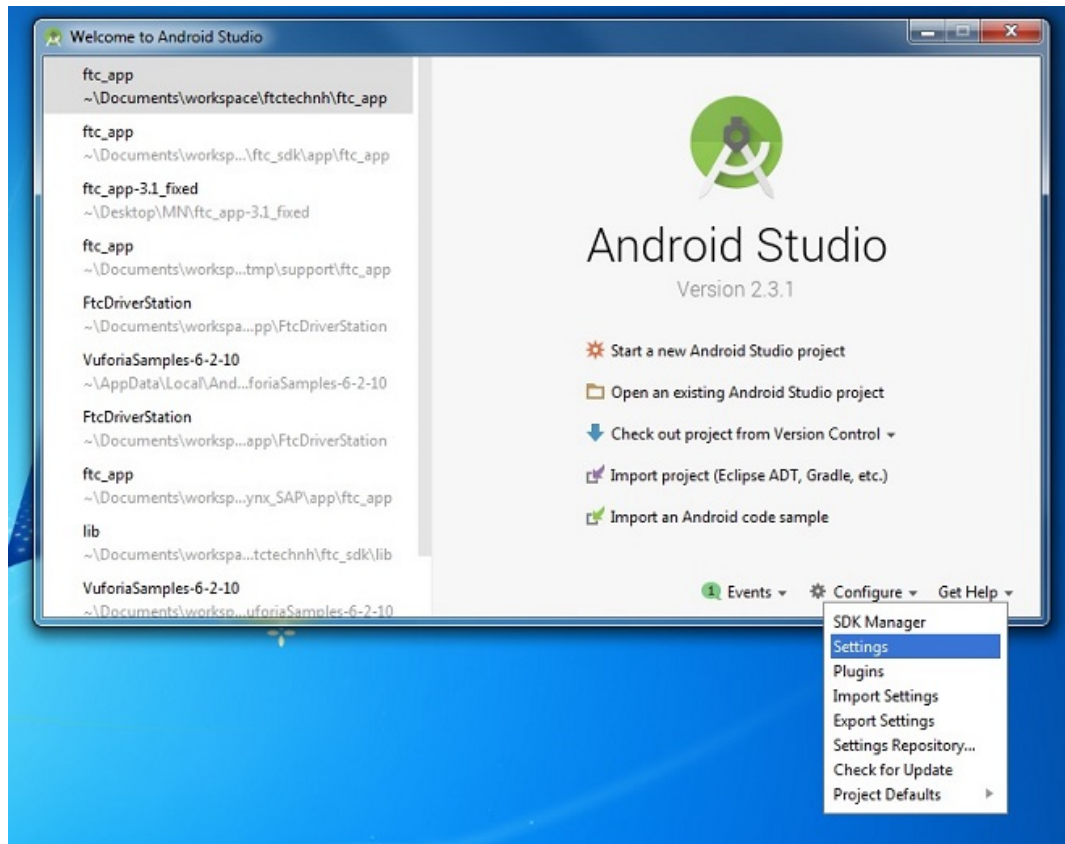
### 11.1. Introduction

If you are an Android Studio user, one of **the most important steps to take** is to disable Android Studio Instant Run. Instant Run is a feature that is designed to streamline the development process by reducing the time to apply code changes to your app. Unfortunately, Instant Run is limited in function and when used with the *FIRST* Tech Challenge Android Studio project folder, can cause **severe** and **difficult-to-troubleshoot** problems.

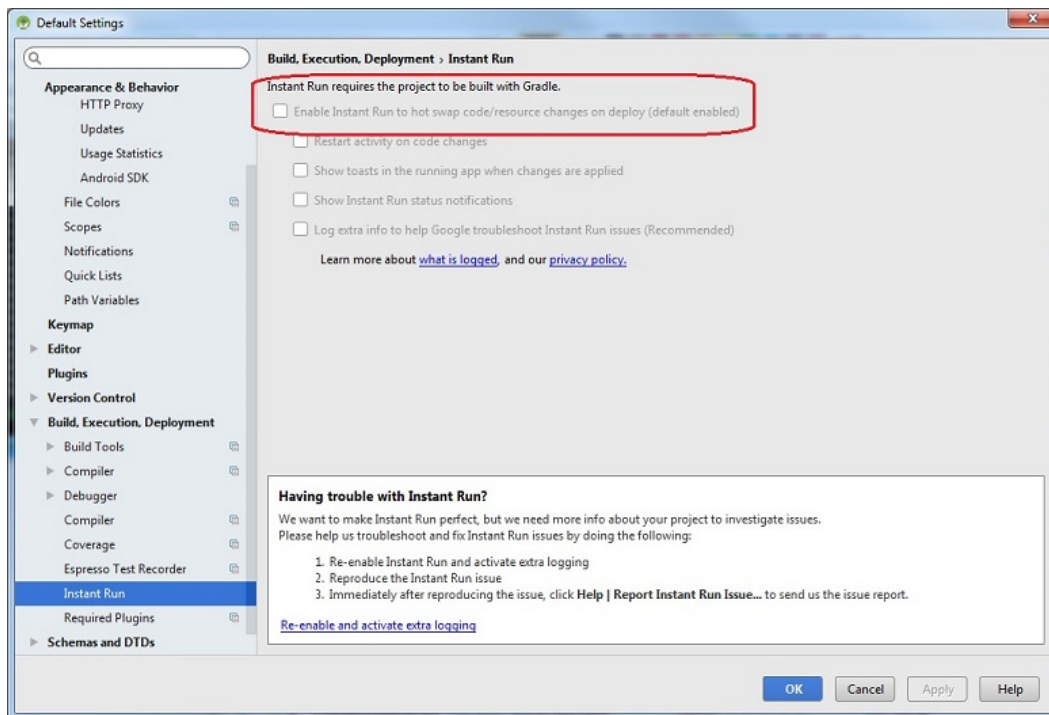
Teams who use Android Studio **must** disable Instant Run.

### 11.2. Locating Instant Run Settings

When you first launch Android Studio a Welcome screen should appear. You can navigate to the Instant Run Settings from this Welcome screen by selecting the "Configure->Settings" item from the "Configure" dropdown list in the lower right hand corner of the screen.



On the left hand side of the Settings window, there should be a category called "Build, Execution, Deployment". Within this category, click on the "Instant Run" subcategory to display the Instant Run settings for your Android Studio installation. By default, Instant Run is enabled when you first install Android Studio. Uncheck the "Enable Instant Run to hot swap code/resource changes on deploy (default enabled)" option and then click on the "OK" button to disable Instant Run.



### 11.3. Additional Information

The Google Android Developer website has additional information about Instant Run. It also has instructions on how to disable this feature:

<https://developer.android.com/studio/run/index.html#disable-ir>

## 12. Downloading the Android Studio Project Folder

### 12.1. 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/>

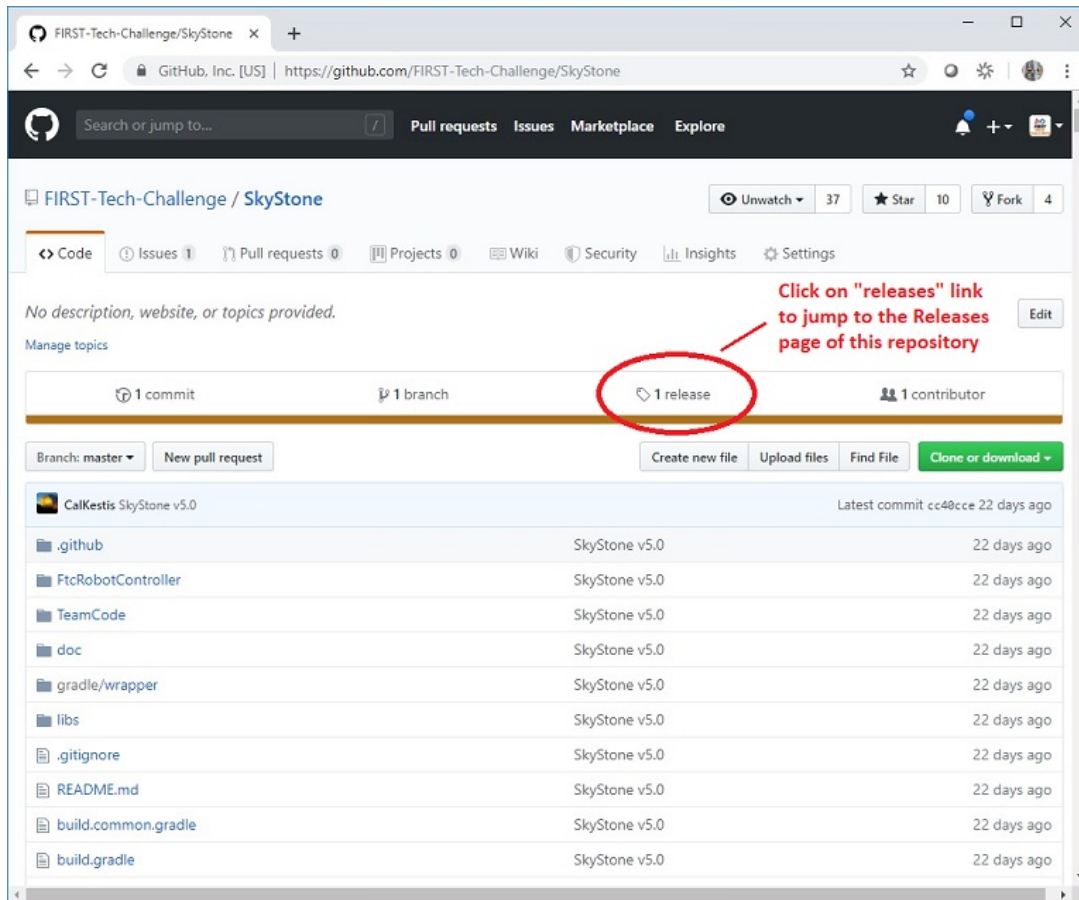
For the 2019-2020 season and beyond, each new FIRST Tech Challenge season has its own unique public GitHub repository.

The current season's public repository can be found at the following address:

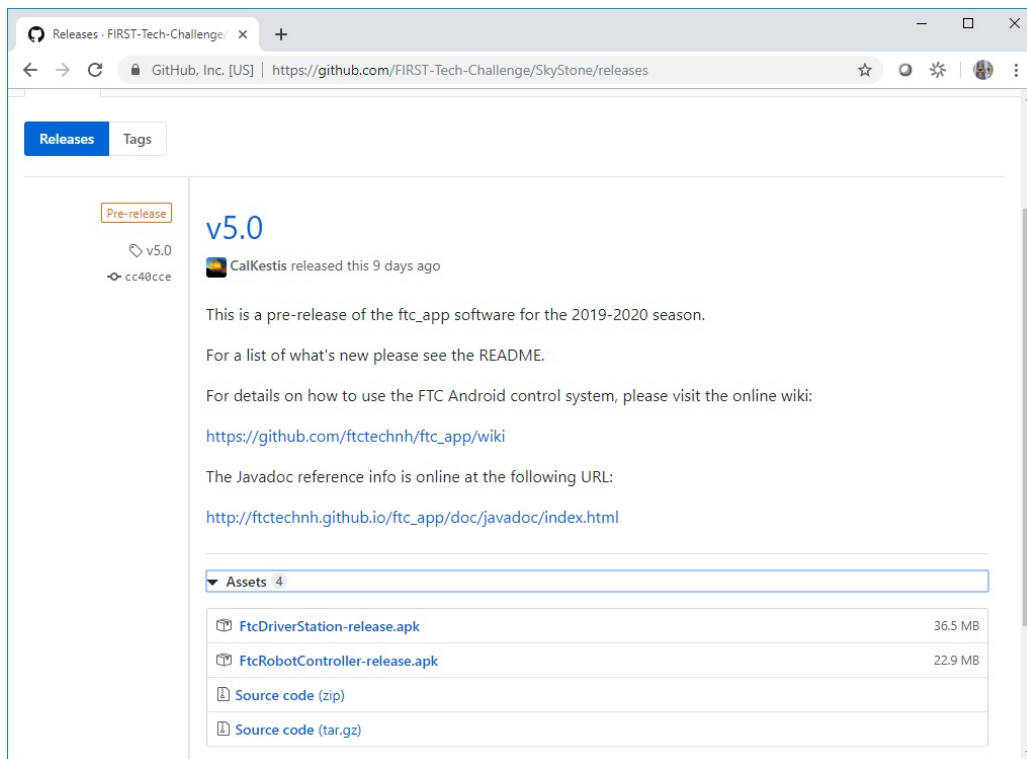
<https://github.com/FIRST-Tech-Challenge/SkyStone>

**Important Note for Advanced GitHub Users:** This tutorial 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 *clone* a local copy of the current season's public GitHub 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.

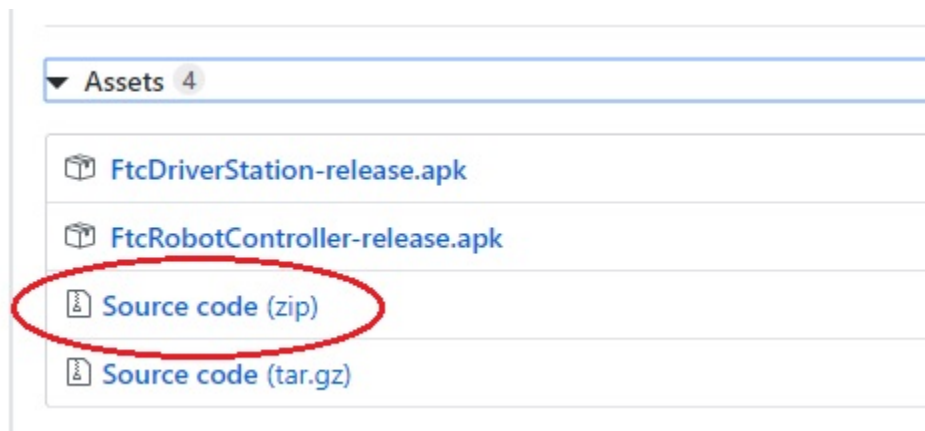
**Gracious Professionalism®** - "Doing your best work while treating others with respect and kindness - It's what makes FIRST, first."



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.



Each software release should include an **Assets** section that you can use to download the software that you will need to program your robot. Note that you might have to click on the triangular symbol to expand this **Assets** section.

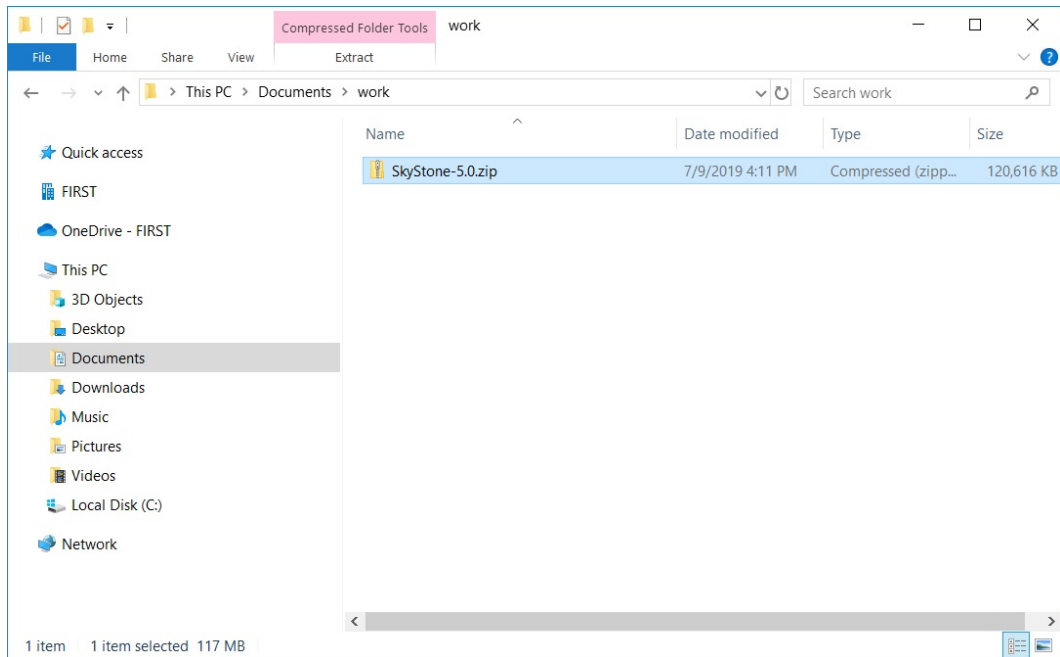


Click on the Source code (zip) link to download the compressed Android Studio project folder.

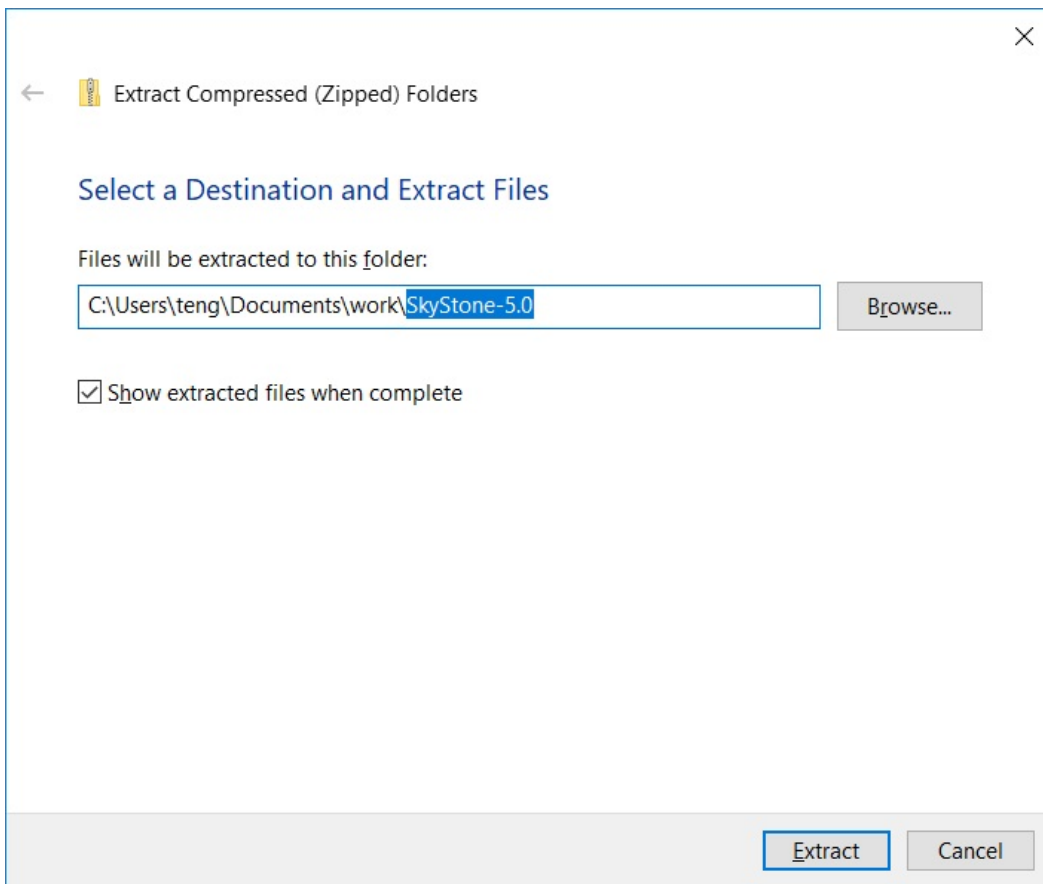
## 12.2. **Extracting the Contents of the Archived Project File**

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





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 the figure below.



Highlight the suggested name for the destination folder (in the figure above, the suggested name is “SkyStone-5.0”) 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”.

Files will be extracted to this folder:

C:\Users\teng\Documents\work\SkyStone-5.0



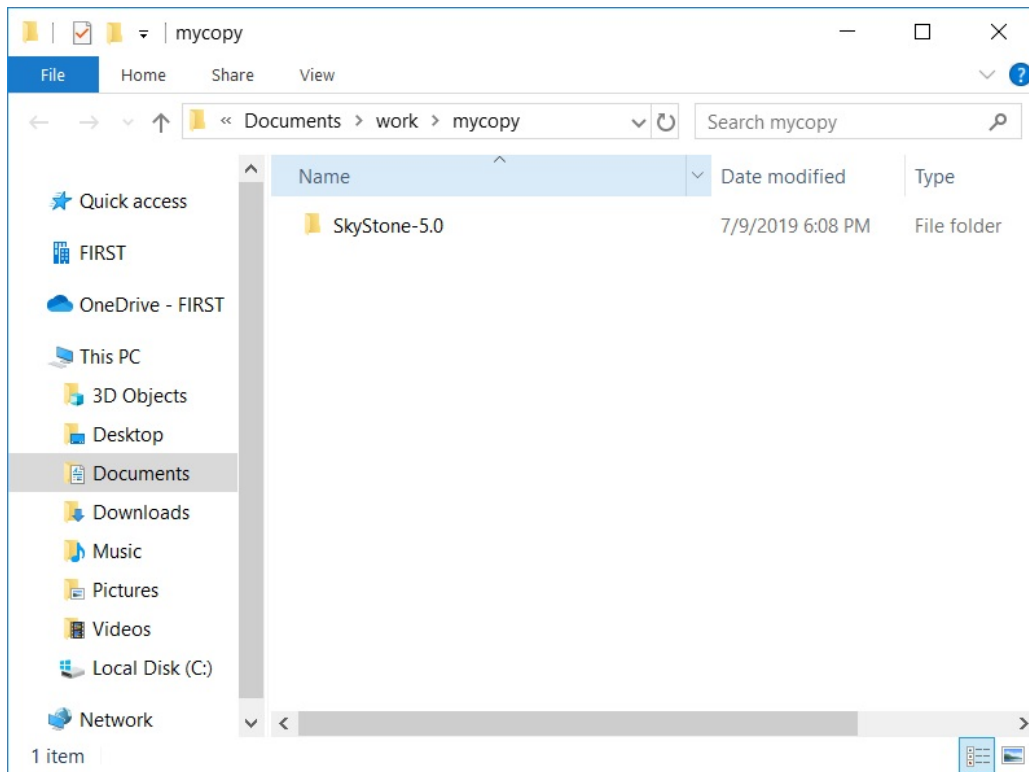
Rename destination folder...



Files will be extracted to this folder:

C:\Users\teng\Documents\work\mycopy

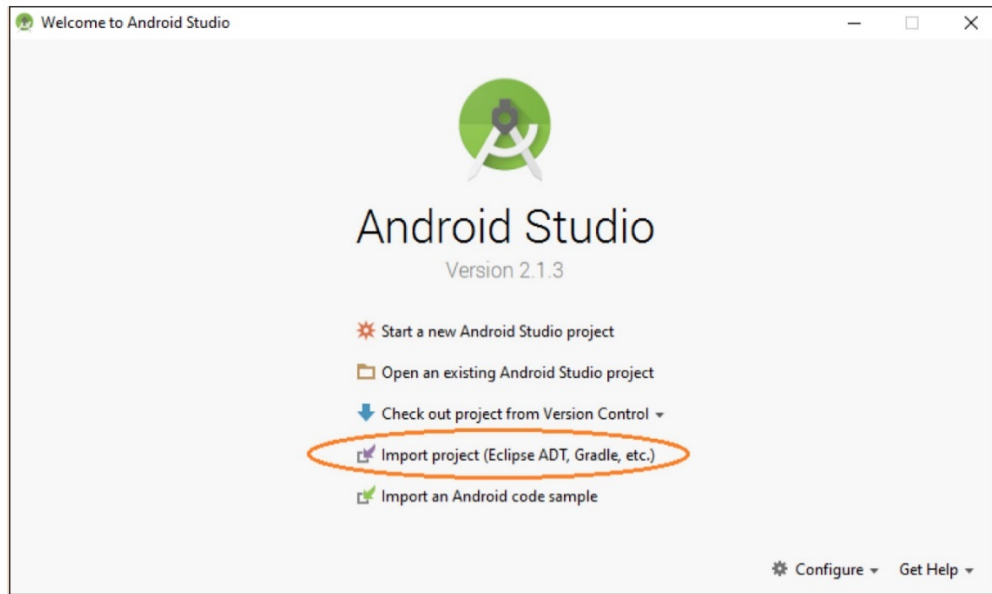
After you have renamed the destination folder, extract the contents of the archive to the folder. After the extraction process is complete, verify that the project folder was successfully extracted to its target destination.



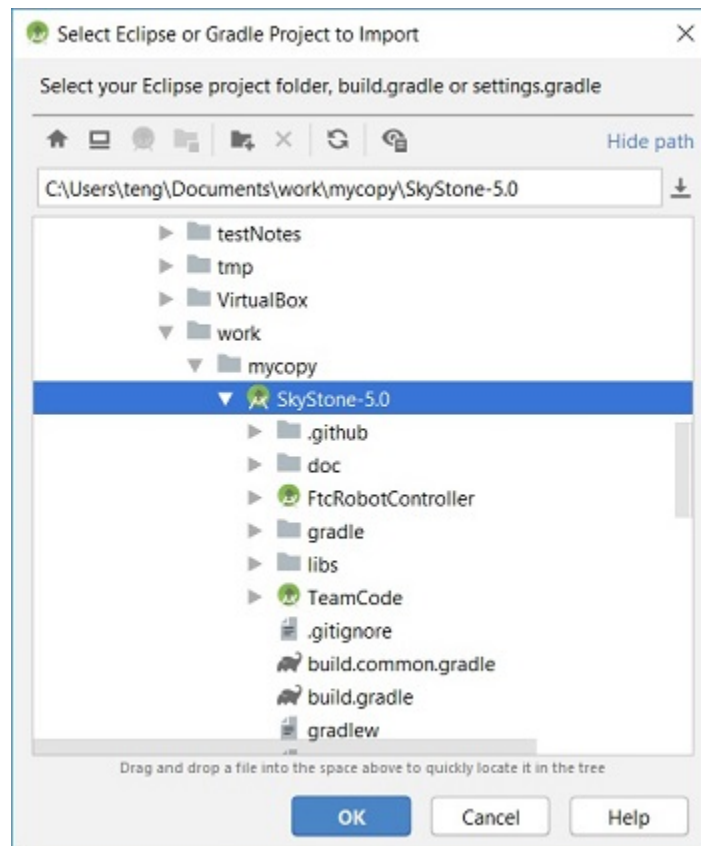
Once you have successfully extracted the contents of the archived file, you are ready to import the FTC project into Android Studio.

### ***Importing the FTC Project into Android Studio***

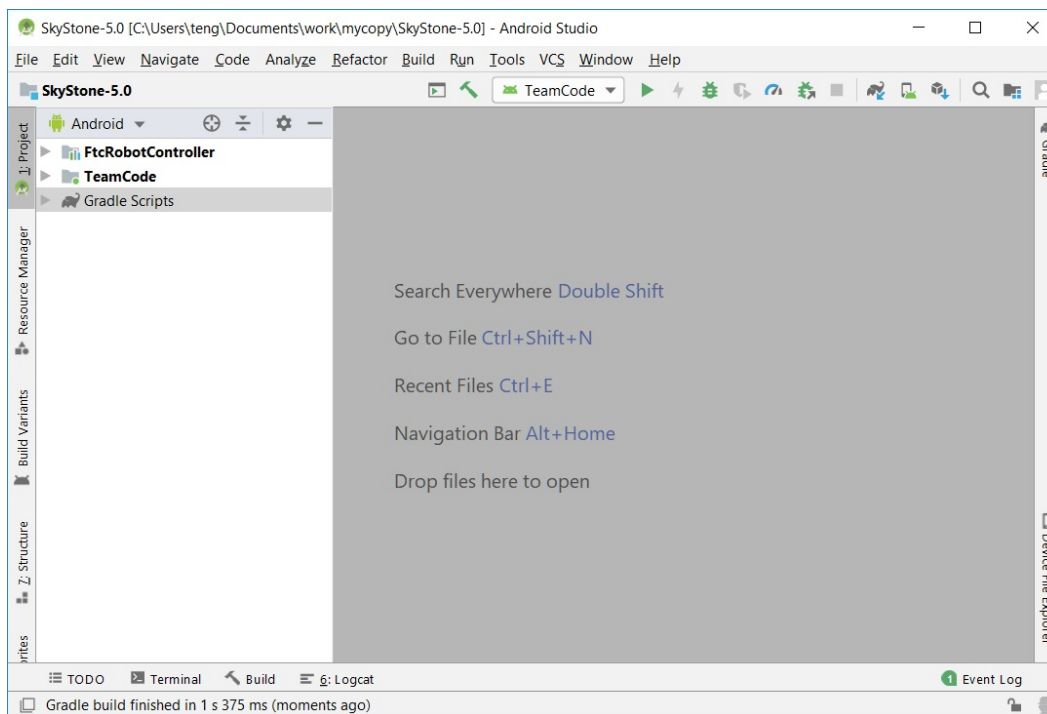
In order to import the FTC Project, you will need to launch the Android Studio software on your computer. On the main Android Studio Welcome screen, select the option to “Import project (Eclipse, ADT, Gradle, etc.)” to begin the import process.



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 an section 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.



In the figure above the project folder called “SkyStone-5.0” 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 the image below.



### 13. Enabling Developer Options

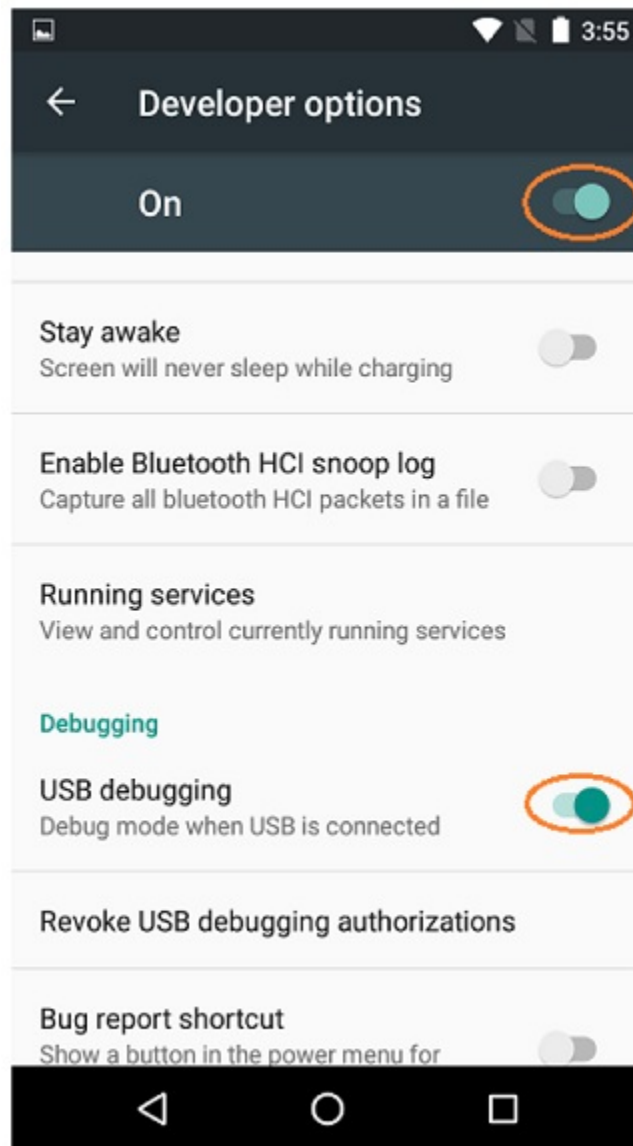
After you have configured your Android phone, you will also have to make sure that your phone is in developer mode before you will be able to install apps onto the phone using the tools that are included with Android Studio.

**Important Note for Control Hub Users:** The Control Hub has Developer Options automatically enabled from the factory, so you do **NOT** need to do this step for your Control Hub.

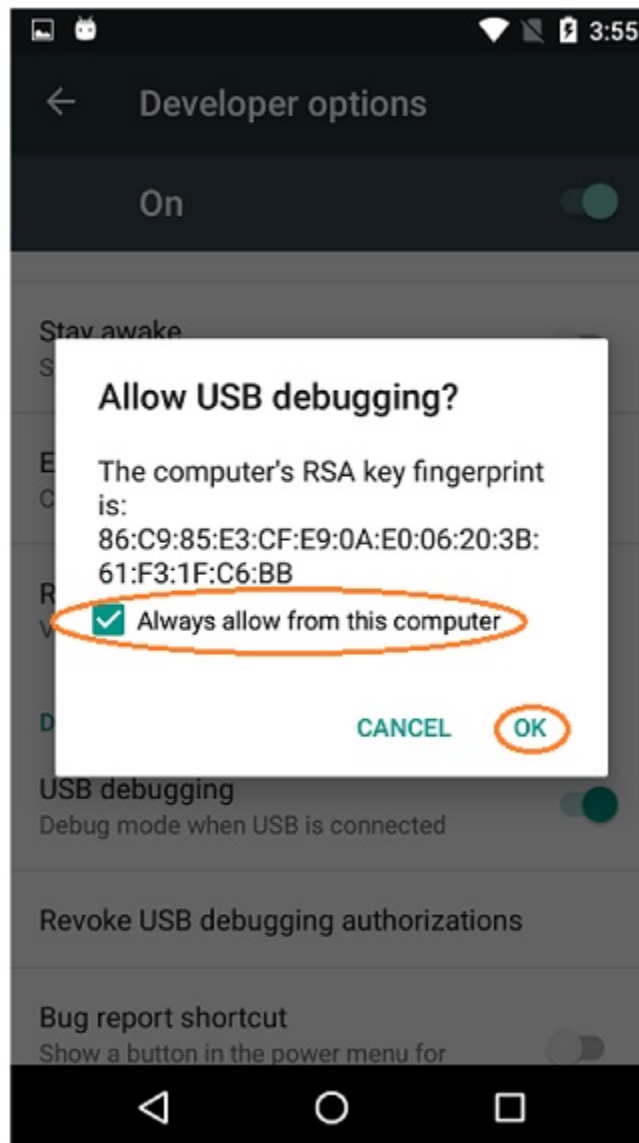
The Android Developer website contains information on how to enable Developer Options onto your phone. If you visit the following link and read the section entitled “Enabling On-device Developer Options” you will see that you can enable Developer Options on you Android phone by going to Settings->About 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.



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.

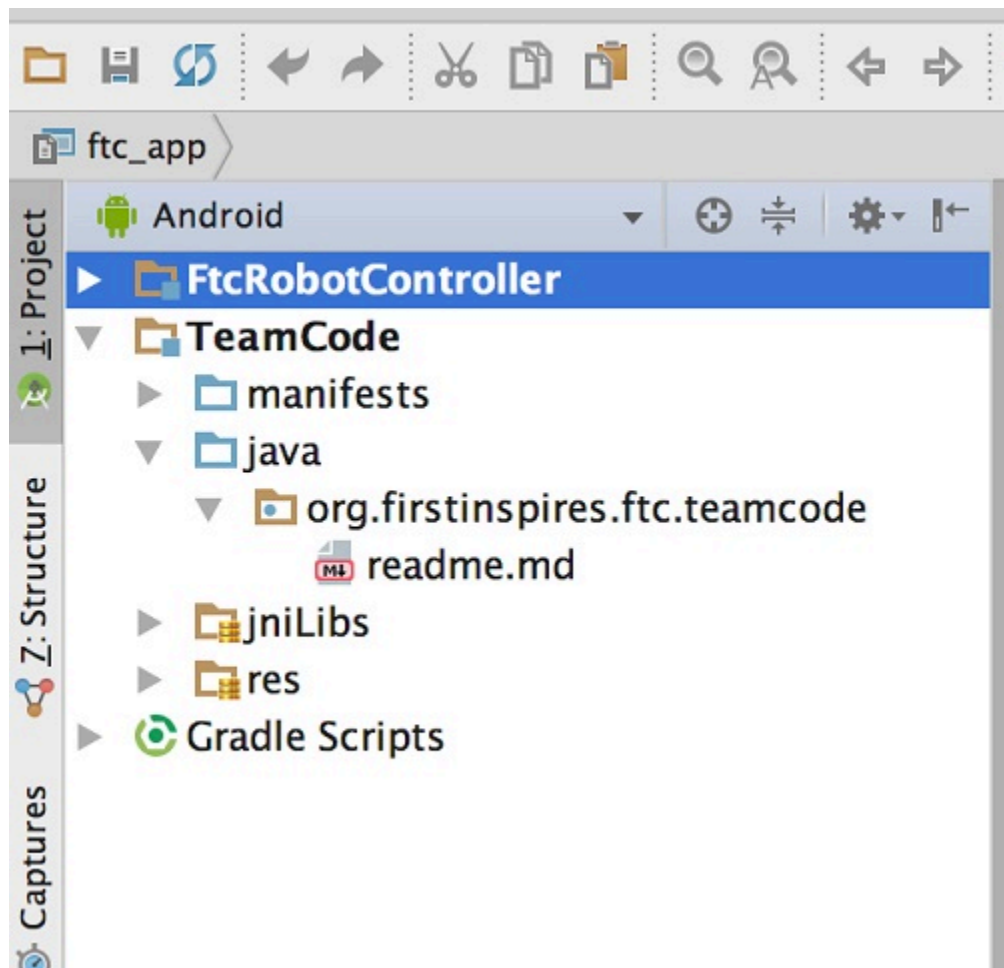


## 14. Creating and Running an Op Mode (Android Studio)

### 14.1. TeamCode Module

If you successfully imported the FTC Android Studio project folder, you will see on the project browser an Android module named "TeamCode". The FTC Android Studio project folder will be used to build a version of the FTC Robot Controller app that includes the custom op modes that you will write to control your competition robot.





When you create your classes and op modes, you will create them in the `org.firstinspires.ftc.teamcode` package that resides in the TeamCode module. This package is reserved for your use within the FTC Android Studio project folder.

### 14.2. Javadoc Reference Information

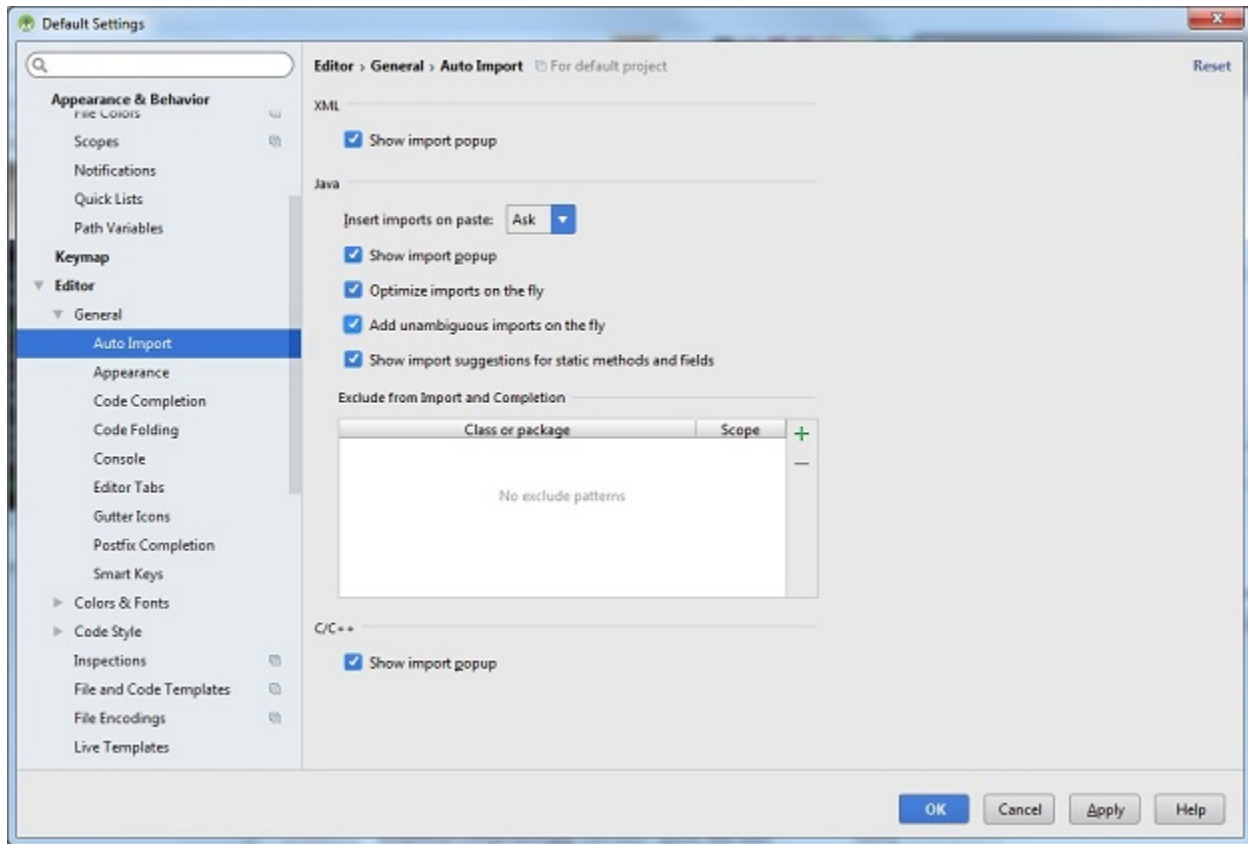
The Javadoc reference documentation for the FTC SDK is available online. Visit the following URL to view the FTC SDK documentation:

[http://ftctechnh.github.io/ftc\\_app/doc/javadoc/index.html](http://ftctechnh.github.io/ftc_app/doc/javadoc/index.html)

### 14.3. Enabling Auto Import

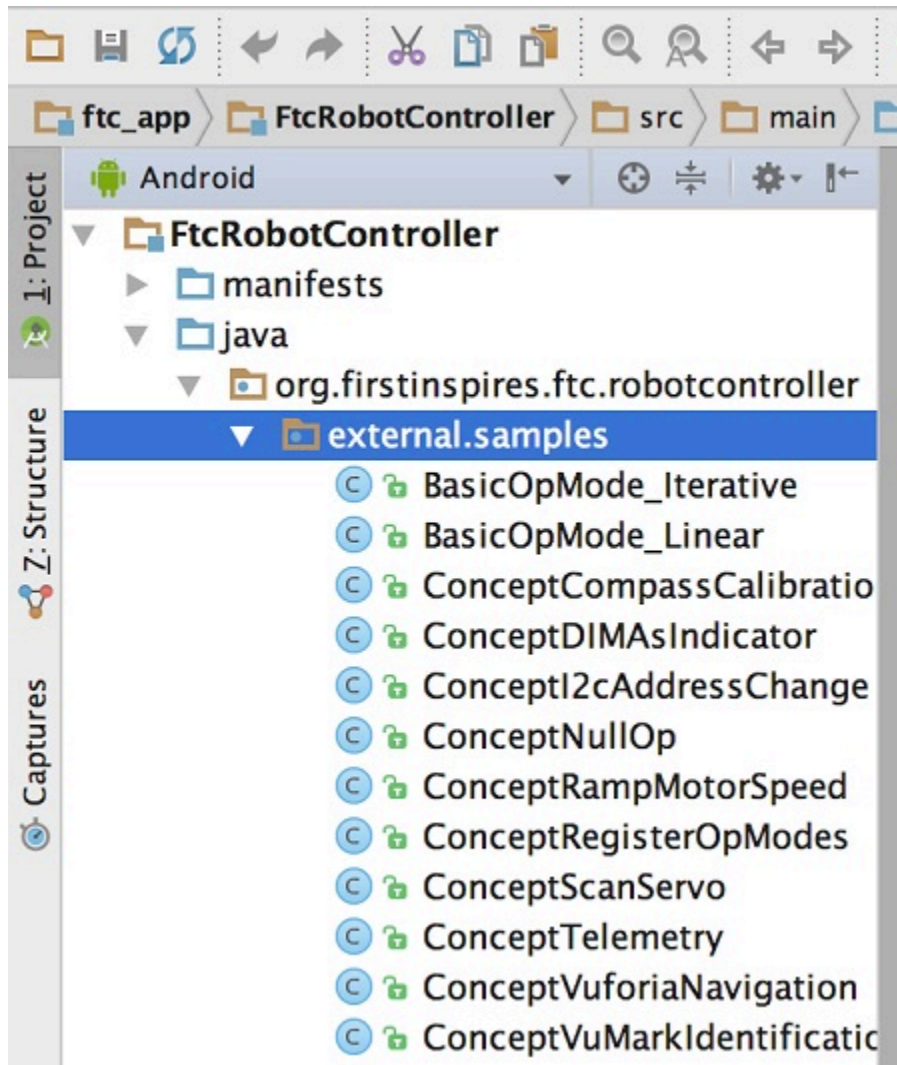
The auto import feature of Android Studio is a convenient function that helps save time as you write your op mode. If you would like to enable this feature, select the Editor->General->Auto Import item from the Android Studio Settings screen. This will display the editor's auto import settings.

Check the "Add unambiguous imports on the fly" so that Android Studio will automatically add the required import statements for classes that you would like to use in your op mode.



#### 14.4. Sample Op Modes

A great way to learn how to program a robot is to examine the sample op modes that are included with the FTC Android Studio project folder. You can locate these files in the FtcRobotController module in the package "org.firstinspires.ftc.robotcontroller.external.samples".



If you would like to use a sample op mode, copy it from the "org.firstinspires.ftc.robotcontroller.external.samples" package and move it to the "org.firstinspires.ftc.teamcode".

In your newly copied op mode, look for the following annotation,

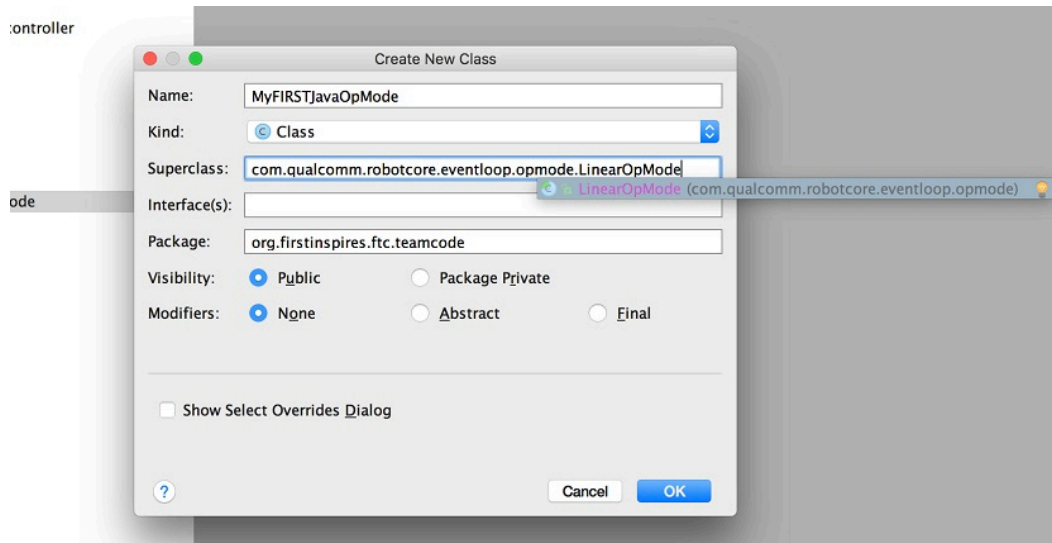
```
@Disabled
```

and comment out this line to enable the op mode and allow it to be run on the Robot Controller:

```
//@Disabled
```

### 14.5. Creating Your FIRST Op Mode

Right mouse click on the "org.firstinspires.ftc.teamcode" package and select New->Java Class from the pop-up menu. The Create New Class dialog box appear. Specify the name of the new class as "MyFIRSTJavaOpMode" and specify as its superclass the class LinearOpMode which is in the package "com.qualcomm.robotcore.eventloop.opmode".



Press the OK button to create the new class. The source code for the new class should appear in the editing pane of the Android Studio user interface.

```

MyFIRSTJavaOpMode.java x
MyFIRSTJavaOpMode
1 package org.firstinspires.ftc.teamcode;
2
3 import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
4
5 /**
6  * Created by tom on 9/19/17.
7  */
8
9 public class MyFIRSTJavaOpMode extends LinearOpMode {
10 }
11

```

Modify the main portion of your op mode so that it looks like the following code (note that the package definition and some import statements have been omitted in the following source code):

@TeleOp

```

public class MyFIRSTJavaOpMode extends LinearOpMode {
    private Gyroscope imu;
    private DcMotor motorTest;
    private DigitalChannel digitalTouch;
    private DistanceSensor sensorColorRange;
    private Servo servoTest;

```

```
@Override
public void runOpMode() {
    imu = hardwareMap.get(Gyroscope.class, "imu");
    motorTest = hardwareMap.get(DcMotor.class, "motorTest");
    digitalTouch = hardwareMap.get(DigitalChannel.class, "digitalTouch");
    sensorColorRange = hardwareMap.get(DistanceSensor.class, "sensorColorRange");
    servoTest = hardwareMap.get(Servo.class, "servoTest");

    telemetry.addData("Status", "Initialized");
    telemetry.update();
    // Wait for the game to start (driver presses PLAY)
    waitForStart();

    // run until the end of the match (driver presses STOP)
    while (opModeIsActive()) {
        telemetry.addData("Status", "Running");
        telemetry.update();
    }
}
```

We will use this source code as the framework for your first op mode. Note that Android Studio automatically saves your source code as you are editing it.

Congratulations! You've written an op mode. It does not do much, but we will modify it to make it more useful.

#### **14.6. Examining the Structure of Your Op Mode**

It can be helpful to think of an op mode as a list of tasks for the Robot Controller to perform. For a linear op mode, the Robot Controller will process this list of tasks sequentially. Users can also use control loops (such as a while loop) to have the Robot Controller repeat (or iterate) certain tasks within a linear op mode.



If you think about an op mode as a list of instructions for the robot, this set of instructions that you created will be executed by the robot whenever a team member selects the op mode called “MyFIRSTJavaOpMode” from the list of available op modes for this Robot Controller.

Let’s look at the structure of your newly created op mode. Here’s a copy of the op mode text (minus some comments, the package definition, and some import package statements):

```
@TeleOp
```

```
public class MyFIRSTJavaOpMode extends LinearOpMode {
    private Gyroscope imu;
    private DcMotor motorTest;
    private DigitalChannel digitalTouch;
    private DistanceSensor sensorColorRange;
    private Servo servoTest;
```

```
@Override
```

```
public void runOpMode() {
    imu = hardwareMap.get(Gyroscope.class, "imu");
    motorTest = hardwareMap.get(DcMotor.class, "motorTest");
    digitalTouch = hardwareMap.get(DigitalChannel.class, "digitalTouch");
```



```

sensorColorRange = hardwareMap.get(DistanceSensor.class, "sensorColorRange");
servoTest = hardwareMap.get(Servo.class, "servoTest");

telemetry.addData("Status", "Initialized");
telemetry.update();
// Wait for the game to start (driver presses PLAY)
waitForStart();

// run until the end of the match (driver presses STOP)
while (opModelsActive()) {
    telemetry.addData("Status", "Running");
    telemetry.update();
}
}
}

```

At the start of the op mode there is an annotation that occurs before the class definition. This annotation states that this is a tele-operated (i.e., driver controlled) op mode:

```
@TeleOp
```

If you wanted to change this op mode to an autonomous op mode, you would replace the “@TeleOp” with an “@Autonomous” annotation instead.

You can see from the sample code that an op mode is defined as a Java class. In this example, the op mode name is called “MyFIRSTJavaOpMode” and it inherits characteristics from the LinearOpMode class.

```
public class MyFIRSTJavaOpMode extends LinearOpMode {
```

You can also see that the OnBot Java editor created five private member variables for this op mode. These variables will hold references to the five configured devices that the OnBot Java editor detected in the configuration file of your Robot Controller.

```

private Gyroscope imu;
private DcMotor motorTest;
private DigitalChannel digitalTouch;
private DistanceSensor sensorColorRange;
private Servo servoTest;

```

Next, there is an overridden method called runOpMode. Every op mode of type LinearOpMode must implement this method. This method gets called when a user selects and runs the op mode.

```
@Override
public void runOpMode() {
```

At the start of the runOpMode method, the op mode uses an object named hardwareMap to get references to the hardware devices that are listed in the Robot Controller’s configuration file:

```

imu = hardwareMap.get(Gyroscope.class, "imu");
motorTest = hardwareMap.get(DcMotor.class, "motorTest");
digitalTouch = hardwareMap.get(DigitalChannel.class, "digitalTouch");
sensorColorRange = hardwareMap.get(DistanceSensor.class, "sensorColorRange");
servoTest = hardwareMap.get(Servo.class, "servoTest");

```

The hardwareMap object is available to use in the runOpMode method. It is an object of type HardwareMap class.



Note that when you attempt to retrieve a reference to a specific device in your op mode, the name that you specify as the second argument of the `HardwareMap.get` method must match the name used to define the device in your configuration file. For example, if you created a configuration file that had a DC motor named “motorTest”, then you must use this same name (it is case sensitive) to retrieve this motor from the `hardwareMap` object. If the names do not match, the op mode will throw an exception indicating that it cannot find the device.

In the next few statements of the example, the op mode prompts the user to push the start button to continue. It uses another object that is available in the `runOpMode` method. This object is called `telemetry` and the op mode uses the `addData` method to add a message to be sent to the Driver Station. The op mode then calls the `update` method to send the message to the Driver Station. Then it calls the `waitForStart` method, to wait until the user pushes the start button on the driver station to begin the op mode run.

```
telemetry.addData("Status", "Initialized");
telemetry.update();
// Wait for the game to start (driver presses PLAY)
waitForStart();
```

Note that all linear op modes should have a `waitForStart` statement to ensure that the robot will not begin executing the op mode until the driver pushes the start button.

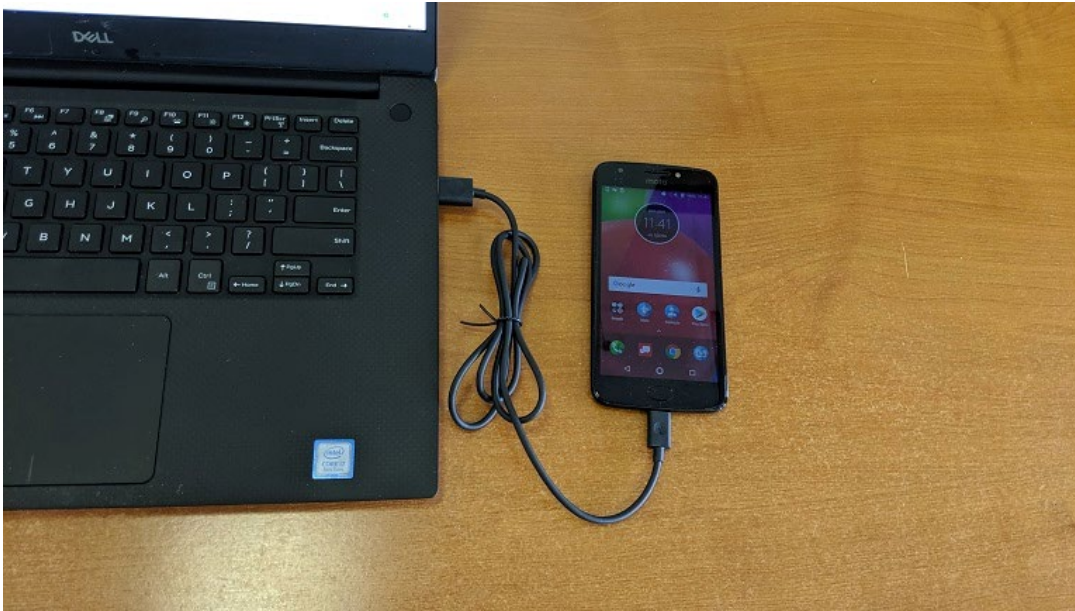
After a start command has been received, the op mode enters a while loop and keeps iterating in this loop until the op mode is no longer active (i.e., until the user pushes the stop button on the Driver Station):

```
// run until the end of the match (driver presses STOP)
while (opModelsActive()) {
    telemetry.addData("Status", "Running");
    telemetry.update();
}
```

As the op mode iterates in the while loop, it will continue to send telemetry messages with the index of “Status” and the message of “Running” to be displayed on the Driver Station.

### 14.7. **Building and Installing Your Op Mode**

Verify that the Robot Controller phone is connected to your laptop and that the laptop has USB debugging permission for the phone.



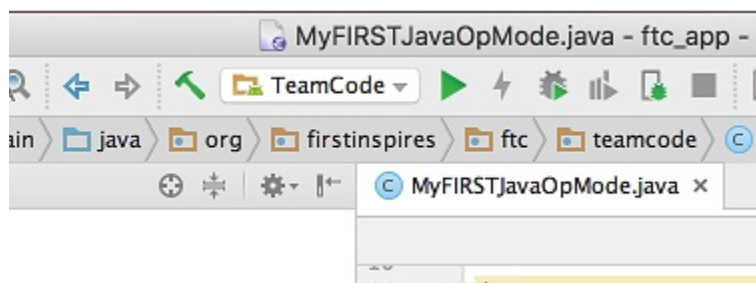
Or, if you are using a Control Hub, verify that the Control Hub is powered by a freshly charged 12V battery, and that it is connected to your laptop through its USB Type C port. Note that the Control Hub should automatically have USB debugging permission enabled.



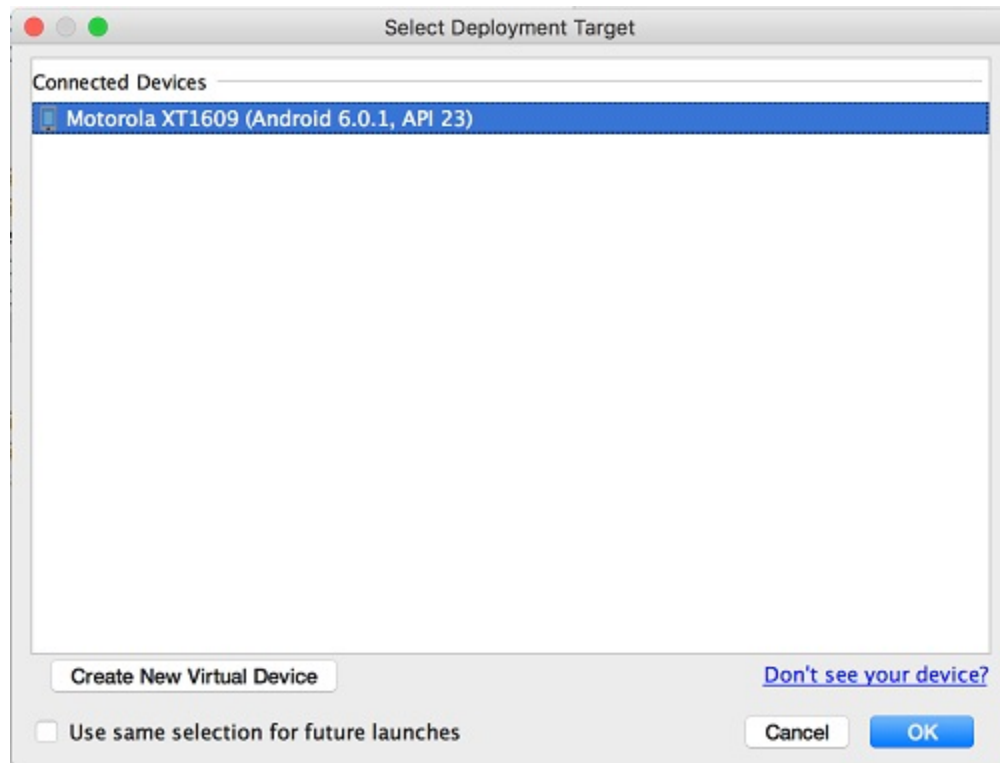
When using the Control Hub, please make sure you use the Type C port (and not the USB Mini port) to connect the Control Hub to your development laptop.



Look towards the top of the Android Studio user interface and find the little green Play or Run 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.

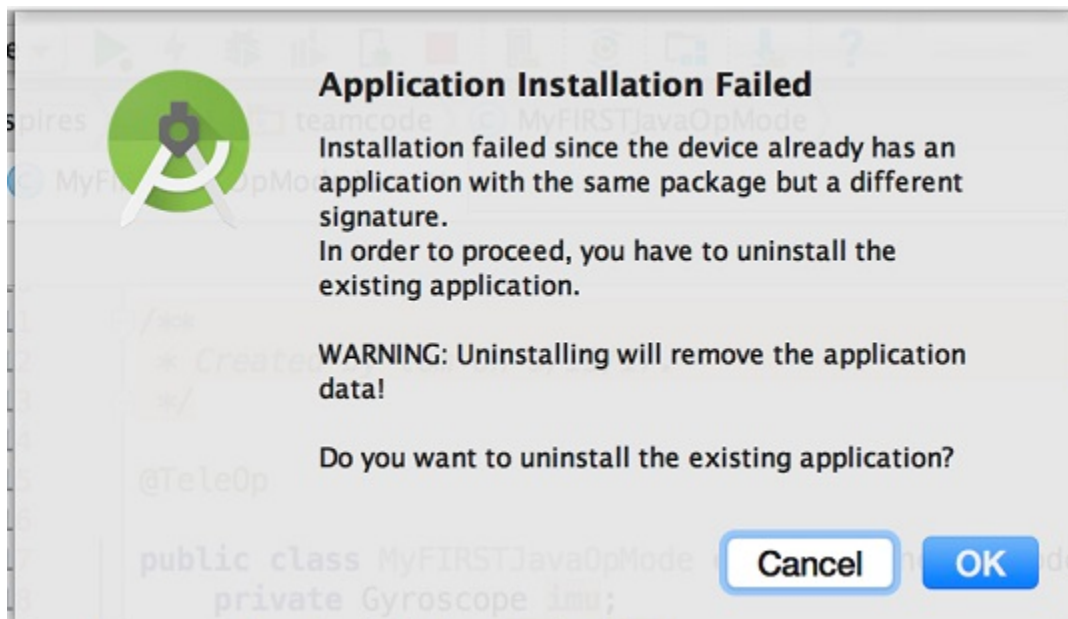


Android Studio should prompt you to select a target device to install the Robot Controller app. Your screen might look something like the image shown below.



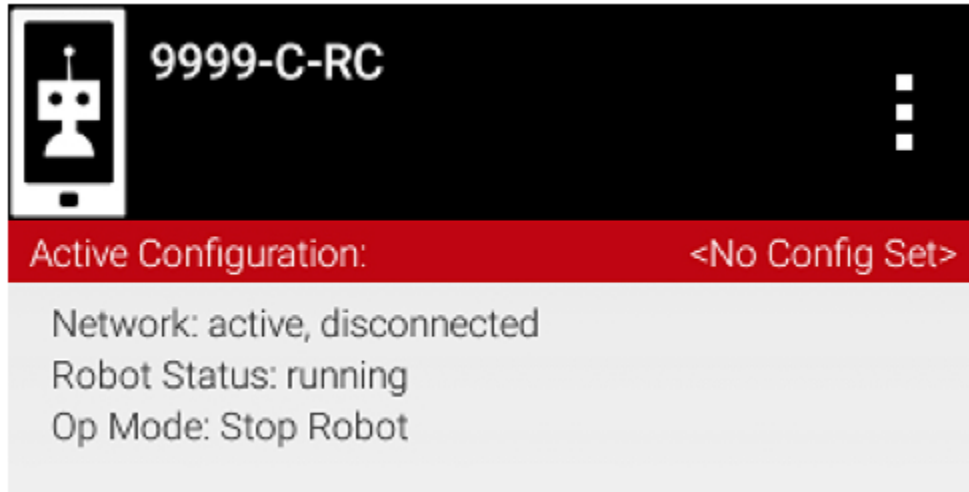
Make sure that you select the correct target device. In the figure above the Motorola phone is selected as the target device. Hit OK to build the APK file and install it on the target device.

**Note:** If you previously installed a copy of the FTC Robot Controller app from the Google Play store, the installation of your newly built app will fail the first time you attempt to install it. This is because Android Studio detects that the app that you just build has a different digital signature than the official version of the FTC Robot Controller app that was installed from Google Play.





If this happens, Android Studio will prompt you if it's OK to uninstall the previous (official) version of the app from your device and replace it with the updated version of the app. Select "OK" to uninstall the previous version and to replace it with your newly created Robot Controller App (see image above).



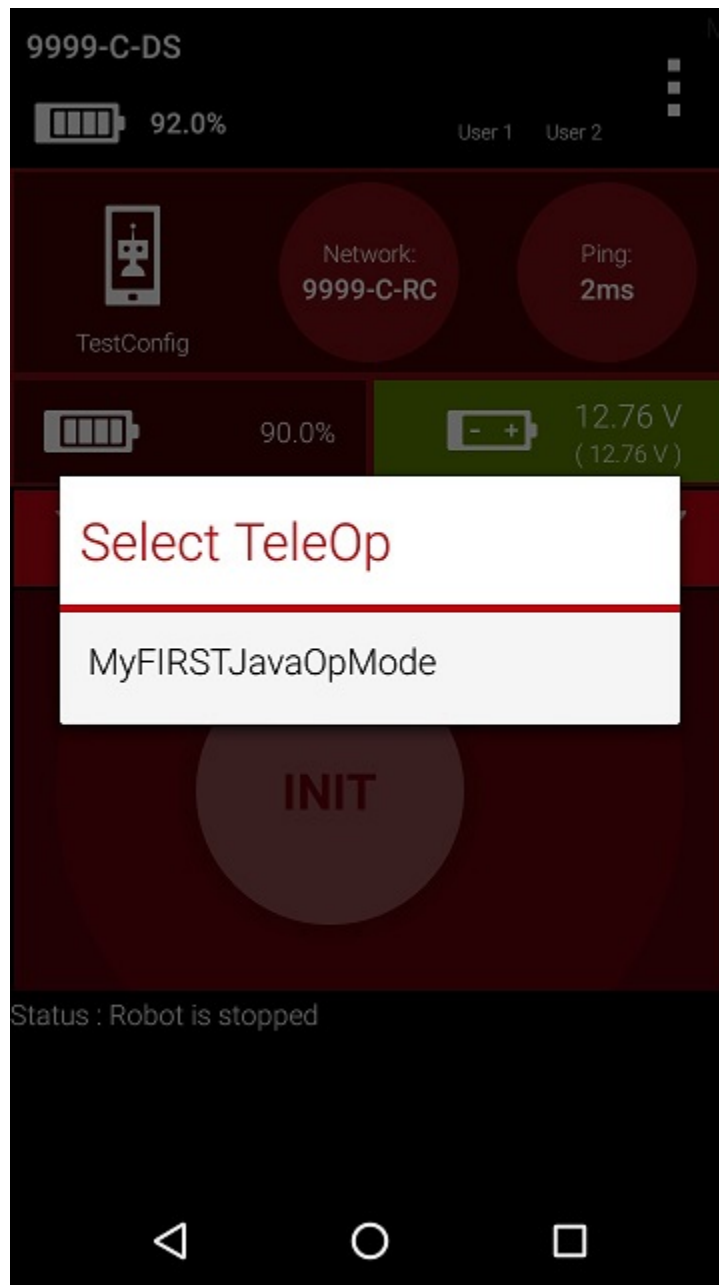
If the installation was successful, the Robot Controller app should be launched on the target Android device. If you are using an Android phone as your Robot Controller, you should see the main Robot Controller app screen displayed on the phone.

Although the Control Hub lacks a built in screen, if you are Control Hub user, you can verify that the app was installed onto your Control Hub properly by looking at your Driver Station. If the Driver Station indicates that it is successfully connected to the Control Hub (after momentarily disconnecting while the update was occurring) then the app was successfully updated.

#### **14.8. Running Your Op Mode**

If you successfully built and installed your updated Android app with your new op mode, then you are ready to run the op mode. Verify that the Driver Station is still connected to the Robot Controller. Since you designated that your example op mode is a tele-operated op mode, it will be listed as a "TeleOp" op mode.

On the Driver Station, use the "TeleOp" dropdown list control to display the list of available op modes. Select your op mode ("MyFIRSTJavaOpMode") from the list.

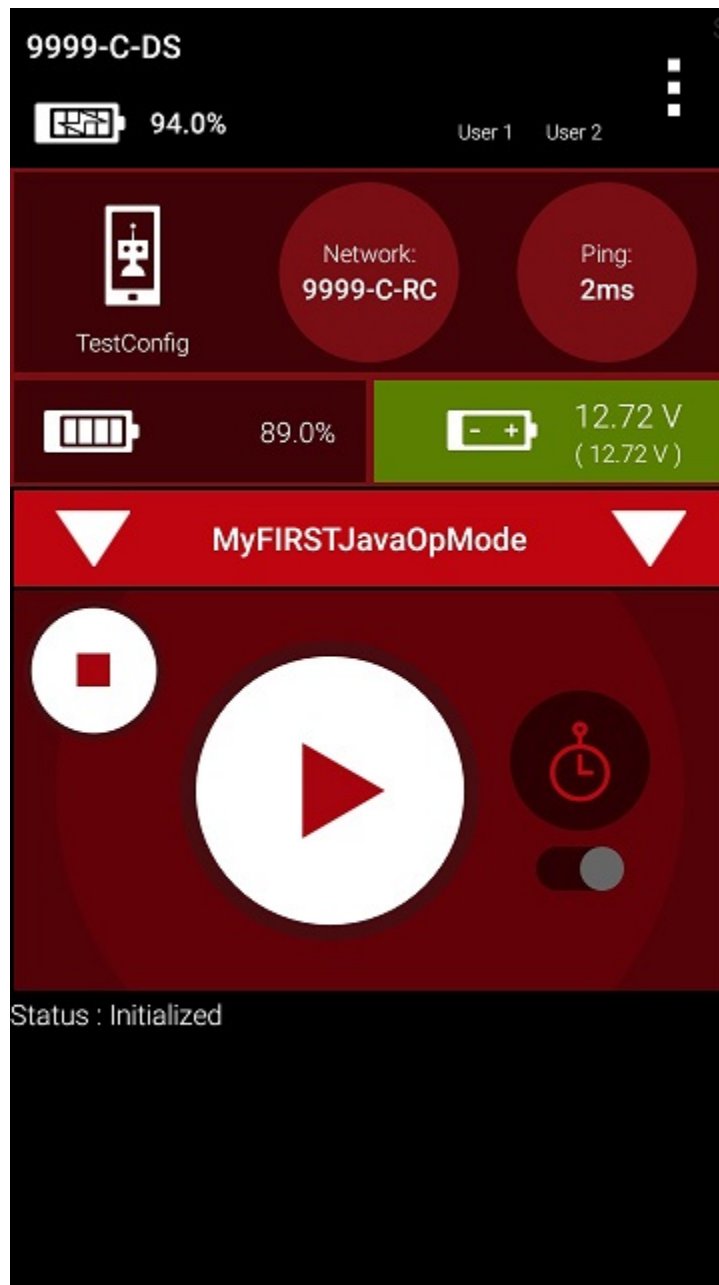


Press the INIT button to initialize the op mode.

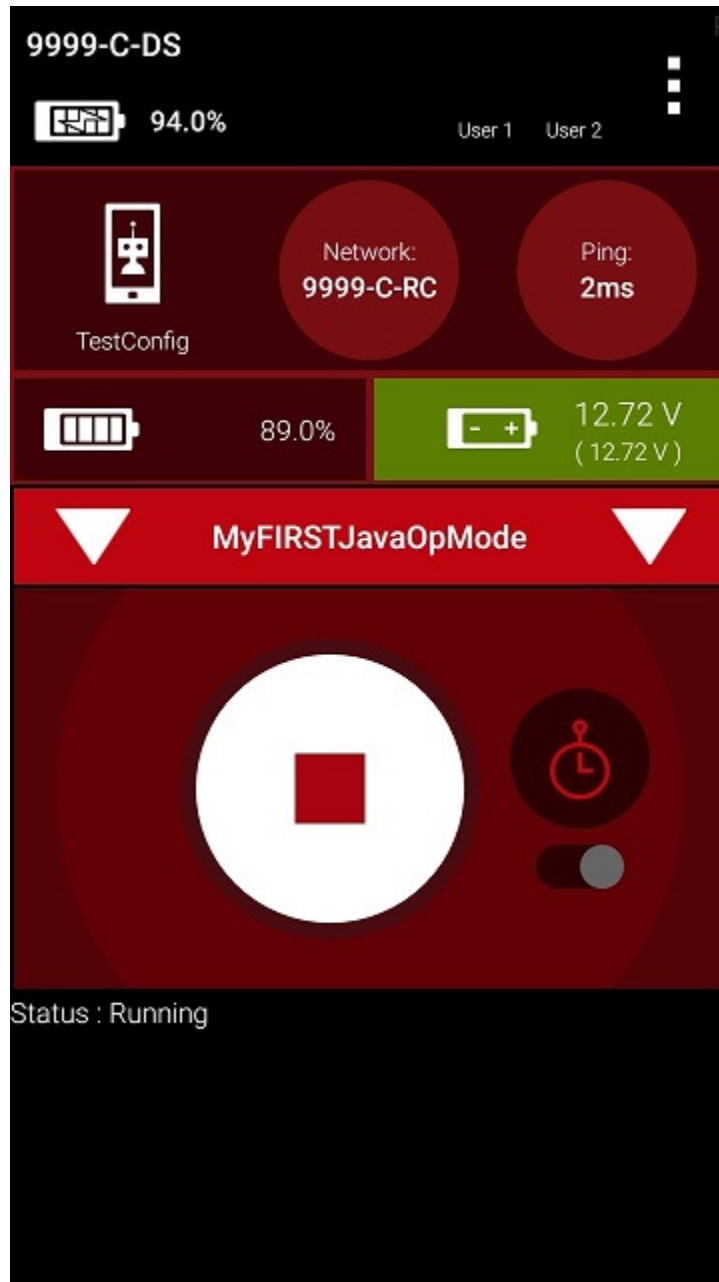


The op mode will execute the statements in the runOpMode method up to the waitForStart statement. It will then wait until you press the start button (which is represented by the triangular shaped symbol) to continue.





Once you press the start button, the op mode will continue to iterate and send the “Status: Running” message to the Driver Station. To stop the op mode, press the square-shaped stop button.



Congratulations! You ran your first java op mode!

#### 14.9. Modifying Your Op Mode to Control a Motor

Let's modify your op mode to control the DC motor that you connected and configured for your REV Expansion Hub. Modify the code for the program loop so that it looks like the following:

```
// run until the end of the match (driver presses STOP)
double tgtPower = 0;
while (opModelsActive()) {
    tgtPower = -this.gamepad1.left_stick_y;
    motorTest.setPower(tgtPower);
}
```

```
telemetry.addData("Target Power", tgtPower);
telemetry.addData("Motor Power", motorTest.getPower());
telemetry.addData("Status", "Running");
telemetry.update();
```

```
}
```

If you look at the code that was added, you will see that we defined a new variable called target power before we enter the while loop.

```
double tgtPower = 0;
```

At the start of the while loop we set the variable `tgtPower` equal to the negative value of the gamepad1's left joystick:

```
tgtPower = -this.gamepad1.left_stick_y;
```

The object `gamepad1` is available for you to access in the `runOpMode` method. It represents the state of gamepad #1 on your Driver Station. Note that for the F310 gamepads that are used during the competition, the Y value of a joystick ranges from -1, when a joystick is in its topmost position, to +1, when a joystick is in its bottommost position. In the example code above, you negate the `left_stick_y` value so that pushing the left joystick forward will result in a positive power being applied to the motor. Note that in this example, the notion of forwards and backwards for the motor is arbitrary. However, the concept of negating the joystick y value can be very useful in practice.



The next set of statements sets the power of `motorTest` to the value represented by the variable `tgtPower`. The values for target power and actual motor power are then added to the set of data that will be sent via the telemetry mechanism to the Driver Station.

```
tgtPower = -this.gamepad1.left_stick_y;
motorTest.setPower(tgtPower);
telemetry.addData("Target Power", tgtPower);
```

```
telemetry.addData("Motor Power", motorTest.getPower());
```

After you have modified your op mode to include these new statements, press the build button and verify that the op mode was built successfully.

### ***Running Your Op Mode with a Gamepad Connected***

Your op mode takes input from a gamepad and uses this input to control a DC motor. To run your op mode, you will need to connect a Logitech F310 gamepad to the Driver Station.

Before you connect your gamepad to the phone, verify that the switch on the bottom of the gamepad is set to the "X" (i.e., the "Xbox" mode) position.



Connect the gamepad to the Driver Station using the Micro USB OTG adapter cable.



Your example op mode is looking for input from the gamepad designated as the user or driver #1. Press the Start button and the A button simultaneously on the Logitech F310 controller to designate your gamepad as user #1. Note that pushing the Start button and the B button simultaneously would designate the gamepad as user #2.



If you successfully designated the gamepad to be user #1, you should see a little gamepad icon above the text “User 1” in the upper right hand corner of the Driver Station Screen. Whenever there is activity on gamepad #1, the little icon should be highlighted in green. If the icon is missing or if it does not highlight in green when you use your gamepad, then there is a problem with the connection to the gamepad.

Select, initialize and run your “MyFIRSTJavaOpMode” op mode.



If you configured your gamepad properly, then the left joystick should control the motion of the motor. As you run your op mode, be careful and make sure you do not get anything caught in the turning motor. Note that the User #1 gamepad icon should highlight green each time you move the joystick. Also note that the target power and actual motor power values should be displayed in the telemetry area on the Driver Station.



## 15. Controlling a Servo (Android Studio)

---

In this section, you will modify your op mode to control a servo motor with the buttons of the gamepad.

### 15.1. What is a Servo Motor?

A servo motor is a special type of motor. A servo motor is designed for precise motion. A typical servo motor has a limited range of motion.

In the figure below, “standard scale” 180-degree servo is shown. This type of servo is popular with hobbyists and with FIRST Tech Challenge teams. This servo motor can rotate its shaft through a range of 180 degrees. Using an electronic module known as a servo controller you can write an op mode that will move a servo motor to a specific position. Once the motor reaches this target position, it will hold the position, even if external forces are applied to the shaft of the servo.



Servo motors are useful when you want to do precise movements (for example, sweep an area with a sensor to look for a target or move the control surfaces on a remotely controlled airplane).

### 15.2. Modifying Your Op Mode to Control a Servo

Let's modify your op mode to add the logic required to control a servo motor. For this example, you will use the buttons on the Logitech F310 gamepad to control the position of the servo motor.

With a typical servo, you can specify a target position for the servo. The servo will turn its motor shaft to move to the target position, and then maintain that position, even if moderate forces are applied to try and disturb its position.

For the FIRST Tech Challenge control system, you can specify a target position that ranges from 0 to 1 for a servo. A target position of 0 corresponds to zero degrees of rotation and a target position of 1 corresponds to 180 degrees of rotation for a typical servo motor.





In this example, you will use the colored buttons on the right side of the F310 controller to control the position of the servo. Initially, the op mode will move the servo to the midway position (90 degrees of its 180-degree range). Pushing the yellow “Y” button will move the servo to the zero-degree position. Pushing the blue “X” button or the red “B” button will move the servo to the 90-degree position. Pushing the green “A” button will move the servo to the 180-degree position.



Modify your op mode to add the following code:

```
// run until the end of the match (driver presses STOP)
double tgtPower = 0;
while (opModelsActive()) {
    tgtPower = -this.gamepad1.left_stick_y;
    motorTest.setPower(tgtPower);
    // check to see if we need to move the servo.
    if(gamepad1.y) {
        // move to 0 degrees.
        servoTest.setPosition(0);
    } else if (gamepad1.x || gamepad1.b) {
        // move to 90 degrees.
        servoTest.setPosition(0.5);
    } else if (gamepad1.a) {
        // move to 180 degrees.
        servoTest.setPosition(1);
    }
    telemetry.addData("Servo Position", servoTest.getPosition());
    telemetry.addData("Target Power", tgtPower);
    telemetry.addData("Motor Power", motorTest.getPower());
    telemetry.addData("Status", "Running");
    telemetry.update();
}
```

This added code will check to see if any of the colored buttons on the F310 gamepad are pressed. If the Y button is pressed, it will move the servo to the 0-degree position. If either the X button or B button is pressed, it will move the servo to the 90-degree position. If the A button is pressed, it will move the servo to the 180-degree position. The op mode will also send telemetry data on the servo position to the Driver Station.

After you have modified your op mode, you can build it and then run it. Verify that gamepad #1 is still configured and then use the colored buttons to move the position of the servo.

## 16. Using Sensors (Android Studio)

### 16.1. Color-Distance Sensor

A sensor is a device that lets the Robot Controller get information about its environment. In this example, you will use a REV Robotics Color-Distance sensor to display range (distance from an object) info to the driver station.

The Color-Range sensor uses reflected light to determine the distance from the sensor to the target object. It can be used to measure close distances (up 5" or more) with reasonable accuracy. Note that at the time this document was most recently edited, the REV Color-Range sensor saturates around 2" (5cm). This means that for distances less than or equal to 2", the sensor returns a measured distance equal to 2" or so.

Modify your op mode to add a telemetry statement that will send the distance information (in centimeters) to the Driver Station.

```
telemetry.addData("Servo Position", servoTest.getPosition());
telemetry.addData("Target Power", tgtPower);
telemetry.addData("Motor Power", motorTest.getPower());
telemetry.addData("Distance (cm)", sensorColorRange.getDistance(DistanceUnit.CM));
telemetry.addData("Status", "Running");
telemetry.update();
```

After you have modified your op mode, build and install the updated Robot Controller app, then run the op mode to verify that it now displays distance on your Driver Station. Note that if the distance reads "NaN" (short for "Not a Number") it probably means that your sensor is too far from the target (zero reflection). Also note that the sensor saturates at around 5 cm.

### 16.2. Touch Sensor

The REV Robotics Touch Sensor can be connected to a digital port on the Expansion Hub. The Touch Sensor is HIGH (returns TRUE) when it is not pressed. It is pulled LOW (returns FALSE) when it is pressed.



The Expansion Hub digital ports contain two digital pins per port. When you use a 4-wire JST cable to connect a REV Robotics Touch sensor to an Expansion Hub digital port, the Touch Sensor is wired to the second of the two digital pins within the port. The first digital pin of the 4-wire cable remains disconnected.

For example, if you connect a Touch Sensor to the “0,1” digital port of the Expansion Hub, the Touch Sensor will be connected to the second pin (labeled “1”) of the port. The first pin (labeled “0”) will stay disconnected.

Modify the code in your op mode that occurs before the `waitForStart` command to set the digital channel for input mode.

```
// set digital channel to input mode.
digitalTouch.setMode(DigitalChannel.Mode.INPUT);

telemetry.addData("Status", "Initialized");
telemetry.update();
// Wait for the game to start (driver presses PLAY)
waitForStart();
```

Also, modify the code in your while loop to add an if-else statement that checks the state of the digital input channel. If the channel is LOW (false), the touch sensor button is pressed and being pulled LOW to ground. Otherwise, the touch sensor button is not pressed.

```
// is button pressed?
if (digitalTouch.getState() == false) {
    // button is pressed.
    telemetry.addData("Button", "PRESSED");
} else {
    // button is not pressed.
    telemetry.addData("Button", "NOT PRESSED");
}
```

```
telemetry.addData("Status", "Running");
telemetry.update();
```

Build and install the updated Robot Controller app, then reinitialize and restart your op mode. The op mode should now display the state of the button (“PRESSED” or “NOT PRESSED”).

## 17. Appendix A – Resources

---

### Game Forum Q&A

<https://ftcforum.firstinspires.org/>

Anyone may view questions and answers within the *FIRST*® Tech Challenge game Q&A forum without a password. To submit a new question, you must have a unique Q&A system username and password for your team.

### FIRST Tech Challenge Game Manuals

Part 1 and 2 - <https://www.firstinspires.org/resource-library/ftc/game-and-season-info>

### FIRST Headquarters Pre-Event Support

Phone: 603-666-3906

Mon – Fri

8:30am – 5:00pm

Email: [Firsttechchallenge@firstinspires.org](mailto:Firsttechchallenge@firstinspires.org)

### FIRST Websites

*FIRST* homepage – [www.firstinspires.org](http://www.firstinspires.org)

[FIRST Tech Challenge Page](#) – For everything *FIRST* Tech Challenge.

[FIRST Tech Challenge Event Schedule](#) – Find *FIRST* Tech Challenge events in your area.

### FIRST Tech Challenge Social Media

[FIRST Tech Challenge Twitter Feed](#) - If you are on Twitter, follow the *FIRST* Tech Challenge Twitter feed for news updates.

[FIRST Tech Challenge Facebook page](#) - If you are on Facebook, follow the *FIRST* Tech Challenge page for news updates.

[FIRST Tech Challenge YouTube Channel](#) – Contains training videos, game animations, news clips, and more.

[FIRST Tech Challenge Blog](#) – Weekly articles for the *FIRST* Tech Challenge community, including outstanding volunteer recognition!

[FIRST Tech Challenge Team Email Blasts](#) – contain the most recent *FIRST* Tech Challenge news for teams.

### Feedback

We strive to create support materials that are the best they can be. If you have feedback about this manual, please email [firsttechchallenge@firstinspires.org](mailto:firsttechchallenge@firstinspires.org). Thank you!